TEKST NR 449 2006

Matematikken bag Mars-missionen

En empirisk undersøgelse af matematikken i MER med fokus på kildekodning og kanalkodning

Uffe Thomas Jankvist & Bjørn Toldbod

Februar 2006

TEKSTER fra

IMFUFA ROSKILDE UNIVERSITETSCENTER
INSTITUT FOR STUDIET AF MATEMATIK OG FYSIK SAMT DERES FUNKTIONER I UNDERVISNING, FORSKNING OG ANVENDELSER

IMFUFA
Roskilde Universitetscenter
Postboks 260
DK-4000 Roskilde

t +45 46 74 22 63
f +45 46 74 30 20
m imfufa@ruc.dk
w imfufa.ruc.dk

Uffe Thomas Jankvist & Bjørn Toldbod: Matematikken bag Mars-missionen

IMFUFA tekst nr. 449/2006 392 sider issn 0106-6242

I denne specialerapport foretages en empirisk undersøgelse af en del af de anvendelser af matematik, som ligger til grund for Mars Exploration Rover (MER) missionen fra 2004. Undersøgelsen er delt i to dele.
Den ene del omhandler den kilde- og kanalkodning, der udgør en del af det matematiske grundlag for missionens kommunikationssystem. Undersøgelsens mål er at afdække, hvilken betydning disse teorier har for missionens kommunikation og for missionen generelt. Derudover undersøges i hvilken grad Mars-missionerne fra Pathfinder og frem har haft indflydelse på udviklingen inden for kilde- og kanalkodning.

Den anden del af undersøgelsen drejer sig om den generelle matematik i de nævnte missioner. Målet er at beskrive nogle generelle træk ved matematikken og ved den måde, hvorpå den optræder i det daglige arbejde ved NASAs Jet Propulsion Laboratory (JPL) – institutionen, der leder missionerne.

Undersøgelsen konkluderer, at kilde- og kanalkodning samlet indtager en central rolle i missionen især ved væsentligt at formindske nogen af de problemer, der hæmmer det videnskabelige udbytte af missionen. Det vurderes, at de nævnte Mars-missioner stort set ikke har haft betydning for udviklingen af teorierne for kilde- og kanalkodning.

Til sidst konkluderes det, at JPL i høj grad er en forbruger af matematik, selvom institutionen ikke selv udvikler den matematik, der indgår i missionerne. Desuden er matematikken i det daglige arbejde ved JPL ofte implementeret i software og dermed skjult for institutionens forskere. Samlet ser det ikke ud til, at JPL og Mars-missionerne bidrager til den matematiske grundforskning.

IMFUFA-tekst nr. 449 består af tre deltekster; a, b og c. 449a indeholder specialets undersøgelse og besvarelse af problemformuleringen. 449b indeholder den grundlæggende teori for forståelsen af MERs billedkomprimering og kanalkodning. I 449c findes samtlige transskriberede interviews fra DTU, Brown University, MIT og JPL.

Abstract in English

The subject of the present master thesis is an empirical study concerning some of the underlying mathematical applications of the Mars Exploration Rover (MER) mission of 2004. The study is separated into two parts.

One part concerns the source and channel coding that form part of the mathematical basis of the mission's telemetry system. The study seeks to discover the importance of the theories both in the mission's communications and in the mission in general. Moreover it is investigated to what extend the Mars missions since Pathfinder have influenced the development of source and channel coding.

The other part of the study concerns the general mathematics of the above-mentioned missions to Mars. The goal is to describe general characteristics of the mathematics and the way in which it comes into the daily work at NASAs Jet Propulsion Laboratory (JPL) – the institute in control of the missions.

The study concludes that the source and channel coding plays a central role in the mission, particularly in reducing some of the factors, that limit the scientific payoff of the mission. The importance of the Mars missions in question to the development of source and channel coding is assessed to being virtually none.

Finally it is concluded that the JPL to a large extend acts as a consumer of mathematics although the institution does not develop the mathematics on its own. Also, often the mathematics of the daily work is implemented in software and thereby hidden from the scientists at the JPL. All in all it seems unreasonable to say that JPL and the Mars missions contribute to basic mathematical research.

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

T. S. Eliot, 1959

Forord

Nærværende tekst er et speciale under den matematiske kandidatuddannelse på Roskilde Universitetscenter, nærmere bestemt ved Institut for studiet af Matematik og Fysik samt deres funktioner i Undervisning, Forskning og Anvendelser (IMFUFA). Specialet, der tæller 20 ECTS, er et såkaldt *professionsprojekt* og hører under *forskervarianten*.

Specialet blev så småt påbegyndt i august måned 2004, men grundet kursusaktiviteter i efteråret 2004 på vores andet fag, datalogi, er det meste af arbejdet med specialet udført i perioden fra marts til og med september 2005.

Specialet omhandler udvalgte matematiske problemstillinger i den igangværende Mars-mission MER (Mars Exploration Rover)[†].

Rapporten er skrevet i PDF-LaTeX, nærmere bestemt i en variant af dokumentklassen `imfbook`[‡] i IMFUFA-LaTeX, der bygger oven på The Memoir Class.

Taksigelser

Vi ønsker at takke vores nuværende specialevejleder lektor Anders Madsen (RUC), der så sent som august 2005 indvilgede i at overtage vejledningen af specialet. Specialet var på det tidspunkt så fremskredent, at det var svært at påvirke såvel form som indhold. Alligevel formåede Anders at komme med en række forslag til forbedringer, som hjalp os til at færdiggøre rapporten. Anders' bidrag har været med til at hæve specialet på såvel dets matematiske plan som dets meta-matematiske plan.

Ligeledes tak til vores tidligere specialevejleder lektor Bernhelm Booß-Bavnbek (RUC), der satte os på sporet af *Matematikken bag Mars-missionen*. Allerede fra begyndelsen af specialeforløbet viste Bernhelm os stor tillid ved at lade os gå vores egne veje, selvom han var særdeles interesseret i vores undersøgelser. Desværre følte Bernhelm sig nødsaget til at indstille sin vejledning af specialet i utide.

Også en varm tak til professor emeritus Philip J. Davis (Brown University) for at have ydet sin støtte og hjælp til specialet i form af inspiration til vores undersøgelse samt til vores metode.

Endvidere ønsker vi at takke lektor Tinne Hoff Kjeldsen (RUC) for at have været behjælpelig med forslag til litteratur og i særdeleshed også for at have gennemlæst og kommenteret dele af rapporten.

† Billederne på rapportens omslag er taget af roveren Spirit undervejs i dens færden på Mars' overflade. På forsiden bevæger Spirit sin robotarm hen over en sten for at tage en serie billeder med sit mikroskopkamera. På bagsiden ses et billede fra Spirits bagerste kamera. Roverens venstre baghjul er synligt til højre i billedet. `http://marsrovers.jpl.nasa.gov/gallery/press/spirit`.

‡ Højden af tekstblokken er i nærværende rapport gjort større end den i `imfbook`.

Derudover ønsker vi at takke følgende personer og institutioner: Professor Stig Andur Pedersen (RUC), for at have hjulpet med at afklare videnskabsteoretiske såvel som videnskabshistoriske spørgsmål. Professor Mogens Niss (RUC) og lektor Jens Højgaard (RUC) for at have hjulpet med indsnævring af problemfelt. Ph.d.-studerende Bjarke Skipper Petersen (RUC) for diverse litteraturforslag.

Docent Tom Høholdt (DTU), professor Jørn Justesen (DTU) og lektor Søren Forchhammer (DTU) for at have været behjælpelige i specialeforløbets indledende faser, dels ved at lade sig interviewe, dels ved at foreslå relevant litteratur og dels ved at foreslå kontaktpersoner. Lektor Morten Bo Madsen (KU) for at have sat os i forbindelse med vores 'main man' dr. William Folkner på JPL i Pasadena. Professor Robert G. Gallager (MIT) og professor Robert J. McEliece (CalTech) for ligeledes at have været behjælpelige med etablering af kontakter.

Dr. William Folkner (JPL) og dr. Albert Haldemann (JPL) for at tage sig tid til at arrangere møder til os på JPL, vise os stedets faciliteter samt lade sig interviewe og skaffe os materialer om MER-missionen. Dr. Jon Hamkins (JPL), dr. Matthew Klimesh (JPL), dr. Aaron Kiely (JPL), dr. Jacob Matijevic (JPL), dr. Miguel san Martin (JPL) og dr. Mark W. Maimone for ligeledes at lade sig interviewe.

Professor Gilbert Strang (MIT) og dr. Ralph Kötter (MIT) for at lade sig interviewe under vores ophold i Cambridge og derigennem give os indblik i universitetsverdenens syn på matematikken i Mars-missionerne.

Karen og Olga Nybyes Mindelegat, Oticon Fonden, Roskilde Bys Mindelegat for Kong Chr. X samt RUCs rejsefond for at muliggøre vores studierejse til henholdsvis Brown University i Providence, Rhode Island, MIT i Cambridge, Massachusetts og JPL i Pasadena, Californien.

Forfatter til IMFUFA-LaTeX og statistiker lektor Jørgen Larsen (RUC) for at have besvaret alskens mulige og umulige LaTeX- og statistik-relaterede spørgsmål undervejs i rapportskrivningen. Civilingeniør Ib Høst (RUC) for teknisk assistance og velvilje.

Uffe ønsker specielt at takke sin lille 'kernefamilie' samt de personer, der har taget sig af denne, dels under studierejsen til USA i marts og dels i rapportskrivningens afsluttende fase.

Bjørn ønsker at takke sin kæreste for – uden sure miner – at have accepteret, at specialet har fyldt så meget i det forgangne år.

Uffe Thomas Jankvist & Bjørn Toldbod
Institut for Matematik og Fysik
Roskilde Universitetscenter
Februar 2006

Indhold

En empirisk undersøgelse af matematikken i MER med fokus på kildekodning og kanalkodning

1 Indledning **1**
- 1.1 Mars i medierne . . . 2
- 1.2 Inspirationskilder . . . 3
- 1.3 Videnskabelig motivation . . . 5
- 1.4 Problemformulering . . . 6
- 1.5 Metode . . . 7
- 1.6 Forskningsmæssige bidrag . . . 7
- 1.7 Om rapporten . . . 8

2 JPL og MER **9**
- 2.1 Jet Propulsion Laboratory . . . 9
- 2.2 Om matematikken i MER . . . 15
- 2.3 Afrunding . . . 19

3 En matematisk teori for kommunikation **21**
- 3.1 Et kommunikationssystem . . . 22
- 3.2 Kvantisering af information . . . 24
- 3.3 AWGN-kanalen . . . 30
- 3.4 Afrunding . . . 33

4 Kildekodning og kanalkodning **35**
- 4.1 Kildekodning . . . 35
- 4.2 Kanalkodning . . . 38
- 4.3 Afrunding . . . 40

5 Kildekodning i MER **41**
- 5.1 Særlige omstændigheder for roverne . . . 41
- 5.2 Overbliksbillede . . . 43
- 5.3 Forbehandling . . . 44
- 5.4 Progressiv komprimering . . . 47
- 5.5 Kontekst-modellering . . . 51
- 5.6 Entropikoderen . . . 54
- 5.7 Fejlinddæmning . . . 59
- 5.8 ICERs komprimeringsresultater . . . 61
- 5.9 Kort om LOCO-I . . . 62
- 5.10 Valg af komprimering i MER . . . 63
- 5.11 Afrunding . . . 65

6 Kanalkodning i MER **67**
6.1 Særlige omstændigheder for roverne 67
6.2 Kodningssystemet overordnet 68
6.3 Det ydre system 69
6.4 Det indre system 78
6.5 Valg af koder 84
6.6 Afrunding 85

7 Bag om matematikken i Mars-missionerne **87**
7.1 Kilde- og kanalkodningens betydning for MER 87
7.2 JPLs betydning for kildekodningens seneste udvikling . . . 91
7.3 Kort om turbokoder og LDPC-koder 93
7.4 JPLs betydning for kanalkodningens seneste udvikling . . . 96
7.5 Generelt om matematikken i de aktuelle Mars-missioner . . 99
7.6 Afrunding 103

8 Konklusion **105**

9 Efterskrift **107**
9.1 Stagnationshypotesen 107
9.2 Den videnskabshistoriske dimension 108
9.3 Afsluttende kommentarer 109

Litteratur **111**

A NASAs gennemførte Mars-missioner **117**
A.1 Tidligere missioner til Mars 118
A.2 Igangværende missioner til Mars 125
A.3 Fremtidige missioner til Mars 130
A.4 Afrunding 131

Indførelse i den grundlæggende teori for kildekodning og kanalkodning i MER

1 Indledning **1**

I Kildekodning i MER-missionen **3**

2 Introduktion til data- og billedkomprimering **5**
2.1 Komprimering med og uden tab af data 6
2.2 Elementære begreber i datakomprimering 7
2.3 Eksempler på simple komprimeringsteknikker 13
2.4 Afrunding 15

3 Wavelets **17**
3.1 Førstetrins waveletbasis 17
3.2 Filterbanks 23
3.3 p'te-trins waveletbasis 26
3.4 Billedkomprimering med wavelets 32

3.5 Lifting . . . 34
3.6 Afrunding . . . 38

4 Modellering **39**
4.1 Diskret hukommelsesløs kilde . . . 39
4.2 Markov-kilde . . . 42
4.3 Sammensatte kilder . . . 45
4.4 Afrunding . . . 46

5 Entropikodning **47**
5.1 Entropi og kodning . . . 47
5.2 Huffman-kodning . . . 48
5.3 Golomb-kodning . . . 55
5.4 Sammenflettet entropikodning . . . 57
5.5 Adaptiv entropikodning overordnet . . . 59
5.6 Afrunding . . . 62

II Kanalkodning i MER-missionen **63**

6 Introduktion til kanalkodning **65**
6.1 Elementære kodningsteoretiske begreber . . . 66
6.2 Fejldetektion versus fejlkorrektion . . . 68
6.3 En geometrisk tilgang til blokkoder . . . 69
6.4 Perfekte koder . . . 71
6.5 Eksempler på simple koder . . . 73
6.6 Afrunding . . . 74

7 Lineære koder **75**
7.1 Grundlæggende for lineære koder . . . 75
7.2 Cykliske koder . . . 78
7.3 Afrunding . . . 84

8 Reed-Solomon koder **85**
8.1 Reed-Solomon koder som cykliske koder . . . 85
8.2 En moderne tilgang til Reed-Solomon koder . . . 86
8.3 Reed-Solomon koder og bit . . . 90
8.4 Afrunding . . . 92

9 Foldningskoder **95**
9.1 Fra blokkoder til foldningskoder . . . 95
9.2 Foldningsindkoderen . . . 96
9.3 Polynomial beskrivelse af foldningskoder . . . 99
9.4 Egenskaber for foldningskoder . . . 103
9.5 Matrixbeskrivelse af foldningskoder . . . 105
9.6 Trellis-beskrivelse af foldningskoder . . . 107
9.7 Afrunding . . . 109

Litteratur **111**

A Fouriertransformationen på $\mathbb{Z}$ **117**

B Wavelets **127**
B.1 Waveletbeviser 127
B.2 Resultater for Haar-basen 144

Transskriberede interviews fra DTU, Brown University, MIT og JPL

1 Interviews fra Danmarks Tekniske Universitet **1**
1.1 Docent Tom Høholdt 26/8 2004 1
1.2 Professor Jørn Justesen & docent Tom Høholdt 27/9 2004 . 3
1.3 Lektor Søren Forchhammer 31/1 2005 6

2 Interview fra Brown University **13**
2.1 Professor emeritus Philip J. Davis 6/3 2005 13

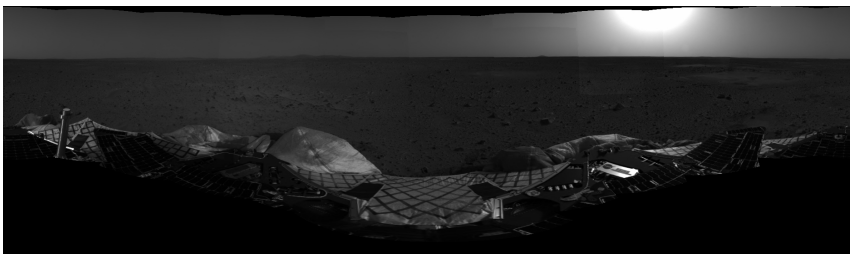
3 Interviews fra Massachusetts Institute of Technology **25**
3.1 Doktor Ralph Kötter 7/3 2005 25
3.2 Professor Gilbert Strang 9/3 2005 29

4 Interviews fra Jet Propulsion Laboratory **33**
4.1 Doktor Aaron Kiely 14/3 2005 33
4.2 Doktor Matthew Klimesh 14/3 2005 44
4.3 Doktor Jon Hamkins 14/3 2005 47
4.4 Doktor Jacob Matijevic 14/3 2005 50
4.5 Doktor Miguel san Martin & doktor William Folkner 14/3 2005 54
4.6 Doktor William Folkner 16/3 2005 64
4.7 Doktor William Folkner 17/3 2005 70
4.8 Doktor Mark W. Maimone 17/3 2005 78

1 Indledning

Den 4. januar 2004 gik Mars Exploration Rover A, med tilnavnet *Spirit*, ind i Mars' atmosfære. Det 827 kg tunge fartøj havde på dette tidspunkt en fart på 19.300 km/t. I løbet af de næste 4 minutter blev farten ved mødet mellem atmosfæren og fartøjets frontskal reduceret til 1.600 km/t. Fartøjets faldskærm blev på dette tidspunkt foldet ud, og hastigheden faldt hurtigt til 300 km/t. Hver gang fartøjet udførte en ny handling, blev en tone via en satellit transmitteret til Jorden som indikation på succes. 100 meter over Mars' overflade affyredes fartøjets bremseraketter, og tilsidst udfoldede de kæmpemæssige airbags sig. Det indhyllede fartøj ramte planetens overflade med en hastighed på cirka 50 km/t. Derefter hoppede og rullede airbag-kuglen omkring en kilometer henover Mars' overflade, ligesom Mars Pathfinder havde gjort det syv år tidligere. Da landingmodulet stoppede sin fremfærd blev dets airbags afmonteret, dets kronblade udfoldet og den igennem seks måneder indkapslede rover, Spirit, kunne endelig folde sine solpaneler ud. Tre timer senere transmitterede Spirit sit første billede af Gusev-krateret til Jorden. Den 15. januar forlod Spirit sit landingsmodul og kørte ud på Mars' overflade. 10 dage senere, den 25. januar gentog hele scenariet sig for Mars Exploration Rover B, også kaldet *Opportunity*.

De to roveres landinger blev fulgt med stor interesse. Ikke mindst på NASAs[1] Jet Propulsion Laboratory[2] (JPL), hvor der herskede en euforisk stemning og blev grædt et utal af glædestårer, da det stod klart, at roverne havde klaret landingen og var fuldt ud operationsdygtige. Euforien



Figur 1.1 Første billede af Gusev-krateret offentliggjort til pressen. Billedet, som er en mosaik, er taget af navigationskameraet ombord på Spirit. http://nssdc.gsfc.nasa.gov/planetary/mars/mars_exploration_rovers/mera_images.html

1 National Aeronautics and Space Administration.
2 Jet Propulsion Laboratory hører under California Institute of Technology (CalTech) og ligger i Pasadena, Californien.

skyldtes, at *Mars Exploration Rover*-missionen (MER) på det tidspunkt havde været en del af arbejdet på JPL igennem en del år. Et arbejde med at planlægge missionen ned i mindste detalje og med at konstruere roverne og alt deres udstyr. Et arbejde af enormt omfang. Et arbejde, der da MER-missionen var på sit højeste, beskæftigede over 1.000 folk; ingeniører, radiofolk, geologer, fysikere, dataloger samt matematikere og mange, mange andre videnskabelige medarbejdere. Dette speciale handler om den matematik, der var involveret i disse folks arbejde.

1.1 Mars i medierne

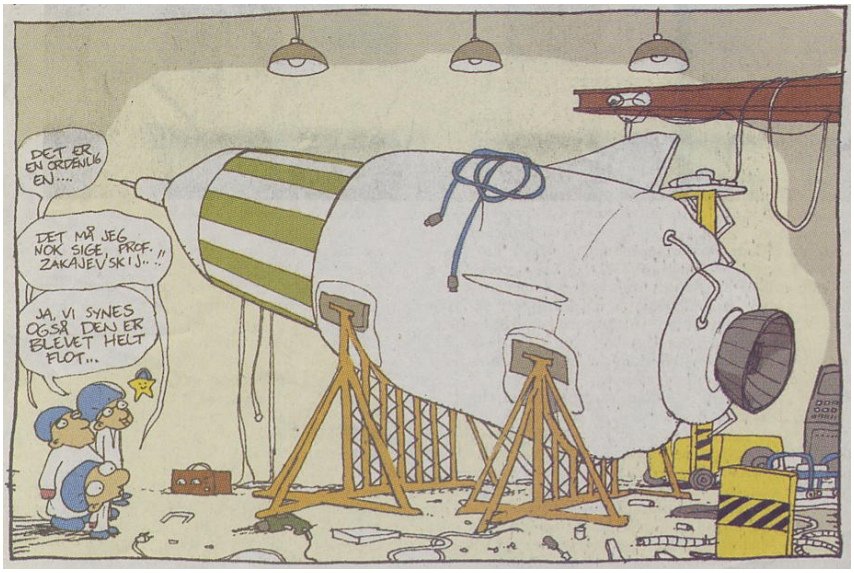
Målet med specialet er ikke at retfærdiggøre rummissionerne på nogen måde, men vi vil indledningsvist bringe et bud på en retfærdiggørelse givet af en forsker fra JPL, William Folkner. Ifølge Folkner er det ikke kun en afklaring af, om vi er alene i universet, der driver os til de omkringliggende planeter i en søgen efter liv. T. S. Eliot citatet i rapportens begyndelse er en tilkendegivelse af, at vi er nødt til at skue udad for at kunne opdage vores egen plads i universet. William Folkners bud på missionernes formål kan opfattes som en naturvidenskabelig forskers formulering af samme princip.

> Suppose you have only one type of writing and that's the only thing you've ever seen written. It would be hard to imagine say computer disc encodings – it's just totally foreign to you. And so we have one type of DNA on our planet. If we could find a second one to compare it to that would be tremendously powerful, right? So there is a big science payoff if we could find a different type of life. (Folkner; 2005a, side 68 i INTERVIEWS)

Ovenstående udlæg synes generelt at være accepteret i medierne, som sjældent problematiserer missionerne. Allerede før opsendelserne fandt sted i juni og juli 2003 var Spirit og Opportunity på en vis måde 'mediestjerner'. De optrådte ofte i dagbladene og i TV, og da de først begyndte deres færden på Mars' overflade og kontinuerligt sendte billeder af fascinerende landskaber til Jorden, blev mediernes bevågenhed overfor missionen kun yderligere skærpet. Man skulle nu tro, at dette billede ville have ændret sig i 2005, og at MER på nuværende tidspunkt ville være henlagt som 'en gammel nyhed'. Det lader imidlertid ikke til at være tilfældet. MER og dermed Mars har i løbet af 2005 været forholdsvis ofte i medierne. Diskussionen af Mars-missionerne er ikke begrænset til videnskabelige tidsskrifter, men foregår i høj grad også i fora, der er tilgængelige for den brede befolkning. Som eksempel kan nævnes DSBs *Ud & Se* (maj, 2005), som bragte en artikel med titlen: 'Til Mars fra Danmark'. I artiklen blev den sædvanlige spin-off tese bragt:

> Mennesket skal tilbage til Månen og helt ud til Mars. I rummet venter prestige og højteknologiske kvantespring for landene med de bedste ideer. Danmark kan komme med derud, blandt andet fordi vi ved noget om havstrømme og is. (Koldby i *Ud & Se*, maj, 2005)

Jacob Martin Strids tegneserie 'Strid' er et andet eksempel på, at Mars-missionerne er noget, man kan tage op uden for videnskabelige kredse.



Figur 1.2 Strids M2M (dag 7) fra Politiken den 9. april, 2005. Strid, Flemming Fadøl, professor Zakajevskij og Lille Stjerne beskuer rumfartøjet. Bemærk det klare fokus på teknologien. (Striben er bragt med Strids tilladelse.)

Den meget populære tegneserie handlede i længere tid om en *Mission To Mars* (M2M), som dog aldrig nåede Mars, fordi Strid opsagde sit job på Politiken, men Mars-missionen kom alligevel ind i de danske hjem gennem Strids forberedelser til rumrejsen mod den røde planet (se figur 1.2).

1.2 Inspirationskilder

Fokus i medierne er oftest på de teknologiske aspekter af MER-roverne, de flotte billeder og ikke mindst de geologiske samt biologiske resultater, missionen løbende har givet. At der skulle være matematik involveret i missionerne nævnes så godt som aldrig. Vores interesse for MER blev vakt, da vi læste en artikel skrevet af matematikeren Philip Davis om faget matematiks natur. I artiklen stod der:

> Consider the recent flight to Mars that put a 'laboratory vehicle' on that planet. Whether or not that excites you, you must admit that it is an almost unbelievable technological accomplishment. Now, from start to finish, the Mars shot would have been impossible without a tremendous underlay of mathematics built into chips and software. It would defy the most knowledgeable historian of mathematics to discover and describe all the mathematics that was involved. The public is hardly aware of this; it is not written up in the newspapers. (Davis; 2004, side 23)

På det tidspunkt, vores speciale blev påbegyndt, var missionen faktisk formelt set afsluttet, og roverne havde allerede transmitteret billeder i adskillige måneder. Disse billeder havde selvfølgelig også gjort et vist

indtryk på os.

Vi var nok klar over, at vi ikke var de mest vidende matematikhistorikere, men alligevel syntes en undersøgelse af en del af matematikken i MER at virke interessant.

I specialets indledende fase arbejdede vi på at få et overblik over den matematik, der var involveret i missionen, og vi faldt over to eksempler, som vi mente ville være interessante at undersøge nærmere, nemlig *kildekodning af billeder* og *kanalkodning*.

Kilde- og kanalkodning[3] er to matematiske discipliner, som indgår i den teknologi, der anvendes til kommunikation mellem Mars og Jorden. Personligt syntes vi, at disse discipliner var interessante, fordi de har visse forbindelser til datalogien, som vi begge studerer ved siden af matematikken. Kilde- og kanalkodning kom derfor til at være hovedområderne for undersøgelsen.

Under arbejdet med specialet fandt vi hurtigt ud af, at undersøgelsen ville blive mest interessant og i et vist omfang grundet mangel på litteratur kun var mulig, hvis vi kunne få lov at besøge JPL og interviewe forskere, som havde været en del af missionen. Inden vores besøg på JPL tog vi 'et smut' omkring Brown University i Providence for at tale med Philip Davis og lade os inspirere til vores efterfølgende interviews på JPL. Blandt andet talte vi med Davis om matematikken i dagligdagen og i vores samfund generelt.

> It's a wonderful subject mathematics of course and the interesting thing is that it is coming into our lives more and more. The age is the mathematical age. Most of the mathematics is hidden, it's invisible to people because it's in programs, it's in chips, it's in laws... So you don't see it and if you don't see it you don't think it is there. (Davis; 2005, side 17 i INTERVIEWS)

Skjultheden af matematikken kunne vi nikke genkendende til, og det var et aspekt, som vi fandt virkeligt interessant. Vi besluttede os derfor for at lade en del af vores undersøgelse handle om den skjulte matematik i MER. Med på vejen fik vi af Davis følgende råd i forhold til undersøgelsen:

> I would say, try to be a little bit like journalists rather than teachers of abstracts mathematics. Look at it this way: when you get through with all the pages there may be ten pages, rearranged and so on, that you could even send to the newspapers for general interest – I think you could make an interesting story there. (Davis; 2005, side 13 i INTERVIEWS)

Vi diskuterede videre med Davis, hvordan denne journalistiske tilgang kunne bidrage med noget særligt til vores speciale. Han foreslog os, at vi skulle forsøge at dreje en del af undersøgelsen i en mere 'menneskelig' retning.

> I think you want to go also for what you might call the human side of the story. When you interview these people try to get something about their background, what were their experiences before they came to the JPL, what their training is, whether

3 I rapporten vil vi også benytte betegnelserne data-/billedkomprimering og fejlkorrigerende koder.

> it's in computer science, mathematics, physics, whatever. [...] Try to find out what each one does, how much independence they have, how much opportunity is there for them to develop new ideas or new things whether it's in mathematics or whether it's in writing algorithms, software and that kind of thing. I think you want to make a story here of what it means to work in a space program as a mathematician or a computer scientist or whatever these people are. (Davis; 2005, side 13 i INTERVIEWS)

Vi har i nogen grad taget Philip Davis' råd til os, hvilket vi skal vende tilbage til i beskrivelsen af vores metode.

1.3 Videnskabelig motivation

En rummission udgør et interessant udgangspunkt for et videnskabeligt studie, fordi missionerne til det ydre rum står som et af de største videnskabelige ikoner i vores samfund. Matematikkens plads i sådanne store projekter er interessant, fordi den i et vist omfang afspejler matematikkens plads i vores samfund generelt.

I forhold til en langsigtet plan for udforskning af rummet er Mars det 'næste skridt'. Dette skyldes dels, at planeten er tættest på Jorden, men også, at den ligner Jorden så meget, at der er grund til at tro, at der her muligvis er eller har været liv. Planlægningen af bemandede missioner til Mars er også i fuld gang.

En videnskabelig undersøgelse af en Mars-mission er altså interessant, fordi den kan give et billede af, hvad der foregår 'bag kulisserne' i et af de mest aktuelle videnskabelige projekter anno 2005.

Som tidligere omtalt opfattes Mars-missionerne ofte som en stor teknologisk bedrift, mens matematikken sjældent nævnes. Et kig bag kulisserne kunne derfor indebære at se på den matematik, som rent faktisk ligger til grund for en del af teknologien. Af denne grund mener vi, at kilde- og kanalkodning er helt oplagte valg, da disse discipliner netop er indlejrede i Mars-missionens kommunikationsteknologi.

Viden om matematikken bag kommunikationsteknologien mener vi desuden er interessant grundet denne teknologis totale gennemsyring af vores samfund. Valget er derfor også truffet med det sigte, at vi som studerende får så stort et fagligt udbytte af undersøgelsen som muligt.

En undersøgelse af matematikken i en Mars-mission, mener vi, er et godt udgangspunkt for et speciale af typen forskervariant. I IMFUFAs studievejledning[4] står der om denne type speciale:

> I *forskervarianten* skal man beskæftige sig med forskningsfaget matematik. I projektarbejdet kan indgå egen matematisk forskning, men det vil formentlig kun ske sjældent. Det hyppigste vil være at arbejdet baserer sig på rapporter fra en nuværende eller fortidig »forskningsfront«. Projektet kan gives en videnskabsteoretisk eller videnskabshistorisk drejning.

Specialet tager netop udgangspunkt i en sådan nutidig forskningsfront.

4 Se eventuelt http://mmf.ruc.dk/mat/main.pdf

1.4 Problemformulering

Først og fremmest vil vi benytte vores undersøgelse af kilde- og kanalkodning i MER til at klarlægge, hvilke problemstillinger i missionen, disse to discipliner bidrager til løsningen af og hvor meget, de egentlig betyder for missionen. Vi vil også forsøge at afklare, hvorvidt MER har påvirket de eksisterende matematiske teorier for kilde- og kanalkodning.

Med udgangspunkt i vores undersøgelse vil vi endvidere forsøge at afklare, i hvilket omfang Mars-missionerne fra 1990'erne og frem har været og er en drivkraft for matematisk grundforskning generelt. Vi skal referere til disse missioner som de *aktuelle* Mars-missioner.

Problemformulering

Ovenstående betragtninger har udmøntet sig i følgende problemformulering for specialet:

1. Hvilken funktion og betydning har kilde- og kanalkodning i MER?
2. Har de aktuelle Mars-missioner nogen betydning for udviklingen af kilde- og kanalkodning og i så fald hvilken?
3. Er matematikken i de aktuelle Mars-missioner udviklet af JPL til lejligheden, er den synlig for JPLs ansatte, og bidrager de aktuelle Mars-missioner til matematisk grundforskning?

1.4.1 Forventninger om besvarelsen

Vi vil her kort beskrive vores forestillinger om, hvordan problemformuleringen kan besvares.

Det første spørgsmål vil vi forstå på et praktisk niveau, det vil sige, vi vil tage udgangspunkt i konkrete problemer i missionen og se på, hvordan kilde- og kanalkodning afhjælper disse problemer. På den måde kan vi beskrive kodernes funktion i missionen. Desuden vil vi forsøge at vurdere, om de problemer, der afhjælpes af kilde- og kanalkodning, er centrale problemer for missionen eller i småtingsafdelingen. Hermed kan vi besvare spørgsmålet om, hvor betydningsfulde teorierne er.

For spørgsmål to vil vi for det første forsøge kort at beskrive den vigtigste udvikling inden for kilde- og kanalkodning i de senere år. Dernæst vil vi se på, om de aktuelle Mars-missioner har haft nogen indflydelse på denne udvikling. Spørgsmålet forventer vi at kunne besvare kort med et ja eller et nej.

Det sidste spørgsmål indeholder tre delspørgsmål, for hvilke besvarelserne i et vist omfang er afhængige af hinanden. Hvis svaret på det første delspørgsmål er et ja, vil det også være ja for de to sidste delspørgsmål. Hvis svaret derimod er nej, er de to andre delspørgsmål stadig åbne. Det sidste delspørgsmål kan vi kun med sikkerhed besvare, hvis vi kan påpege matematik i Mars-missionerne, der er udviklet til lejligheden. En fuldstændig afvisning af et bidrag til grundforskningen fra missionernes side kan vi ikke med rimelighed give. Hvis vores undersøgelse skulle pege på et nej til dette delspørgsmål, må vi altså forsøge at vurdere rimeligheden af et sådant svar.

1.5 Metode

Besvarelsen af problemformuleringen skal ske dels med udgangspunkt i en undersøgelse af selve MER-missionen og dens problemstillinger og dels i en undersøgelse af kilde- og kanalkodning. Undersøgelsen har ikke fokus rettet på den teknologiske side af kilde- og kanalkodning, men på den matematiske side.

Selve MER-missionen kunne vi ikke undersøge gennem et litteraturstudie, fordi missionen er så ny, at der ikke findes litteratur om den, som var brugbar for os. Vi har derfor valgt i et vist omfang at lade os inspirere af Philip Davis' ide om at bedrive matematik-journalistik og foretage undersøgelsen af MER gennem forskellige interviews. Vi har foretaget disse ved henholdsvis DTU, MIT og JPL. På sidstnævnte institution har vi interviewet nogle af de forskere, der har arbejdet med missionen. Vi har ikke benyttet os af alle Davis' ideer, men brugt vores interviews til at afdække forhold på JPL såvel som i MER, som vi fandt interessante:

- Identificering af flaskehalsproblemer i MER-missionen.
- Det daglige matematiske arbejde for ansatte ved JPL.
- Muligheden for at bedrive grundforskning på JPL.
- Generelle karakteristika for matematik i missionen.

Første spørgsmål i problemformuleringen vil vi besvare ved at identificere flaskehalsproblemer i missionen og afgøre, om kodningsteorierne er med til at løse disse. Teorierne for kilde- og kanalkodning vil vi ikke udforske i alle kroge, men derimod rette fokus mod de koder og problemstillinger i teorierne, som er interessante for rumanvendelser.

Andet spørgsmål vil vi besvare dels ved overordnet at identificere den seneste udvikling inden for kilde- og kanalkodning og dels ved at se på det daglige matematiske arbejde, der bedrives på JPL.

Vores undersøgelse af generelle træk ved matematikken i missionen skal benyttes til for det første at illustrere at kilde- og kanalkodning i en vis forstand er eksemplariske for missionen forstået på den måde, at andre matematiske områder i missionen antyder, at lignende forhold gør sig gældende. Den lidt bredere undersøgelse, mener vi, vil bidrage til bedre at rodfæste diskussionen af sidste spørgsmål i problemstillingen.

Ingen af os har en journalistisk baggrund, og vores tilgang til det journalistiske er derfor nok en smule naiv. Som garanti for at vi ikke har fordrejet udtalelser eller lignende, har vi valgt at vedlægge samtlige interviews i transskriberet form som bilag. Alle interviews er naturligvis optaget med de involveredes samtykke.

1.6 Forskningsmæssige bidrag

En del af specialets forskningsmæssige bidrag er vores fremstilling af den kilde- og kanalkodning, som er i anvendelse i MER. Detaljerne i denne kodning har vi i høj grad selv måttet studere i de originale kilder, som JPL-dokumenterne refererer til. Om arbejdet med at opspore og formidle de detaljer, som er skjult i JPLs dokumenter, er *egentlig* forskning kan muligvis diskuteres. Arbejdet med fremstillingen har gjort det nødvendigt at arbejde med problemstillinger, som grænser op til ingeniørfagene, hvilket til tider har været lidt af en udfordring.

En anden del af det forskningsmæssige bidrag er vores interviews, som giver et indblik i arbejdet på en institution, der generelt er meget lukket for omverdenen. På den måde tilvejebringer vores interviews informationer, som i høj grad er svært tilgængelige, herunder informationer inden for et videnskabssociologisk område, som ikke er beskrevet i litteraturen.

De konklusioner, som specialet indeholder, mener vi, kan være relevante for alle med generel interesse for, hvordan matematik optræder og indgår i større forskningsprojekter. Mars-missionernes ikon-status, mener vi kun, er med til at gøre nærværende specialerapports undersøgelse mere interessant.

Specialets undersøgelse er speciel, fordi den tager sit udgangspunkt i et samtidigt forskningsprojekt. Undersøgelsen kan i nogen grad sammenlignes med en 'historisk undersøgelse'. Blot er der den væsentlige forskel, at undersøgelsens udgangspunkt forefindes kun et par år tilbage i historien. Af denne grund kan specialet ligeledes have interesse for personer, der studerer anvendt matematik i vores øjeblikkelige samfund.

1.7 Om rapporten

Specialet er delt i tre delrapporter kaldet Hovedrapporten, Indførelsen og Interviews.

Hovedrapporten indeholder specialets undersøgelse og besvarelse af problemformuleringen. I kapitel 2 findes den nævnte fremstilling af MER og af JPL herunder beskrivelserne af institutionens ansatte og deres forskning. Herefter følger rapportens undersøgelser af de to eksempler. Kapitel 3 indeholder en kort overordnet beskrivelse af nogen af de teoretiske problemstillinger, som kilde- og kanalkodning skal afhjælpe. Beskrivelsen tager udgangspunkt i Shannons matematiske teori for kommunikation. Kapitel 4 udgør et kortfattet sammendrag af den grundlæggende teori for kilde- og kanalkodning, som beskrives i Indførelsen. Herefter følger beskrivelsen af MERs kilde- og kanalkodning i kapitel 5 og kapitel 6. Diskussionen af de spørgsmål, som er givet i problemformuleringen, findes i kapitel 7. Specialerapportens konklusion følger i kapitel 8. Kapitel 9 udgør rapportens efterskrift, som indeholder enkelte af de spørgsmål og svar, som blev fravalgt i den endelige rapport. Tilsidst følger rapportens litteraturliste og appendiks A, som indeholder en beskrivelse af NASAs gennemførte Mars-missioner samt en opremsning af billedkomprimering og kanalkodning i disse.

Indførelsen indeholder den grundlæggende teori for forståelsen af MERs billedkomprimering og kanalkodning. Førstnævnte beskrives i del I og sidstnævnte i del II.

Samtlige transskriberede interviews findes i delrapporten Interviews, som indeholder fire kapitler. Et med interviews fra DTU, et med Davis-interviewet fra Brown University, et med interviews fra MIT og endelig et med interviews fra JPL. I vores interviews er der en del snak om 'the stagnation of science'. Dette aspekt var oprindeligt tiltænkt en central rolle, men blev på et sent tidspunkt fravalgt specialets problemstilling og figurer derfor kun i vores interviews og i efterskriftet.

2 JPL og MER

> The complexity of the space problem is keeping it simple. (san Martin & Folkner; 2005, side 58 i INTERVIEWS)

I dette kapitel skal vi fortælle vores historie om JPL og MER. En historie, der dels handler om det arbejde, der foregår i dagligdagen på JPL, og dels handler om de særlige udfordringer, der er forbundet med en mission som MER. Vi har så vidt muligt forsøgt at illustrere arbejdet med udgangspunkt i små konkrete historier fra vores interviews. Disse historier mener vi er eksemplariske for arbejdet generelt.

Kapitlet skal være en del af grundlaget for at besvare spørgsmålet om, hvorvidt JPL bidrager til matematisk grundforskning. Derudover skal kapitlet være med til at nuancere begrebet 'den skjulte matematik'.

2.1 Jet Propulsion Laboratory

NASAs forskningsinstitution *Jet Propulsion Laboratory* er placeret et afsidesliggende sted på vej op i bjergene, omgivet af skov lidt nord for Los Angeles' forstad Pasadena. Efter en halv times kørsel fra Pasadena Motor Inn i downtown Pasadena når man frem til områdets indkørsel, hvor man hver dag mødes af uniformerede vagter, som undersøger bilen og hører, om man har en konkret aftale på stedet. Som udefra kommende får man det førstehåndsindtryk, at indkørslen er en indgang til en 'hemmelig verden'. Dette eventyrlige indtryk af stedet forstærkes yderligere, når man indfinder sig på *JPL's Visitor Center* og får at vide, at man under ingen omstændigheder må bevæge sig udendørs på området uden en passende

Figur 2.1 Jet Propulsion Laboratory i Pasadena, Californien. Til venstre: JPL set fra oven. http://ipac.jpl.nasa.gov/media_images/jpl_small.jpg Til højre: En af JPLs hovedbygninger. http://www2.jpl.nasa.gov/files//images/browse/jpl18161ac.gif

eskorte. Inde i bygningerne er der større frihed, og »man kan sagtens få lov til at gå på toilettet selv«. Da vi ankom første dag blev vi hilst velkommen af en dame i receptionen ved navn 'Bobby'. Mængden af besøgende på JPL må være forholdsvist begrænset, for allerede dagen efter kunne hun genkende os. Da vi ankom første dag, informerede hun os om sikkerhedsprocedurerne på stedet og ringede dernæst til dr. William Folkner for at meddele vores ankomst. Folkner dukkede op efter et par minutter og tog os med rundt på JPLs arealer. I løbet af den uge, vi tilbragte på stedet, mødte vi en håndfuld af JPLs ansatte, som vi i det følgende skal forsøge at tegne et billede af.

2.1.1 JPLs mennesker

William Folkner var den person på stedet, vi havde mest kontakt med, og, som det også fremgår af vores interviews, den person vi fik flest muligheder for at interviewe. Under disse interviews fik vi lejlighed til at spørge ham om hans karriere og om hans billede af de mennesker, der arbejder ved JPL. Folkner fortalte os, at han selv havde taget en ph.d. inden for fysik fra University of Maryland. Derefter besluttede han at søge ansættelse ved JPL for at være en del af den planetare udforskning. Han har været på stedet siden 1988. Vi spurgte ham, om han mente, at hans vej til JPL og efterfølgende karriere på stedet var typisk.

> Most of the people I know has been here a long time or tend to be here a long time. There aren't many people who come here who just want to be here a year or two and then go away again. People who want to do space work tend to want to do space work. I think this is the best place in the world to do space work. The people here are all very, very good. They are all very dedicated and they want to make these things fly. There are other space institutes around the world but I don't think any of them has the breadth of experiments going on as we do. (Folkner; 2005b, side 76 i Interviews)

Motivationen for at arbejde på JPL synes i høj grad at stamme fra en fascination af selve missionerne. Dette billede blev forstærket hver gang vi interviewede en ny person. Mark Maimone var en af disse personer. Selv er han uddannet ingeniør, som færdiggjorde en ph.d. i Pittsburgh, Pennsylvania efterfulgt af en post doc inden for *space robotics*, inden han fik en stilling ved JPL i 1998. Maimone, som er involveret i udviklingen af rovernes styring, forklarede:

> When I was a kid I saw the Viking missions on Mars and I thought that would be pretty neat, hunched over the terminal looking at these pictures that nobody else would see for a year before it got published. But of course now every picture we take gets published on the Internet the next day so it's not much of an advantage but it's still nice. (Maimone; 2005, side 85 i Interviews)

Mark Maimone var overbevist om, at han ville blive på JPL, så længe det var muligt. Også Matthew Klimesh, som er en af udviklerne bag ICER, ønskede at blive på JPL. Han er uddannet ingeniør, som fik en ph.d.-grad

fra et universitet i Michigan. Efterfølgende var han ansat dér i et år som post doc, inden han kom til JPL. Han fortalte, at han ikke er typen, der rejser meget rundt, men var kommet til JPL for at blive der. Han fortalte i øvrigt, at mange ansatte bliver på JPL i lang tid, og at en del af dem når at opleve 50 års jubilæum, inden de trækker sig tilbage. (Klimesh; 2005, side 46 i INTERVIEWS)

Den anden hovedperson bag ICER, Aaron Kiely, var ligeledes kommet til JPL for at arbejde med roverne. Ligesom Klimesh havde han en universitetsuddannelse fra Michigan med en ph.d.-grad bag sig. Arbejdet på JPL var hans første efter endt universitetsstudie. Han beskrev ligeledes JPL som et sted, man kommer for at blive. Det tiltrækkende ved JPL er således ikke de økonomiske aspekter ved stillingerne, men arbejdets natur. (Kiely; 2005, side 44 i INTERVIEWS)

Den overordnede karakteristik af JPLs forskere synes at være folk med det højeste uddannelsesniveau, som kommer til JPL næsten umiddelbart efter afslutningen af deres universitetsstudie. De er alle drevet af muligheden for at arbejde med roverne og den planetare udforskning og er i ringe grad interesserede i økonomiske aspekter af deres ansættelser. Alle de personer, vi talte med, ønskede at blive på JPL så længe som muligt.

2.1.2 Arbejdet på JPL

Under vores besøg ved JPL fik vi desværre ikke mulighed for at observere stedets forskere i en arbejdssituation. Der var formentlig flere årsager til dette, men alt det sikkerhedsmæssige besvær, der var forbundet med at eskortere os rundt, var højst sandsynligt hovedårsagen. Selvom vi ikke fik oplevet JPL på den måde, mener vi alligevel, at vi gennem vores interviews har fået et indblik i det arbejde, der foregår på stedet. Dette indblik skal vi forsøge at videregive her.

Der foregår naturligvis en utrolig mængde arbejde, som involverer konstruktion af videnskabeligt udstyr og manuel afprøvning af samme, men dette arbejde vil vi i store træk ignorere og derimod koncentrere os om det videnskabelige.

En af de første personer, som vi talte med om arbejdets natur, var Jacob Matijevic, som er uddannet matematiker og har været på JPL i en årrække. Han fortalte følgende om matematikken i det daglige arbejde:

> When I first arrived here over twenty years ago there were still efforts to hand-implement certain mathematical models for certain applications. And there were specialist applications here for specialists in the applied mathematical sciences who worked here to make those applications possible. But over time much of that has been incorporated in fairly standard and available simulation and modelling packages – computer packages. Expansions have been introduced slowly over time to these packages and that's basically how the engineers here do their job. Instead of going back to first principles they apply these tools... the foundation theories are from the eighteenth century to a large degree. (Matijevic; 2005, side 50 i INTERVIEWS)

En stor del af matematikken i JPLs dagligdag er altså indbygget i

softwarepakker og ikke direkte synlig for de ansatte.

En mission som MER handler i høj grad om at være i stand til at forudsige diverse begivenheder. Når fartøjet flyver oppe i rummet, er det umuligt at foretage justeringer, som kræver mere end et radiosignal, og alting skal derfor fungere som forventet.

Som eksempel kan nævnes Mars-miljøets påvirkning af instrumenterne om bord på Spirit og Opportunity. Man er nødt til at have meget præcis viden om, hvordan varme og kulde fordeler sig i roverens indre og påvirker instrumenterne. For at opnå en sådan viden bygges virtuelle modeller af roverne i software, som kan simulere termiske forhold. Sådanne termiske modeller er typisk baseret på en række differentialligninger, som løses i programmerne. Arbejdet for den ansatte på JPL består i at bygge den virtuelle model af roveren. Den konkrete løsningsmetode, som programmet implementerer er underordnet så længe den virker og ikke er alt for langsommelig.

For at kunne forudsige den konkrete påvirkning af instrumenterne i de nævnte modeller, er man også nødt til at have modeller for, hvordan miljøet afhænger af årstiderne på Mars. Sådanne modeller baseres dels på data fra de forskellige orbitere og dels på konkrete målinger foretaget på overfladen. Korrektheden af sidstnævnte målinger afhænger i høj grad af, hvor god beskrivelsen af instrumenternes opførsel i Mars-miljøet er, og kan altså ikke garanteres. Man kan dog sammenligne sine orbiter-data med de konkrete målinger og på denne måde modificere sine modeller, så disse langsomt bliver bedre og bedre. Alt dette foregår i software. Med hensyn til modellerne af årstidernes påvirkning af Mars-miljøet er det nok ikke urimeligt at sammenligne JPLs arbejde med arbejde, der foretages på et meterologisk institut.

Der foregår altså en masse arbejde, som involverer modellering og simulering, men alt dette foregår i softwarepakker. Det kunne give én grund til at tro, at der sidder matematikere på JPL og udvikler sådanne pakker, men Matijevic fortalte, at pakkerne for det meste stammer fra kommercielle firmaer.

Under vores samtale med William Folkner og Miguel san Martin (en anden ingeniør fra JPL) berørte vi spørgsmålet om, hvordan man holder styr på rovernes position og orientering på Mars' overflade. san Martin karakteriserede dette arbejde således:

> The surface [...] is also very basic and we are using tricks that, you know, sailors have been using for centuries. We look at the Sun. (san Martin & Folkner; 2005, side 57 i Interviews)

Sammen med en gravitationsvektor, som kan måles af roveren, giver Solens position på himlen i store træk den information, der er nødvendig for at kunne navigere såvel i rummet som på Mars. Den matematik, der er involveret, er ret simpel. san Martin sammenfattede det således:

> So that's it. Not much. The most important is that you have millions of these little, simple things. And that's the trick; to make them all work, and talk to each other and make sure that no parameter is tightened too much or too little. The complexity of the space problem is keeping it simple. (san Martin & Folkner; 2005, side 58 i Interviews)

William Folkner, der også var tilstede på dette tidspunkt, svarede følgende:

> Well, you hit a lot of mathematics in your descriptions, right, because you need to know the positions of the axes of Mars around the Sun as a function of time and you need to know what the orbits around the Sun and the Earth were. There is a lot of mathematics hidden in what you just said. [...]
> They are hidden from us. We've worked all that out for us in the tables. So to know where the Sun is now, you just look it up. Somebody had to figure it out the first time. (san Martin & Folkner; 2005, side 58 i Interviews)

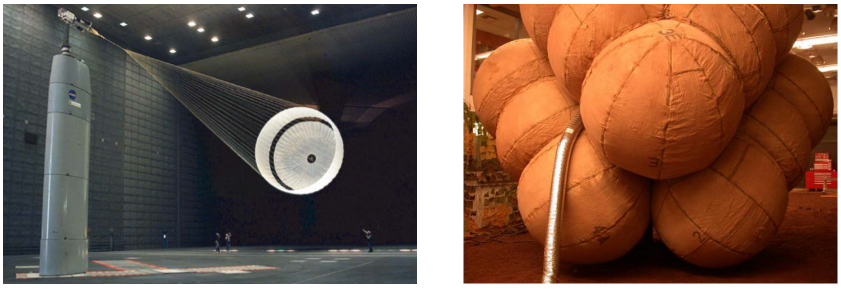
Folkners svar illustrerer, hvorfor spørgsmålet om hvilken matematik, der anvendes i MER, er svært at besvare. På sigt kan viden, som matematik tidligere har banet vejen for, blive en så integreret del af vores opfattelse af verden, at vi ikke længere forbinder den med matematik. Mars' banekurve omkring Solen som funktion af tiden er et udmærket eksempel på dette: Et tabelopslag, som giver Solens position i forhold til Mars til et givet tidspunkt, vil de færreste vel kalde matematik. Udarbejdelsen af tabellen derimod *er* et matematisk problem. Desuden illustrerer samtalen, at JPLs ansatte ikke er så opmærksomme på, hvornår de anvender matematiske redskaber.

Matematikken er altså tilstede i det videnskabelige arbejde på JPL, men den forekommer til tider i form af en masse informationer, der langsomt føjes til vores viden, uden at vi egentlig er klar over matematikkens eksistens i denne proces.

2.1.3 Arbejdets præmisser

Som omtalt i indledningen involverer et projekt som MER undertiden op imod 1.000 mennesker ad gangen. Et så omfangsrigt projekt kræver naturligvis en enorm mængde planlægning og ajourføring mellem forskellige afdelinger og arbejdsgrupper. Desuden kræves en utrolig høj pålidelighed af det arbejde, der udføres. En enkelt fejl i et stykke teknologi eller en algoritme kan få alvorlige konsekvenser og i værste tilfælde resultere i adskillige års spildt arbejde for mange hundrede mennesker. En stor del af det arbejde, der foregår på JPL, er derfor præget af omhyggelig udvikling og afprøvning. Jacob Matijevic fortalte følgende om arbejdet med rovernes faldskærme:

> We did drop tests. We did wind tunnel tests with the parachutes. But even before this time it was through models of the profiles of these devices that we came up with things like what the entry angles would be, what sorts of release points should we be looking at, as well as designing the algorithm that checks for height above the surface and finding out at which time to deploy the parachute and at which time to fire the rockets for slowing the descent. All of this was based on what we expected to be the environmental profile that the vehicle would see as it came down to the surface. So this was all done in simulation. (Matijevic; 2005, side 52 i Interviews)



Figur 2.2 Til venstre: Vindtunneltest af MER-landingsmodulets faldskærm. http://www.nasa.gov/centers/ames/images/content/79641main_picture_2.jpg Til højre: MER-landingsmodulets airbags. http://photojournal.jpl.nasa.gov/jpegMod/PIA04999_modest.jpg

Pålidelighed er således altafgørende for missionerne. Står man og skal vælge mellem to forskellige tilgange til en problemstilling, er man derfor mere tilbøjelig til at vælge en velkendt, gennemprøvet løsning end en ny og måske lidt mere effektiv løsning, som ikke har været forsøgt tidligere. Dette forhold gælder i alle områder af missionen. Matijevic omtalte dette som »steady progress« (Matijevic; 2005, side 54 i INTERVIEWS). Der foregår naturligvis en vis udvikling, men denne udvikling foregår langsomt, og de nye ideer, som bliver introduceret i missionerne vil for det meste være omkring ti år gamle ved opsendelsen, fordi de allerede skal fastlægges under planlægningen af missionen.

Omfanget af projektet betyder også, at de forskellige afdelingers arbejde *skal* færdiggøres til bestemte deadlines. Deadlines virker ikke overraskende som en hæmsko for udviklingen af nye ideer, da det er sværere at overholde deadlines, når man arbejder med noget, man ikke har den store erfaring med. Det er helt utænkeligt at gå igang med et stykke arbejde, hvis man ikke er sikker på, at der er tid til at foretage tilstrækkelige afprøvninger af dette arbejde. William Folkner kommenterede, hvordan de konstante deadlines påvirker sammensætningen af folk, der arbejder på JPL:

> There are five thousand people here. How many post docs are there? I don't know, maybe a hundred and fifty – something like that. We don't have a lot of graduate students and part of that is because almost everything we do here has to be done on a particular time scale. Very little of our budget is spent doing research where if we get the answer next year or the year after that, it doesn't matter. Everything has to be done on a schedule. So it is not a good training environment for graduate students. Graduate students here are not used much, because the people who could supervise them are busy doing other things. (Folkner; 2005b, side 78 i INTERVIEWS)

JPL beskæftiger altså en masse mennesker med helt konkrete, fastlagte arbejdsopgaver, som skal udføres efter et nøje fastlagt skema. Få mennesker er på JPL for at bedrive grundforskning. De tidligere nævnte krav om pålidelighed og rettidighed af det arbejde, der udføres, er naturligvis

en væsentlig grund til dette. Der er dog også et yderligere forhold, som forstærker tendensen til at fravælge det nye og usikre: På JPL er man i nyere tid gået bort fra få store, dyre missioner, og drejet i retning af mange små, men billige missioner. Pathfinder-missionen fra 1996 havde således et samlet budget på kun 265 millioner dollars[1], hvilket sammenlignet med Viking-missionernes budget på 8 milliarder dollars må betegnes som 'low-cost'. MER-missionen har været dyrere end Pathfinder, men stadig ikke i nærheden af Viking-beløbet.

Kan man genbruge noget fra en tidligere mission, kan man spare store mængder af penge og tid, og de mange billige missioner må derfor nødvendigvis benytte en del genbrug fra mission til mission. Genbruget gælder i en vis forstand også de mennesker, som deltager i missionerne. Folkner fortæller:

> A problem in doing the smaller missions is that you can only do them with reasonably experienced people, and because they are small you don't have the budget to have an experienced person train an inexperienced person. So JPL is getting older on the average, because we don't have a big mission to afford enough people to have senior and junior people. So JPL is short of junior people. That is not a problem yet, but in five years or ten years it will be a disaster. The management here knows that and is trying to deal with it but it is not often you fix problems until they occur. They are trying to get ahead of that, but it is a difficult thing. Because we are trying to do so many cheap missions we depend on experienced people and we are not budgeting training inexperienced people. It's probably true throughout NASA and the aerospace industry. (Folkner; 2005b, side 77 i INTERVIEWS)

JPLs økonomisk betingede konservatisme i forhold til ansættelse af ny arbejdskraft må i et vist omfang udgøre en begrænsende faktor for tilførelsen af nye ideer.

Vores beskrivelse af arbejdets mere eller mindre fastlåste natur skal ikke opfattes som et udtryk for, at arbejdet på JPL er kedeligt og rutinepræget. Vores indtryk af arbejdet var nærmest det modsatte. Den fastlåste natur betyder blot, at muligheden for at bedrive matematisk grundforskning er yderst begrænset, ligesom tilførslen af nye matematiske ideer kun foregår i ringe grad.

2.2 Om matematikken i MER

Philip Davis oprindelige ide, som godt nok ikke var møntet på et speciale, gik ud på at foretage en afdækning af den matematik, som var involveret i MER. Vi har ikke foretaget en sådan afdækning, men vi vil alligevel på et overordnet plan beskrive, hvad vi lærte om matematikken bag MER, mens vi var på JPL.

Vi talte med en håndfuld forskellige mennesker om diverse forskellige problemstillinger i missionen, og de beretninger, vi fik, antyder en række fælles træk for problemstillingerne.

1 http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1996-068A

Miguel san Martin fortalte os om flere forskellige matematiske problemstillinger, som indgår i missionerne. Et problem, som altid er interessant for de planetare missioner, er *Lost in Space*-problemet. Problemet indgår som fast element i transitfasen, i hvilken størstedelen af rejsen fra Jorden til Mars foregår. Problemet består i at finde ud af, hvor i rummet man er ved at se på stjernerne. Hovedproblemet består sådan set i at identificere de stjerner, man ser på. Det viser sig, at denne identificering ikke er et trivielt problem. Man kan ifølge san Martin ikke identificere en stjerne alene ved dens lysstyrke eller placering relativt til andre uidentificerede stjerner. Man er nødt til at se på de mønstre, som stjernerne danner. (san Martin & Folkner; 2005, side 55 i INTERVIEWS) san Martin fortalte:

> ... the way we do it, again different people do it different ways, you can start playing with star patterns. It's a pattern recognition problem. The way we do it is using the Sun. That gives us two axes. [...] People have been playing with variations of this since the '60s. Some versions are more clever than others. This one for instance cheated because we used the Sun. 'I'm in three dimensions and I don't know where I am. By looking at the Sun I know two dimensions.' Without using the Sun, this problem is called the 'Lost in Space' problem. (san Martin & Folkner; 2005, side 55 i INTERVIEWS)

Problemstillingen, der forekommer i MER, er altså en forsimplet udgave af Lost in Space problemet. Dette problem blev formuleret og løst allerede i tresserne, da rummisionerne begyndte. Lost in Space problemet kræver, at man foretager en del målinger fra rumfartøjet. Dette giver anledning til en ny problemstilling.

> In the cruise we are spinning the spacecraft. We use conservation of angular momentum to keep our spacecraft from turning. The most sophisticated piece of software that we have onboard during that time is a Kalman filter. A Kalman filter is a statistical framework, or algorithm perhaps, to mix information from different sensors. So you have a dynamic model – in this case it's a simple rigid body, which you represent mathematically and then you have some sensors which measure where the Sun is. And you have a statistical noisy model. (san Martin & Folkner; 2005, side 54 i INTERVIEWS)

Kalmanfilteret er en algoritme, som gør det muligt at kombinere data fra forskellige sensorer. Data fra disse sensorer vil ofte være ufuldstændige, afhængige af måleudstyrets opførsel i den konkrete situation eller måske påvirket af støj. Kalmanfilteret giver et estimat af, hvordan målingerne ville have set ud uden alle disse forstyrrende elementer. san Martin kommenterede brugen af filtret:

> This formulation is pretty common. There are several books, you know. It was invented in the '60s ... You can come up with a sequential filter that allows you to optimally combine the information from three things; your dynamic model, your sensors and your star tracking into optimal information about your attitude [position of spacecraft relative to a frame of re-

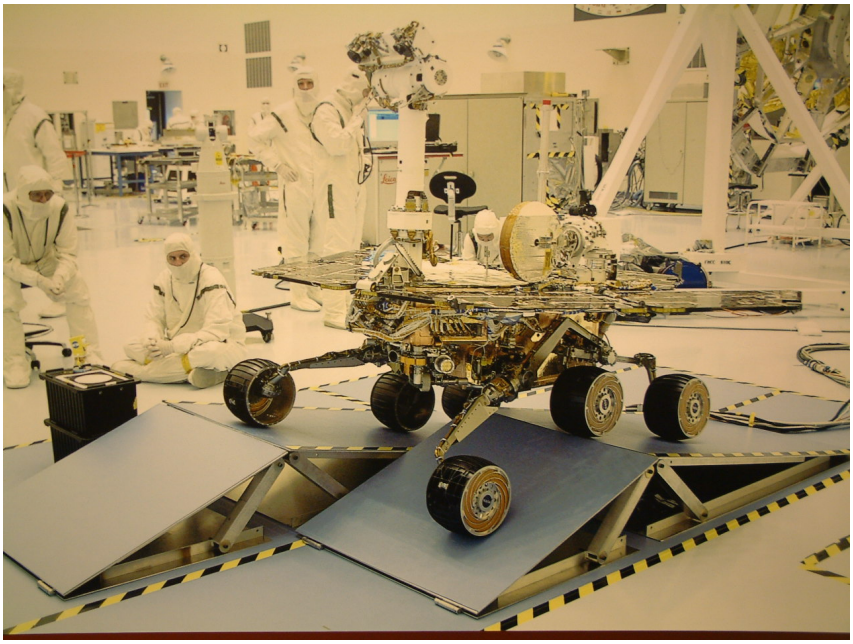
> ference] and the inertial properties of your plant. So that is going on all the time. It's a well known aerospace industry. (san Martin & Folkner; 2005, side 55 i INTERVIEWS)

Kalmanfiltreringen udgør altså ikke et stykke matematik, der er udviklet på JPL til missionen, men er en del af et problem, der allerede var løst i tresserne.

Under vores interviews med William Folkner diskuterede vi blandt andet udfordringen i at bestemme den rette banekurve for rejsen. Marsmissionerne gør altid brug af den samme banekurve, som kaldes *Hohmann*-banekurven. For to legemer (Mars og Jorden), der kredser om et andet legeme (Solen) er Hohmann-banekurven den løsning på banekurveproblemet, som kræver den mindste energi ved opsendelsen. Fordi Mars' og Jordens bevægelser omkring solen ikke ligger i samme plan, fås i praksis to løsninger (type 1 og 2), som kræver den samme energi. Vi spurgte Folkner, om banekurveproblemet er en standardøvelse:

> Yes, for Mars it is a very standard exercise. For the other planets it tends to be more complicated because you'll trade flybys of other planets for angular momentum against the mission operations time. [...] For going to Mars it always comes down to: 'Do you want to do type 1 or type 2?' (Folkner; 2005b, side 71 i INTERVIEWS)

Banekurven til Mars er altså et eksempel på et matematisk problem, som er løst, og som blot skal pilles ned fra hylden fra mission til mission.



Figur 2.3 Test af MER-roverens køreegenskaber. (Billedet stammer fra JPL.)

Hohmann-banekurven stammer fra 1925[2].

De matematiske problemer og begreber, som vi har beskrevet ovenfor, er forholdsvist isolerede. Hvis man skal nævne en større samlet teori med en væsentlig betydning for MER, ville *kontrolteori* være passende. Vi spurgte Mark Maimone om den mængde kontrolteori, der indgår i styringen af roverne:

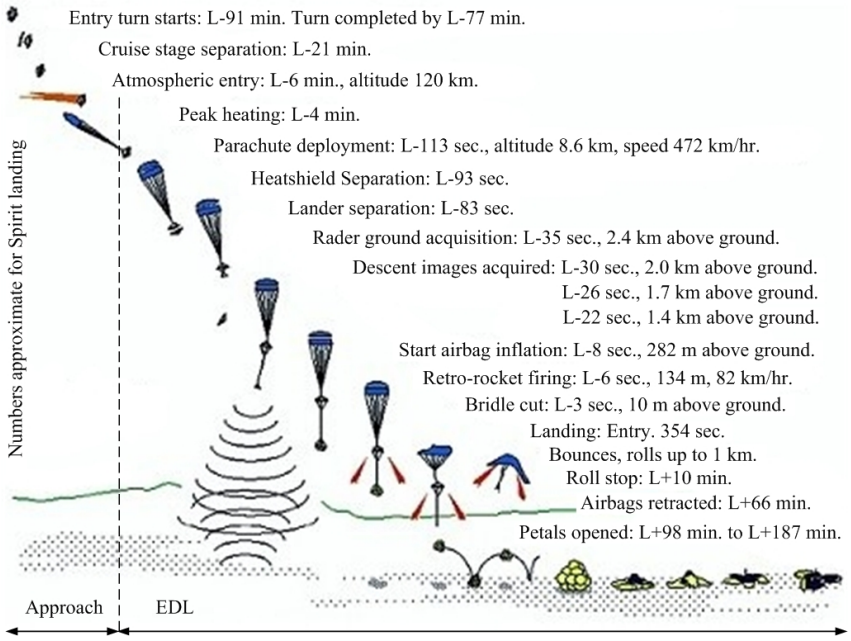
> We have to drive the wheels so there is software that controls the motors there and all the instruments have motors and have to be controlled so there is some amount of control theory being applied there. I don't know what has been published about the details of those algorithms so I can't tell you anything about that. But there are a lot of motors, there are motors that drive the wheels, that spin the cameras, that control the arm and when we landed we had a lot of motors that were used simply to get up off the lander; stand up, spread out the wheels, spread out the legs, open up the solar panels, pull up the mast and do all those things. (Maimone; 2005, side 81 i INTERVIEWS)

Ovenstående udtalelse fra Mark Maimone illustrerer de besværligheder, der kan være forbundet med at foretage en undersøgelse på et sted som JPL. Vi oplevede flere gange[3] at de uskyldige spørgsmål, vi stillede, overskred en grænse for klassificering, hvorfor de blev besvaret som ovenfor. Fra Jacob Matijevic kunne vi dog forstå, at styringen måske ikke var så kompleks matematisk set:

> Mainly what we're taking advantage of to at least create the driving pattern, is the fact that each of the individual six wheels is controllable at least for steering and so we can create in essence any kind of arc condition. Because of the individual control we can modulate the speed of the motor turns. That gives us a means for being able to accommodate surface interactions between the wheels and the terrain. The foundation is actually fairly simple. We're using in each case a simple proportional integral derivative; the control algorithm is very linear. (Matijevic; 2005, side 50 i INTERVIEWS)

Mere avanceret kontrolteori kunne muligvis opdages i *Entry-Descent-Landing*-fasen (EDL). I *EDL*-fasen styrer fartøjet ind gennem atmosfæren og lander forhåbentligt på planeten. Fasen indeholder en masse hændelser, hvis formål er at bremse fartøjet; landeren bremses først af en faldskærm, som udfoldes, mens fartøjet er på vej ned gennem atmosfæren. Herefter affyres bremseraketter, og tilsidst pustes landerens airbags op. Når landeren er færdig med at hoppe henover planetens overflade lukkes luften ud af airbags og modulets tre kronblade åbnes. Når disse er åbne, folder roveren sine solpaneler ud. I hver af disse trin, som er illustreret på figur 2.4 , er involveret en vis mængde kontrolteori. Vi spurgte William Folkner om de nye missioner var mere avancerede i denne henseende end de tidligere:

2 Se eventuelt http://vesuvius.jsc.nasa.gov/er/seh/know2.html
3 Se eventuelt også (Kiely; 2005, side 36 i INTERVIEWS)



Figur 2.4 Ankomst, nedstigning og landing. (JPL; 2002, side 6.1)

> The Viking had powered descent. They had rockets coming down, a much more complex control than Pathfinder or MER. Much more complex. (Folkner; 2005b, side 76 i INTERVIEWS)

Kontrolteori virker altså heller ikke som det rette sted at lede efter matematik udviklet specielt til missionen.

2.3 Afrunding

Vi har i kapitlet forsøgt at beskrive nogen af de forhold, der gør sig gældende på JPL – både med hensyn til de ansatte og med hensyn til arbejdets form og indhold.

JPL er et sted næsten udelukkende med færdiguddannede forskere, hvis arbejdsopgaver i meget høj grad er fastlagte og skemalagte. Deadlines *skal* overholdes, og de fleste elementer i arbejdet, som kan bringe en sådan overholdelse i fare, fravælges. Dette betyder i stort omfang, at de problemstillinger, der arbejdes med, er velkendte problemstillinger med velkendte løsninger.

Indførelsen af de mindre 'low-cost' missioner har forstærket dette forhold yderligere, fordi man er nødt til at genbruge så meget som muligt fra mission til mission. Dette gælder både på det matematiske såvel som på det menneskelige plan.

Den beskrivelse, vi har givet af udvalgte matematiske problemstillinger, vidner om en matematik, som er forholdsvis gammel og velkendt og som holdes på et relativt simpelt niveau. Det sidste forhold skyldes højst sandsynligt, at der ikke er behov for mere avanceret matematik.

Man kunne indvende mod vores fremstilling, at vi bare har valgt eksempler, der illustrerer lige netop denne pointe. Rent faktisk er vi ikke stødt på eksempler, som vidnede om avanceret matematik udviklet specielt til missionen, og den her givne klare pointe skyldes således ikke manipulation med udtalelser.

I de følgende kapitler skal vi se nærmere på den del af matematikken i MER-missionen, som indgår i kommunikationen mellem roverne og Jorden.

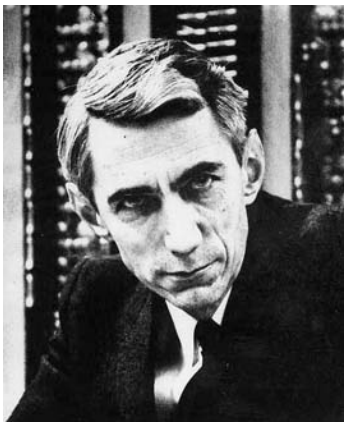
3 En matematisk teori for kommunikation

> Space missions, they are a great test case because the channel is very clean and because of the whole set up of the AWGN channel. So this is the only case where this really makes sense. (Kötter; 2005, side 26 i Interviews)

I dette kapitel vil vi overordnet præsentere en række begreber, som ligger til grund for vores behandling af kilde- og kanalkodning. Udgangspunktet for præsentationen er Shannons *A Mathematical Theory of Communication* fra 1948, hvorfra kodningsteori (kanalkodning) og *informationsteori*, som omfatter datakomprimering (kildekodning), udsprang. Med Shannons teori kvantiseres informationsbegrebet på en sådan måde, at man kan tale om, hvor meget information en besked indeholder. Informationsteori opfattes i dag som en del af matematisk sandsynlighedsteori og statistik.

Kapitlet indledes med opstillingen af en generel ramme for kommunikation i form af et såkaldt *kommunikationssystem*. Indsigt i kommunikationssystemets bestanddele danner grundlaget for en forståelse af, hvordan kodning og datakomprimering spiller sammen ved en transmission.

Efter opstillingen af kommunikationssystemet følger en kort præsentation af nogle centrale begreber i Shannons teori, samt en beskrivelse af, hvordan teorien gør sig gældende ved kommunikation i det ydre rum. Denne beskrivelse er centreret omkring den såkaldte *deep-space* kanal. Nærværende kapitel er fortrinsvist baseret på Shannon (1948).



Claude Elwood Shannon
(1916-2001)

Shannon modtog i 1940 sin ph.d.-grad i matematik ved MIT for et arbejde omhandlende populationsdynamik. Inden da havde han studeret såvel matematik som elektroteknik ved University of Michigan. Fra 1941-1972 var Shannon tilknyttet AT&T Bell Telephones i New Jersey som forskningsmatematiker, og det var herfra han udgav sin berømte 1948-artikel. I 1958 blev han udnævnt til Donner Professor of Science ved MIT. Shannon beskæftigede sig også med kunstig intelligens, herunder specielt skakprogrammer, og udgav i 1956 en artikel om den universelle Turing-maskine. Shannon var kendt for at holde sig for sig selv, men blev ofte set kørende rundt på sin ethjulede cykel, undertiden samtidigt jonglerende, til stor fare for sine kollegaer. Shannon skulle efter eget udsagn have arbejdet på en motoriseret kængurustylte, hvilket han hævdede skulle afløse den af kollegaerne frygtede ethjulede cykel. http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Shannon.html

Selv beskrev Shannon det grundlæggende problem inden for kommunikation således:

> The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is the one *selected from a set* of possible messages. (Shannon; 1948, side 5)

3.1 Et kommunikationssystem

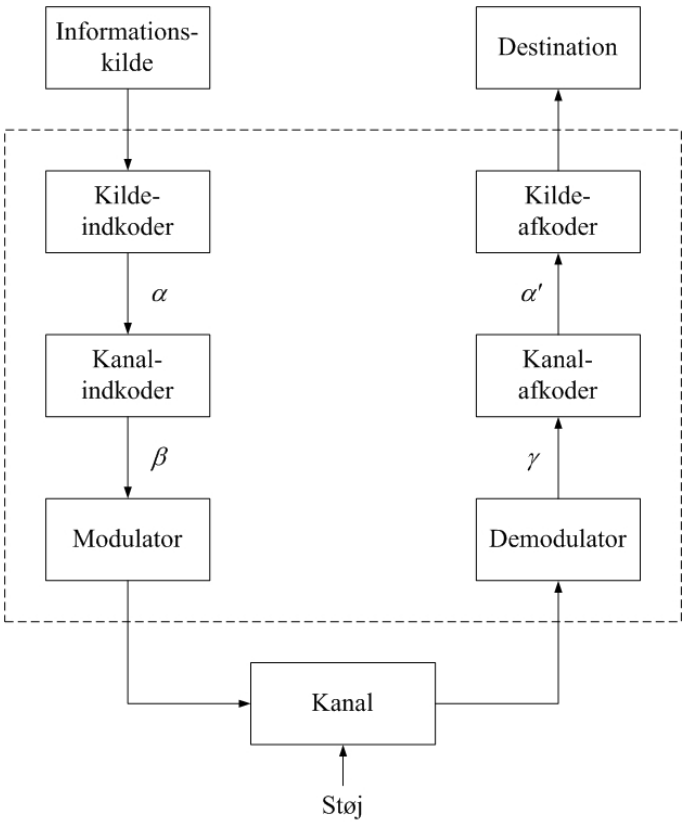
Kort fortalt forbinder et kommunikationssystem en informationskilde og en destination gennem en kanal. Kanalen er et medie, hvorigennem signalet, som indeholder en mængde information, udbredes og kan for eksempel være et kabel, en harddisk, en CD eller som i vores tilfælde æteren mellem Jorden og Mars. Nærværende afsnit er baseret på kilderne: (Blahut; 2003, side 2-4), (Lin & Costello, Jr.; 1983, side 1-3) og (Shannon; 1948, side 1-2, 5-6).

Informationskilden leverer en strøm af symboler kaldet en *datastrøm*, som kommunikationssystemet skal give mulighed for at transmittere til destinationen. For modtageren er kommunikationssystemets indre struktur skjult, men undervejs foregår en række behandlinger af datastrømmen, som beskrives i det følgende. Kommunikationssystemets bestanddele er illustreret på figur 3.1.

Datastrømmen fra informationskilden passerer først igennem en *kildeindkoder*, som omformer data, så disse opnår en mere kompakt form. Denne proces kaldes datakomprimering eller *kildekodning*. Uddata fra kildeindkoderen er en følge α af symboler, som kaldes en *kildekodestrøm*. Denne strøm sendes nu igennem en *kanalindkoder*, som tilføjer redundans til strømmen. Formålet med at tilføje denne redundans beskrives om lidt. Uddatafølgen β fra kanalindkoderen kaldes for en *kanalkodestrøm*. En *modulator* konverterer hvert symbol i kanalkodestrømmen til et analogt symbol, som kanalen er i stand til at transmittere. Symbolerne, som kan transmitteres gennem kanalen, tilhører en endelig mængde af symboler kaldet kanalens *alfabet*. Efter modulationen transmitteres symbolerne igennem kanalen. Der skal her gøres opmærksom på, at de forskellige strømmes symboler ikke nødvendigvis er tilhører samme alfabet.

Efter transmissionen sendes kanalkodestrømmen igennem en *demodulator*, som omformer strømmen til en følge γ af symboler fra kanalkodestrømmens alfabet. Denne strøm passerer herefter igennem en *kanalafkoder*, som omformer strømmen til en følge α' af symboler fra kildekodestrømmens alfabet. Under kanalafkodningen fjernes den tilføjede redundans. Tilsidst sendes denne strøm igennem en *kildeafkoder*, som dekomprimerer data. Uddata fra kildeafkoderen bliver sendt ud af kommunikationssystemet til modtageren.

Hvis alt går vel, er den besked, som modtages ved destinationen, identisk med den besked, der blev sendt ind i kommunikationssystemet fra



Figur 3.1 Indholdet af den stiplede kasse udgør et diagram for et typisk kommunikationssystem, hvor α er kildekodestrømmen, β er kanalkodestrømmen, γ er den modtagne kanalkodestrøm og α' kaldes et estimat af kildekodestrømmen. (Blahut; 2003, side 3), (Lin & Costello, Jr.; 1983, side 2)

informationskilden. Imidlertid vil der for de fleste kanalers vedkommende blive tilføjet en del støj under transmissionen. Denne støj kan resultere i at kanalkodestrømmen, som sendes til modulatoren og kanalkodestrømmen, som fremkommer efter demodulation, afviger fra hinanden, altså $\beta \neq \gamma$. Formålet med at tilføje den omtalte redundans er at gøre det muligt at gendanne den korrekte kildekodestrøm ved kanalafkodning, det vil sige $\alpha = \alpha'$. Hvordan dette foregår er beskrevet i del II i INDFØRELSEN.

For det konkrete tilfælde, som vi skal beskæftige os med, er informationskilden data fra MER-roveren. Disse data er digitale og udgøres derfor af en strøm af bit (eller bytes). Kilden siges af denne grund at være *binær*. Kildekodestrømmen og kanalkodestrømmen er ligeledes binære. For transmission mellem Mars og Jorden er kanalens alfabetet et alfabet bestående af forskellige bølgeformer, som sammensættes og transmitteres. Støjen på kanalen optræder i form af kosmisk stråling, der interfererer med den sendte radiobølge.

3.2 Kvantisering af information

Vi skal indledningsvis forklare et begreb, som er af afgørende betydning for, hvor meget information man kan sende gennem en kanal per tid. Vi vil begrænse vores fremstilling til *diskrete* kanaler, det vil sige kanaler med et endeligt kanalalfabet.

Et alfabet for en kanal består af en række symboler $s_1, s_2, \ldots, s_n$. Til hvert symbol er tilknyttet en varighed t_i, som er lig det tidsrum, det tager at transmittere symbolet over kanalen. Symbolernes varighed er ikke nødvendigvis ens[1]. En streng af symboler fra kanalens alfabet kaldes et *signal*. Hvis σ er et signal givet ved

$$\sigma = s_{i_1} s_{i_2} \ldots s_{i_n}$$

er signalets varighed lig $t_{i_1} + t_{i_2} + \ldots + t_{i_n}$, hvor t_{i_j} er varigheden af det j'te symbol i σ.

For en given kanal ønsker man ikke nødvendigvis at kunne transmittere samtlige mulige signaler, men kun at transmittere en særlig delmængde af signalerne. Denne delmængde kaldes kanalens *tilladte signaler*.

Definition 3.1 (Kapacitet af diskret støjfri kanal)
Lad for en given kanal $s_1, s_2, \ldots, s_n$ være kanalens alfabet, og lad de tilladte signaler konstrueret fra disse symboler være fastlagt. Kapaciteten C *af den diskrete støjfri kanal defineres da ved*

$$C = \lim_{T \to \infty} \frac{\log N(T)}{T},$$

hvor $N(T)$ angiver antallet af tilladte signaler af varighed T.

Logaritmen i ovenstående er totals-logaritmen og kanalkapaciteten har derfor enheden bit per sekund. I ovenstående er man nødt til kun at se på de T'er, som svarer til en lovlig varighed for et signal. Ellers konvergerer brøken på højresiden ikke. Det skal i øvrigt bemærkes, at begrebet en *bit* blev indført af Shannon i 1948-artiklen.

Eksempel 3.2
Lad der være givet en kanal, hvis alfabet består af $s_1, s_2, \ldots, s_{32}$, og lad for hvert $j = 1, \ldots, 32$ og hvert symbol s_j, varigheden af symbolet været givet ved $t_j = {}^1\!/_m$. Alle symboler har altså samme varighed, og der kan transmitteres m symboler per sekund, hvor m er en konstant. Lad samtlige kombinationer af symboler udgøre et tilladt signal. Antallet af signaler af varighed T er da lig antallet af signaler bestående af mT symboler, det vil sige 32^{mT}. Kanalens kapacitet er så givet ved

$$C = \lim_{T \to \infty} \frac{\log(32^{mT})}{T} = \lim_{T \to \infty} \frac{mT \log 32}{T} = 5m.$$

Kanalens kapacitet er altså $5m$ bit per sekund. (Shannon; 1948, side 6) $\diamond$

Kanalkapaciteten angiver en øvre grænse for hvor meget information, der kan transmitteres per tid gennem kanalen. For et konkret kommunikationssystem afhænger den faktiske transmissions effektivitet også af informationskilden.

1 Et eksempel på symboler med forskellig varighed kan findes i Morse-alfabetet.

3.2.1 Den diskrete informationskilde

Shannon beskrev den diskrete informationskilde med følgende ord:

> We may think of a discrete source as generating the message, symbol by symbol. It will choose successive symbols according to certain probabilities depending, in general, on preceding choices as well as the particular symbols in question. A physical system, or a mathematical model of a system which produces such a sequence of symbols governed by a set of probabilities is known as a stochastic process. We may consider a discrete source, therefore, to be represented by a stochastic process. (Shannon; 1948, side 6-7)

Målet for hvor meget information kilden 'producerer' ved en sådan stokastisk proces kaldes *entropien.*

Definition 3.3 (Entropi)
Antag at en enkelt tilfældig variabel x kan antage n værdier med sandsynligheder $p_1, \ldots, p_n$. Da defineres entropien *for x som*

$$H(x) = -K \sum_{i=1}^{n} p_i \log p_i,$$

hvor K er en positiv konstant, som fastlægger enheden for entropien.

Entropien giver et mål for, hvor usikker man er på værdien af x og dermed, hvor meget information x indeholder. Jo mere usikker man er, jo mere information indeholder x. Hvis logaritmen tages med basen 2, er enheden for entropien *bit*. At entropien defineres således begrundede Shannon selv med, at man til målet H må stille følgende tre krav:

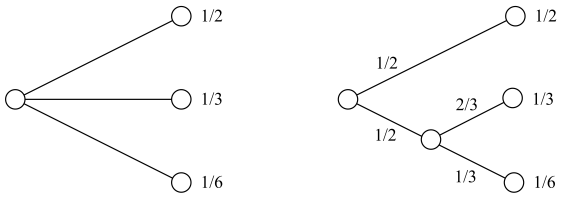
> 1. H should be continuous in the p_i.
> 2. If all the p_i are equal, $p_i = 1/n$, the H should be a monotonic increasing function of n. With equally likely events there is more choice, or uncertainty, when there are more possible events.
> 3. If a choice be broken down into two successive choices, the original H should be the weighted sum of the individual values of H. (Shannon; 1948, side 9)

Det kan vises (se Appendix II i (Shannon; 1948)), at det eneste H, der opfylder betingelserne er det givet i definition 3.3.

Betydningen af punkt 3 er illustreret på figur 3.2: Til venstre på figuren er vist tre udfald af den stokastiske proces med sandsynligheder $p_1 = 1/2$, $p_2 = 1/3$, $p_3 = 1/6$. Til højre vælges først mellem to udfald med sandsynlighed $1/2$, og såfremt anden mulighed indtræffer foretages endnu et valg, denne gang med sandsynligheder $2/3$, $1/3$. Det endelige resultat har samme entropi som før, det vil sige, at

$$H\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2} H\left(\frac{2}{3}, \frac{1}{3}\right),$$

hvor koefficienten $1/2$ skyldes, at dette andet valg kun forekommer halvdelen af gangene.



Figur 3.2 Shannons figur 6: »Decomposition of a choice from three possibilities«. (Shannon; 1948, side 9)

Entropibegrebet er ikke helt trivielt at forstå, fordi ens intuition omkring information ikke nødvendigvis er så veludviklet. Man kan tænke på entropien for en tilfældig variabel som et mål for, hvor svært det er at gætte den variables værdi. Hvis entropien er n bit, er det lige så svært at gætte værdien, som det er at gætte udfaldene af n på hinanden følgende møntkast.

Eksempel 3.4
Lad for et møntkast sandsynligheden for at få krone være $p_{\text{krone}} = 1/2$ og sandsynligheden for at få plat være $p_{\text{plat}} = 1/2$. For et sådant kast er entropien givet ved

$$H = -2 \cdot \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ (bit)},$$

hvilket stemmer overens med ovenstående forklaring. Hvis vi nu forskubber møntens balance sådan, at $p_{\text{krone}} = 1/4$ og $p_{\text{plat}} = 3/4$, må vi miste noget entropi, da det nu er lettere at gætte udfaldet. Entropien bliver da også

$$H = -\left(\frac{1}{4} \log_2 \frac{1}{4} + \frac{3}{4} \log_2 \frac{3}{4}\right) \sim 0,81 \text{ (bit)}.$$

Det er altså lettere at gætte udfaldet af vores særlige møntkast end udfaldet af et almindeligt møntkast. Hvis vi lader p_{krone} gå mod nul i ovenstående, går entropien mod nul i overensstemmelse med den intuitive forklaring om entropi som usikkerhed. ◇

En kilde kan opfattes som en generator af værdier for den tilfældige variabel x med en vis rate m målt i symboler per sekund. For en kilde giver det derfor mening at betragte entropien per sekund, H', som værende lig mH. Shannon (1948) benyttede selv såvel H som H' for *entropien*, den ene målt i bit per symbol og den anden målt i bit per sekund. Vi vil i det følgende gøre lige så.

Følgende vigtige resultat, som forbinder entropien af en kilde og kapaciteten af en kanal, kendes under navnet *The Noiseless Source Coding Theorem*. For et bevis henvises til (Shannon; 1948, side 11-12).

Sætning 3.5 (Hovedsætningen for en støjfri kanal)
Lad en kilde have entropi H målt i bit per symbol, og lad en kanal have kapacitet C målt i bit per sekund. Så er det muligt at indkode uddata fra kilden sådan, at der kan transmitteres med en gennemsnitlig rate givet ved

$$\frac{C}{H} - \epsilon$$

symboler per sekund, $\epsilon > 0$. Det er ikke muligt at transmittere med en rate større end C/H.

Sætningen fortæller, at man ved indkodning (kildekodning) af data kan øge den mængde information, som kan transmitteres over en kanal. Den øvre grænse afhænger af kildens entropi og af kanalens kapacitet.

Entropien kan på naturlig vis udvides til at omfatte et tilfælde med to begivenheder, x og y. Lad $p(i,j)$ være sandsynligheden for den fælles forekomst af i og j. Entropien for den fælles begivenhed defineres da som

$$H(x,y) = -\sum_{i,j} p(i,j) \log p(i,j).$$

Ved udregning ses, at der gælder

$$H(x,y) \leq H(x) + H(y).$$

Definition 3.6
Den betingede entropi, *$H_x(y)$, af y defineres ved*

$$H_x(y) = -\sum_{i,j} p(i,j) \log p_i(j). \tag{3.1}$$

I ovenstående er $p_i(j)$ den betingede sandsynlighed for, at y antager værdien j givet, at x antager værdien i. Den betingede sandsynlighed $p_i(j)$ er givet ved

$$p_i(j) = \frac{p(i,j)}{\sum_i p(i,j)}. \tag{3.2}$$

$H_x(y)$ giver et mål for, hvor usikker man i gennemsnit er på y, når man kender x. Indsættes ligning (3.2) i ligning (3.1) opnås

$$\begin{aligned} H_x(y) &= -\sum_{i,j} p(i,j) \log \frac{p(i,j)}{\sum_j p(i,j)} \\ &= -\sum_{i,j} p(i,j) \Big(\log p(i,j) - \log \sum_j p(i,j) \Big) \\ &= -\sum_{i,j} p(i,j) \log p(i,j) + \sum_{i,j} p(i,j) \log \sum_j p(i,j) \\ &= H(x,y) - H(x), \end{aligned}$$

hvorfor

$$H(x,y) = H(x) + H_x(y).$$

Ovenstående siger (intuitivt), at usikkerheden af den fælles begivenhed x, y er lig usikkerheden af begivenheden x plus den usikkerhed, der er forbundet med y, når x kendes. Resultatet anvendes i den følgende diskussion af støj.

3.2.2 Kapaciteten for en kanal med støj

For en kanal med støj kan et modtaget signal E opfattes som en funktion af det transmitterede signal S og den støj N, som S udsættes for under transmissionen, det vil sige

$$E = f(S, N). \tag{3.3}$$

Når en kanal med støj modtager inddata fra en kilde, er der to statistiske processer i spil, kilden og støjen. Lad i det følgende $H(x)$ betegne entropien af kilden og $H(y)$ entropien af det modtagne signal. I tilfældet uden støj gælder $H(x) = H(y)$. Den fælles entropi af kanalens ind- og uddata angives da ved $H(x, y)$. Den betingede entropi af uddata, når inddata kendes, betegnes $H_x(y)$, og den betingede entropi af inddata, når uddata kendes, betegnes $H_y(x)$. Mellem disse gælder, som tidligere beskrevet, følgende relationer:

$$H(x, y) = H(x) + H_x(y) = H(y) + H_y(x),$$

Relationerne danner grundlaget for endnu et vigtigt begreb.

Definition 3.7 (Rate)
Lad for en given kanal de ovenfor diskuterede entropier være kendt. Raten R *ved transmission af information er givet ved*

$$R = m(H(x) - H_y(x)),$$

hvor m *er antal symboler per sekund.*

Ovenstående forstås bedst ved at tænke på entropien som usikkerhed. Udtrykket $H_y(x)$ angiver, hvor usikre vi er på det transmitterede, når vi modtager signalet. Hvis kanalen er støjfri, er $H_y(x) = 0$, og kanalens rate er derfor lig kildens entropi. Jo større usikkerheden bliver, jo mindre bliver raten.

Med udgangspunkt i raten kan kanalkapaciteten defineres for en kanal med støj.

Definition 3.8 (Kapacitet af kanal med støj)
Kapaciteten *af en kanal med støj defineres som den maksimale rate, hvormed information kan transmitteres over kanalen, det vil sige*

$$C = \max(H(x) - H_y(x)),$$

hvor maksimum er med hensyn til alle mulige informationskilder brugt som inddata til kanalen.

Hvis $H_y(x) = 0$ er definitionen ækvivalent med den definition[2], der tidligere er givet for en støjfri kanal.

Vi er hermed nået til et vigtigt resultat, nemlig sætningen kendt under navnet *The Noisy Coding Theorem*. Sætningen er et eksistensudsagn, som inddrager kanalkodningen for kommunikationssystemet. For beviset henvises til (Shannon; 1948, side 14-15).

2 Definition 3.1

Sætning 3.9 (Hovedsætningen for en diskret kanal med støj)
Lad en diskret kanal have kapacitet C og en diskret kilde entropi per sekund H. Hvis $H \leq C$, eksisterer et kodningssystem, således at uddata fra kilden kan transmitteres over kanalen med en vilkårlig lille frekvens af fejl. Hvis $H > C$, er det muligt at indkode kilden, således at $H_y(x) < H - C + \epsilon$, hvor ϵ er vilkårligt lille.

Sætningen postulerer altså kanalkoder med en rate vilkårligt tæt på kanalkapaciteten og sandsynligheder for fejl vilkårligt tæt på nul.

3.2.3 Modulation

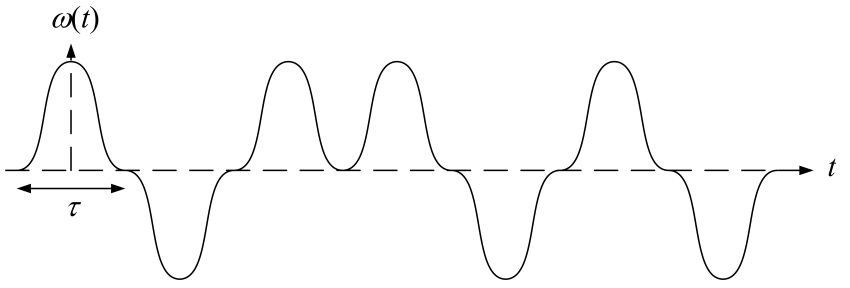
Måden hvorpå en følge af bit oversættes til for eksempel et elektrisk signal eller en radiobølge, der sendes gennem en kanal, afhænger af den anvendte *modulation*. Den foretrukne modulation for deep-space kommunikation er i følge Massey (1992) en modulation kaldet BPSK-modulation (*Binary Phase Shift Keying Modulation*). Modulation ligger uden for dette specialles fokus, og vi vil derfor nøjes med at antyde ideen bag.

Som et simpelt eksempel kan tænkes, at man anvender en enkelt bølgeform $\omega(t)$ til at transmittere to binære symboler. En bølgetop, der repræsenterer 1 og en bølgedal, der repræsenterer 0, sådan at

$$\omega_1(t) = \omega(t) \quad \text{og} \quad \omega_0(t) = -\omega(t).$$

En sådan modulation kaldes for *binær* modulation (se figur 3.3).

Hvis transmissionstiden for en signalbølge er τ, målt i sekunder, kan der ved binær modulation transmitteres data med en rate på $1/\tau$ bit per sekund. Jo smallere impulsen er, desto højere bliver bitraten altså. Vi vil ikke komme nærmere ind på modulation, men dog nævne to forskellige tilgange til den demodulation, der foregår i modtager-enden af kommunikationssystemet. Ved *hard-decision demodulation* sammenlignes den modtagne bølge ω med de to bølger ω_0, ω_1, og transformeres til en bit b, hvis værdi afhænger af hvilken af de to bølger, ω ligner mest. Ved *soft-decision demodulation* ledsages b af to[3] andre bit, som angiver, hvor meget ω er blevet deformeret under transmission, altså hvor godt ω ligner



Figur 3.3 Et eksempel på binær modulation af strengen 1011010. En bølgetop repræsenterer 1, mens en bølgedal repræsenterer 0. Bredden af signalbølgen er τ.

3 Demodulationen kaldes så 3-bit soft-decision demodulation.

ω_0 eller ω_1. Hard-decision demodulation sletter ved sin 1-bit kvantisering af signalet en del information om transmissionens kvalitet og forringer derfor kanalens kapacitet væsentligt. Man er imidlertid nødt til at foretage en kvantisering, og tre bit har vist sig at være nok til, at den information, der smides bort ved kvantiseringen, kan negligeres i deep-space sammenhænge. Fordelen ved soft-decision demodulation har været kendt siden starten af 1960'erne og stort set alle kodningssystemer, der har været i brug i deep-space anvendelser har derfor gjort brug af soft-decision demodulation.

Vores beskrivelse af binær modulation er meget forsimplet. I praksis foregår modulationen i flere trin. Jørn Justesen fortalte:

> Du tager dit digitale signal, som det kommer ud af kredsløbene – det vil sige nuller og et-taler, altså to spændingsniveauer – og det skal igennem et modulationssystem og ende med at være noget i gigahertz-området, som kan sendes ud. Som regel er der imellem de trin en hel række pakketrin, hvor man gradvist modulerer signalerne op og pakker signalerne sammen i en bunke. Tilsidst ender det med at være én stor samlet sag, som bliver sendt af sted. (Justesen & Høholdt; 2004b, side 6 i Interviews)

Vi vil ikke behandle de nævnte pakketrin, da vi kun er interesserede i de ovenfor givne principper for modulation.

3.3 AWGN-kanalen

Der findes en særlig kanal, som er af stor relevans for kommunikationen i rummet. Kanalen kendes i dag under betegnelsen AWGN-kanalen. AWGN står for *Additive White Gaussian Noise*. Betydningen af dette navn forklares i det følgende.

En *additiv kanal* er en kanal, hvor støjens påvirkning af signalet er uafhængig af det transmitterede signal. For en sådan kanal, kan det modtagne signal, E, beskrives ved

$$E = S + N,$$

hvor S og N er henholdsvis det transmitterede signal og støjen (sammenlign eventuelt med ligning (3.3)).

Begrebet *White Noise* dækker over støj, som indeholder alle frekvenser, eller i hvert fald dækker et meget bredt bånd af frekvenser. Adjektivet *Gaussian* angiver, at amplituden af støjen over frekvensspektret er Gauss-fordelt. I rummet kan hvid Gaussisk støj blandt andet stamme fra udstråling fra diverse himmellegemer.

AWGN-kanalen er altså en kanal med et helt særligt støjbillede. Det skal nævnes, at i forhold til de tidligere diskuterede kanaler, er AWGN-kanalen en kontinuert kanal, det vil sige en kanal, hvis tilladte symboler er kontinuerte tidsafhængige funktioner. Når vi anvender binær modulation har kanalen i princippet kun to tilladte symboler, og vi vil derfor ikke forfølge de kontinuerte aspekter yderligere.

3.3.1 Deep-space kanalen

Kommunikation i rummet foregår ved udbredelse af radiobølger gennem æteren. Kanalen, der transmitteres over, kaldes for *deep-space kanalen*. Deep-space kanalen beskrives yderst præcist ved Shannons matematiske model, AWGN-kanalen. Ifølge Massey (1992) er overensstemmelsen imellem de to så god, at der ikke er observeret afvigelser mellem deep-space kanalen og AWGN-kanalen.

Karakteristisk for deep-space kanalen er, at der i praksis er nærmest ubegrænset *båndbredde* til rådighed. Båndbredden angiver hvilket frekvensområde, der kan benyttes til transmission, og kan opfattes som et mål for antallet af kanalsymboler, der hvert sekund kan transmitteres.

For transmission over deep-space kanalen gælder, at det modtagne signal er summen af det transmitterede signal og en hvid Gaussisk støjproces, N_0, målt i watt/Hz (joule). Hvis det transmitterede signal begrænses til at have en båndbredde på højst W Hz samt at have en gennemsnitlig effekt på P watt eller mindre, er kapaciteten C for kanalen målt i bit per sekund givet ved

$$C_W = W \log_2 \left(1 + \frac{P}{N_0 W}\right).$$

Ved at lade båndbredden gå mod uendelig (hvilket giver mening for deep-space kanalen) fås maksimumsværdien af kapaciteten

$$C_\infty = \frac{1}{\ln 2}\frac{P}{N_0} \approx 1,44\frac{P}{N_0}.$$

Antag, at der transmitteres informationsbit med en rate R meget tæt på denne kapacitet. Energien per informationsbit, E_b, er da givet ved:

$$E_b = \frac{P}{R} \approx \frac{P}{C_\infty} = \ln 2 N_0 \approx 0,69 N_0 \text{ (joule)}.$$

Signal-til-støj forholdet er da givet ved

$$\frac{E_b}{N_0} \approx 0,69 \quad \text{(eller -1,6 decibel)}, \tag{3.4}$$

hvilket er det mindste signal-til-støj forhold krævet for vilkårlig pålidelig kommunikation. Dette forhold kendes som *Shannon-grænsen* for AWGN-kanalen. (Shannon; 1949, side 39), (Massey; 1992, side 3-4)

For en given transmission kaldes forholdet mellem antallet af fejlbit (bit, der har ændret sig under transmission) og det samlede antal transmitterede bit for transmissionens *fejlrate* eller BER (*Bit Error Ratio*). Vilkårlig pålidelig kommunikation betyder, at man kan reducere fejlraten så meget, man vil. Generelt er man dog ikke interesseret i vilkårlig pålidelig transmission. For transmission af bestemte typer af data vil man typisk have en øvre grænse for, hvor stor en BER man kan tolerere. Jo mere følsomme data er, jo mindre en BER kan tolereres.

Fejlraten for en transmission afhænger generelt af forholdet mellem signal og støj, E_b/N_0. Kanalkodning giver mulighed for at rette en del af de fejl, som kan opstå i transmissionen, og reducerer derfor transmissionens fejlrate. Når man bedømmer effektiviteten af en kode, ser man dog typisk ikke på kodens reduktion af fejlraten. I stedet specificerer man en

grænse for den tolererede fejlrate og bedømmer koden på, hvor meget den reducerer det forhold E_b/N_0, som sikrer, at transmissionens fejlrate ligger under grænseværdien. Reduktionen af E_b/N_0 er typisk beregnet på baggrund af værdien af E_b/N_0 for ukodet transmission, men kan også være i forhold til en anden kode. Reduktionen kaldes *coding gain* og måles typisk i decibel (dB).

For en given kode kan man ikke umiddelbart sige noget om, hvor mange dB, man vinder ved at anvende koden. Vindingen, som er stærkt afhængig af hvilken BER-grænse, der er fastsat, er man nødt til at evaluere empirisk ved afprøvning.

Enhver kanalkode er blandt andet karakteriseret ved kodens rate. Jo lavere raten er, jo flere bit skal transmitteres per informationsbit, og jo langsommere skulle man tro, at transmissionen ville foregå, men på grund af deep-space kanalens ubegrænsede båndbredde, kan man kompensere for dette forhold sådan, at transmissionen kan foregå lige så hurtigt, som hvis raten var 1. Umiddelbart skulle man tro, at man så bare kan sætte raten tilstrækkeligt langt ned[4] og udnytte den ubegrænsede båndbredde til at få transmissionen igennem med samme bitrate. Imidlertid gælder, at demodulation bliver sværere, når den båndbredde, der benyttes, øges. Demodulationen bliver hurtigt næsten umulig at foretage. Selvom båndbredden er ubegrænset for deep-space kanalen gælder det altså om at finde gode koder, som samtidig har en stor rate. (Massey; 1992, side 8)

I stedet for at anvende kanalkodning kan man øge energien per informationsbit E_b ved simpelthen at øge effekten P for signalet. For en BER på 10^{-5}, som ifølge Costello, Jr. et al. (1998) er en typisk værdi for komprimerede billeddata, kræves ved ukodet transmission en signalstyrke så $E_b/N_0 = 9,6$ dB, altså at signalstyrken er mellem 10^9 og 10^{10} gange så kraftig som støjen. For deep-space anvendelser kan sådanne forhold ikke komme på tale, faktisk kan BER ikke for ukodet transmission bringes i nærheden af de størrelsesordner, der er behov for. Dette kommenteres således i Massey (1992):

> The inescapable conclusion is that *if one wants to signal both energy-efficiently and reliably with binary modulation on the deep-space channel, then one must use channel coding.* (Massey; 1992, side 6)

Kodningens betydning skal ses i lyset af, at 3 dB vundet ved kodning har lige så stor effekt som en fordobling af antennearealet eller effekten, der transmitteres med. Shannon-grænsen dikterer, at kodning ikke kan reducere det krævede E_b/N_0 til mindre end $-1,6$ dB, men for en BER på 10^{-5} kan man se fra udregningen for ukodet transmission, at der i princippet er omtrent $11,2$ dB at vinde ved kodning. (Blahut; 2003, side 406-414)

3.3.2 CCSDS

CCSDS (*Consultative Committee for Space Data Systems*) blev etableret i 1982 af de største rumagenturer som et forum, i hvilket typiske problemstillinger inden for data- og informationssystemer til rummet kunne

4 For eksempel med en repetitionskode. Se kapitel 6 i Indførelsen.

diskuteres. CCSDS har siden etableringen leveret løsninger på standardproblemer inden for udvikling af informationssystemer til rummet i form af anbefalinger til eksempelvis kanal- og kildekodning. Hensigten med at udvikle disse anbefalinger har været at reducere udviklingsomkostninger såvel som at fremme samarbejde rumagenturerne imellem. Over 300 rummissioner har benyttet sig af CCSDS-standarderne[5]. CCSDS' betydning for MER vil blive diskuteret i senere kapitler.

3.4 Afrunding

Vi har i dette kapitel præsenteret udvalgte dele af Shannons teori, herunder hans definition af det generelle begreb *et kommunikationssystem*. Vi har set på deep-space kanalen, som er særligt vigtig for vores fremstilling, og vi har beskrevet andre dele af kommunikationssystemet, hvis roller bliver uddybet i de følgende kapitler.

Vi har diskuteret et par af Shannons resultater. For kildekodningens vedkommende har vi præsenteret sætning 3.5, som grundlæggende set fortæller, at komprimering af data er mulig, og for kanalkodningens vedkommende sætning 3.9, der fastslår, at man gennem kanalkodning kan sikre kommunikation med vilkårlig lav sandsynlighed for transmissionsfejl. Vi har desuden for deep-space kanalen beskrevet de konsekvenser, som resultatet har i praksis.

Det bør bemærkes, at Shannons bevis for sætning 3.9 ikke er et konstruktivt bevis, men derimod et rent eksistensudsagn. Dette udsagn udgjorde i de følgende årtier grundlaget for 'jagten på de lovede koder'. I Calderbank (1998) kommenteres dette således:

> ... it is fair to credit Shannon with creating coding theory in that he established fundamental limits on what was possible, and presented the challenge of finding specific families of codes that achieve capacity. (Calderbank; 1998, side 2561)

5 Oplysningerne stammer fra `http://public.ccsds.org/default.aspx`.

4 Kildekodning og kanalkodning

> Når man laver billedkodning, forsøger man at arbejde med de billeder, man har, mens man med fejlkorrigerende koder anvender en mere ren matematisk opsætning. (Forchhammer; 2004, side 10 i INTERVIEWS)

Dette kapitel indeholder et resume af teorien bag den kilde- og kanalkodning, der anvendes i MER. Kapitlet tjener som en opsummering af INDFØRELSEN for de læsere, der allerede kender detaljerne. For en mere uddybende gennemgang henvises til INDFØRELSEN, som forklarer og eksemplificerer de begreber og teknikker, der opsummeres i dette kapitel.

4.1 Kildekodning

Den del af kildekodningen i MER, som vi er interesserede i, er billedkomprimering. MERs komprimeringsalgoritme ICER følger et generelt paradigme inden for billedkomprimering, nemlig *forbehandling* efterfulgt af en *modellering* og tilsidst *entropikodning*. En kort opsummering af det teoretiske grundlag for disse begreber gives i det følgende.

4.1.1 Forbehandling med wavelets

Formålet med at foretage en forbehandling af et billede, er at gøre billedet mere modtageligt overfor modellering og senere entropikodning. Et billede med $n \times n$ pixels repræsenteres naturligt ved en $n \times n$-matrix B med heltallige værdier.

Den forbehandling, som vi er interesserede i for MERs vedkommende, udgøres af en invertibel transformation

$$T : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

som skal påføres rækker og derefter søjler i B. Ved transformationen foretages et basisskift fra den euklidiske basis ind i en ortonormalbasis, hvis elementer kaldes *wavelets*.

Vektorerne i basen er bølgeformer med varierende frekvenser, der er skaleret, så de muliggør beskrivelser på forskellige opløsningsniveauer. Basisvektorerne er i et vist omfang både lokaliseret i frekvens og i rum. Rumlokaliseringen giver mulighed for kompakte lokale beskrivelser af et billede, og i forhold til harmoniske bølgeformer som for eksempel cosinusfunktioner giver dette en fordel ved komprimering af naturlige billeder.

Den specifikke transformation, som vi er interesserede i, kaldes *Haar*-transformationen. Haar-transformationen opdeler et billede i to delbilleder. Det ene delbillede indeholder alle det oprindelige billedes lave frekvenser, og det andet indeholder alle de høje. Transformationen koncentrerer

en meget stor del af energien i det lavfrekvente område, hvilket er vigtigt, fordi det muliggør effektiv komprimering med tab.

I praksis kan Haar-transformationen foretages ved en såkaldt liftingprocedure. Transformationen foretages så igennem en række simple beregninger, der følger på hinanden. Lifting-proceduren er defineret ved

$$L(z) = \left(\sqrt{2}(z_e + \frac{1}{2}(z_o - z_e)), \frac{1}{\sqrt{2}}(z_o - z_e) \right),$$

hvor z_e angiver de lige indicer i z og z_o de ulige. Lifting-procedurens forudsigelse P er fastlagt på baggrund af antagelsen om, at billedets pixelværdier er mere eller mindre konstante, det vil sige, at en pixel med ulige indeks n forudsiges at være lig pixelen med indeks $n-1$. For hvert trin i liftingen foretages en opdatering U, som udjævner signalet og dermed gør det mere modtageligt overfor forudsigelser på næste lifting-trin.

At Haar-transformationen er invertibel betyder, at den kan benyttes som forbehandling i algoritmer både med og uden tab.

4.1.2 Modellering

Modellering består i at opstille en model – kaldet en *kilde* – der beskriver genereringen af de data, der skal komprimeres (entropikodes). Modellen anvendes til estimering af entropien af en fysisk datagenererende proces. En god model fører til en mere effektiv entropikodning. Kilder kan ligesom kanaler være såvel diskrete som kontinuerte, være hukommelsesløse eller have en hukommelse. Kilden opfattes som værende en stokastisk proces.

En *diskret* kilde beskriver en fysisk proces, der genererer en besked symbol for symbol. For en diskret *hukommelsesløs* kilde gælder, at kildesymbolerne er statistisk uafhængige. Kilden er således fuldstændigt beskrevet ved sit symbolalfabet $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ og den tilhørende sandsynlighedsfordeling $\mathcal{P} = \{p(s_1), p(s_2), \ldots, p(s_n)\}$. For en diskret kilde med *hukommelse* forstås derimod, at sandsynligheden for forekomsten af et bestemt symbol afhænger af et fastlagt antal af de symboler, som tidligere er forekommet. Digitale billeder, hvor sandsynligheden for at en given pixel antager en bestemt værdi afhænger af de omkringliggende nabopixels værdier, kan modelleres ved sådanne kilder.

For kildekodning af billeder er det altså de diskrete kilder med hukommelse, der er de interessante. Sådanne kendes i litteraturen under navnet *Markov-kilder*, som er en generalisering af de diskrete hukommelsesløse kilder.

Definition 4.1 (Markov-kilde)
For en m'te-ordens Markov-kilde *afhænger sandsynligheden for at et kildesymbol* s_i *optræder af de foregående* m *symboler. Markov-kilden er beskrevet ved en mængde af betingede sandsynligheder:*

$$p(s_i | s_{j_1}, s_{j_2}, \ldots, s_{j_m}),$$

for $i = 1, 2, \ldots, n$ *og* $1 \leq j_p \leq n$ *med* $p = 1, 2, \ldots, m$.

En Markov-kilde vil befinde sig i en eller anden tilstand, som afhænger af de foregående m symboler, hvorfor den også gerne beskrives ved hjælp af et tilstandsdiagram. Markov-kilder kendes i litteraturen, og specielt i billedkomprimeringssammenhænge, også under navnet *endelige kontekstmodeller*.

4.1.3 Entropikodning

En entropikoders job består i at komprimere data ved at udskifte kildens symboler med kodeord af varierende længde. Typisk tildeles et symbol en kodeordslængde proportional med $-\log p(s_i)$, hvor $p(s_i)$ er sandsynligheden for at symbolet s_i optræder. De mest hyppige symboler tildeles på den måde de korteste kodeord.

Hvis en entropikode kan afkodes på præcist én måde, siges den at være *entydigt dechifrerbar*. En klasse af koder, som altid besidder denne attraktive egenskab, er prefix-koderne.

Definition 4.2 (Prefix-kode)
En kode $\mathcal{C}$ siges at være en prefix-kode*, eller at have* prefix-egenskaben*, hvis intet kodeord er et prefix af et andet kodeord, det vil sige, at når $c = c_1c_2 \ldots c_n$, så er $c_1c_2 \ldots c_k$ for $1 \leq k \leq n$ ikke et kodeord.*

Der gælder ifølge en sætning af Kraft, at man for en hvilken som helst entydigt dechifrerbar ikke-prefix-kode altid kan finde en prefix-kode med samme kodeordslængder, hvilket betyder, at man ikke mister noget ved at begrænse opmærksomheden til prefix-koder. En prefix-kode siges at være *udtømmende*, hvis der ikke kan tilføjes kodeord til koden uden af krænke dens prefix-egenskab.

Redundansen af en kode er givet ved forskellen mellem den gennemsnitlige længde af kodens ord og entropien H af den kilde, der indkodes. Jo mindre redundansen er, jo mere optimal er den pågældende kode.

Definition 4.3 (Optimum-kode)
En optimum-kode *er en kode med minimum redundans.*

En ofte benyttet teknik til kildeindkodning er *Huffman-kodning*. Huffman-koder er variabel-længde prefix-koder, som i ovenstående forstand er optimale for en given kilde.

Golomb-koder udgør en anden optimal familie af koder parametriseret ved et heltal $m > 0$. En Golomb-kode indkoder heltal under antagelse af, at jo større heltallet er, desto mindre er sandsynligheden for, at det forekommer. En Golomb-kode $\mathcal{G}_m$ repræsenterer et heltal $n > 0$ ved

$$n = qm + r \quad (0 \leq r < m),$$

hvor q er kvotienten og r resten, når n divideres med m. Kvotienten q kan antage værdierne $0, 1, 2, \ldots$ og repræsenteres ved den unære kode for q (q 1-taller efterfulgt af et 0, eller omvendt). Resten r repræsenteres ved at anvende en $\lfloor \log_2 m \rfloor$-bit binær repræsentation for de første $2^{\lceil \log_2 m \rceil} - m$ værdier og en $\lceil \log_2 m \rceil$-bit binær repræsentation af $r + 2^{\lceil \log_2 m \rceil} - m$ for de resterende værdier. Det kan vises, at Golomb-koder er optimale for fordelinger af formen

$$Q(n) = (1 - \rho)\rho^n \text{ for } 0 < \rho < 1.$$

For hver fordeling af denne form eksisterer en værdi af m således, at $\mathcal{G}_m$ giver den kortest mulige gennemsnitlige kodelængde over alle entydigt dechifrerbare koder for ikke-negative heltal. Golomb-kodning tilbyder sådanne fordelinger en simplere metode til generering af variabel-længde optimum-koder end Huffmans algoritme.

Forskellige indkodningsteknikker har forskellige styrker og svagheder. *Sammenflettet entropikodning* er en metode, der forsøger at samle 'det bedste' fra et udvalg af indkodningsteknikker ved i en og samme algoritme at 'flette' uddata fra forskellige koder.

Komprimeringsteknikker kan deles i *adaptive* og *ikke-adaptive* teknikker. De ikke-adaptive er karakteriseret ved, at de ikke løbende modificerer deres operationer og parametre i forhold til de data, der komprimeres. Det gør de adaptive derimod. Huffman- og Golomb-kodning er som udgangspunkt ikke-adaptive teknikker, men findes også i adaptive udgaver. Både adaptive og ikke-adaptive teknikker til entropikodning findes i MERs algoritmer, ICER og LOCO-I.

4.2 Kanalkodning

For MER-missionen er der to koder, der er interessante. Den ene kode er en *blokkode*, nærmere bestemt en *Reed-Solomon* kode. Den anden er en *foldningskode*. Begge koder er lineære.

4.2.1 Reed-Solomon koder

Reed-Solomon koderne tilhører en generel klasse af koder, som kaldes cykliske koder. En cyklisk (n, k)-kode $\mathcal{C}$ over $\mathbb{F}_q$ udgør et k-dimensionalt *cyklisk* underrum af vektorrummet $\mathbb{F}_q^n$. Cykliske koders struktur betyder, at koderne kan defineres med et algebraisk udgangspunkt.

Definition 4.4 (Cyklisk kode)
En cyklisk kode $\mathcal{C}$ med bloklængde n er et ideal i ringen $\mathbb{F}_q[x]/\langle x^n - 1\rangle$.

At en cyklisk kode $\mathcal{C}$ er et ideal i en polynomiumsring betyder, at der findes et entydigt monisk polynomium $g(x)$, der genererer idealet og dermed koden. Dette polynomium kaldes kodens *generatorpolynomium*. Et polynomium g af grad r genererer en $(n, n-r)$-kode. Den algebraiske tilgang tillader, at indkodning af et datapolynomium $a(x)$ kan foretages ved multiplikation med $g(x)$, sådan at

$$c(x) = a(x)g(x).$$

$c(x)$ er kodepolynomiet for $a(x)$.

Reed-Solomon koderne kan indføres som et specialtilfælde af de cykliske koder.

Definition 4.5 (Reed-Solomon kode)
Lad α være et primitivt element i legemet $\mathbb{F}_q$. En Reed-Solomon kode*, $\mathcal{C}$ er en cyklisk kode med bloklængde $q-1$ og generatorpolynomium*

$$g(x) = \prod_{i=b}^{r-1+b} (x - \alpha^i),$$

hvor $b \geq 0$ og $r < n$. Parameteren r angiver antal rødder og dermed graden af generatorpolynomiet.

For at konstruere en Reed-Solomon kode med bloklængde $q-1$ over et legeme $\mathbb{F}_q$ behøver vi altså blot at vælge et α og at specificere r, som

bestemmer kodens dimension, samt b, som er en parameter, der er mest interessant for hardware-implementering af koden.

Det særlige valg af bloklængden samt generatorpolynomiets rødder betyder, at en Reed-Solomon kode udgør et cyklisk underrum $\mathcal{C}$, hvis elementer c opfylder, at de Fouriertransformerede $\hat{c}$ har r på hinanden følgende komponenter lig 0. Fouriertilgangen er interessant, fordi den udgør et stærkt analyseapparat, som åbner op for en række særligt effektive afkodningsprocedurer. En af dem skal vi se i kapitel 6. Desuden er det let at skifte mellem de to tilgange.

Reed-Solomon koderne er karakteriseret ved at opfylde Singleton-grænsen med lighed, det vil sige

$$d_{\min} = n - k + 1.$$

Når blokkoder evalueres er den dominerende parameter typisk kodernes minimumsafstand $d_{\min}$. Reed-Solomon koderne er optimale i denne forstand.

4.2.2 Foldningskoder

En beskrivelse af foldningskodning kan opnås ved at lade en følge x af objekter (rammer) fra en ring $\mathbb{O}$ være repræsenteret ved uendelige formelle *Laurent-rækker*

$$x(D) = \sum_{i \in \mathbb{Z}} x_i D^i,$$

i det ubestemte symbol D, kaldet *forsinkelsesoperatoren*. Nøjes vi med at se på *en-sidede Laurent-rækker*, som har $x_i = 0$ for $i < N \in \mathbb{Z}$ fås en mængde, som skrives $\mathbb{O}((D))$. Mængden af polynomier med koefficienter fra $\mathbb{O}$, kaldet $\mathbb{O}[D]$, er på oplagt vis indlejret i $\mathbb{O}((D))$. Vi kan udstyre $\mathbb{O}((D))$ som en ring således, at multiplikationen er en udvidelse af multiplikation af polynomier. Hvis $\mathbb{O}$ er lig $\mathbb{F}^\ell$, hvor $\mathbb{F}$ er et legeme, så har vi identifikationen

$$\mathbb{F}^\ell((D)) = \mathbb{F}((D))^\ell.$$

I denne specielle situation er $\mathbb{F}((D))$ et legeme, og mængden af rationale udtryk af formen $p(D)/q(D)$, hvor $p(D)$ og $q(D)$ er polynomier med koefficienter fra $\mathbb{F}$ (og $q(D) \neq 0$), udgør da selv et legeme, kaldet $\mathbb{F}(D)$ (brøklegemet over integritetsområdet $\mathbb{F}[D]$).

En *generatormatrix* for en foldningskode er en matrix over legemet $\mathbb{F}(D)$ og foldningskoderne kan defineres ved:

Definition 4.6 (Foldningskode)
En (n, k)-foldningskode $\mathcal{C}$ *er et* k*-dimensionalt underrum af* $\mathbb{F}_q(D)^n$.

En (n, k, m)-*foldningsindkoder* kan beskrives ved en $k \times n$ generatormatrix, $G(D)$, bestående af generatorpolynomier $g_{ij}(D)$ af grad (højst) m med koefficienter fra det endelige legeme $\mathbb{F}_q$. Forholdet mellem indkoderens ind- og uddata er givet ved

$$y_j(D) = \sum_{i=0}^{k-1} x_i(D) g_{ij}(D),$$

hvor $x_i(D)$ udgør de k-dimensionale datarammer og $y_i(D)$ de n-dimensionale koderammer. En kode genereret med en (n, k, m)-foldningsindkoder kaldes en (n, k, m)-*foldningskode*.

Til en foldningsindkoder stilles krav om, at den skal være *lineær* såvel som *tidsinvariant*, hvilket vil sige

$$G(D^n x) = D^n G(x).$$

Ydermere stilles det krav til generatormatricen, at den skal være *forsinkelsesfri* og *realiserbar*. Et generatorpolynomium hvorom gælder, at det ikke har D som faktor, kaldes et forsinkelsesfrit polynomium. En generatormatrix, der indeholder et sådant forsinkelsesfrit polynomium, kaldes en forsinkelsesfri generatormatrix. Et element $a(D)/b(D)$ kaldes *realiserbart*, hvis D ikke er en faktor i $b(D)$. En generatormatrix hvis elementer alle er realiserbare kaldes en *realiserbar* generatormatrix. Kravet om realiserbarhed er automatisk opfyldt i kraft af definitionen af $\mathbb{F}(D)$. Der gælder, at enhver foldningskode har en forsinkelsesfri, (realiserbar,) polynomial generatormatrix.

Foldningsindkodningen kan beskrives ved en semi-uendelig generatormatrix G af undermatricer, som

$$y = xG, \tag{4.1}$$

hvor x er følgen af k-dimensionale datarammer, og y er følgen af n-dimensionale koderammer. Højre-siden i ligning (4.1) definerer således en *foldning* mellem x og en følge af undermatricer $G_0\, G_1\, \ldots\, G_m$.

Foldningskodernes sidestykke til blokkodernes minimumsafstand er den såkaldte *frie* Hammingafstand, d_{fri}. Der gælder, at en foldningskode ved 'nærmeste-nabo afkodning' kan korrigere et vilkårligt mønster af t fejl, så længe $t < d_{\text{fri}}/2$.

4.3 Afrunding

De begreber og teknikker, der er præsenteret overordnet i dette kapitel, vil blive konkretiseret i de følgende to kapitler, som handler om de konkrete kilde- og kanalkoder, der anvendes i MER.

Den beskrivelse, som forefindes i det følgende, afspejler i høj grad det underliggende i dette kapitels indledende citat, nemlig at det er nemmere at give en matematisk beskrivelse af kanalkodningen end af kildekodningen, fordi sidstnævnte i høj grad drejer sig om 'tricks' frem for matematik. Dette forhold fremgår også af de to følgende kapitler, der handler om henholdsvis kilde- og kanalkodningen i MER.

5 Kildekodning i MER

> It doesn't sound like a big deal but bits from deep space are pretty expensive so well, if it is ten percent more images you can get, you know, we spend a lot of millions to put the rovers up there in the first place. (Kiely; 2005, side 41 i INTERVIEWS)

I del I af INDFØRELSEN findes en beskrivelse og forklaring af mange af de begreber, som diskuteres i dette kapitel.

Vi vil i kapitlet begynde med at nævne de særlige omstændigheder, der gør sig gældende for en komprimeringsalgoritme om bord på en MER-rover. Derefter vil vi give en overordnet beskrivelse af MER-missionens komprimeringsalgoritme ICER[1], herunder en forklaring af de konkrete valg, der er foretaget med hensyn til kildekodningen. Efterfølgende vil vi i større detalje beskrive ICERs forskellige bestanddele.

Der gives desuden en kortfattet beskrivelse af en teknik, der anvendes i MER til tabsfri komprimering, LOCO-I. Afslutningsvis diskuteres valget af komprimeringsteknikker i MER samt udviklingen af ICER.

5.1 Særlige omstændigheder for roverne

Komprimering af billeder kræver en stor mængde regnekraft, som grundet udviklingen af hurtige processorer ofte tages for givet i dagligdags anvendelser på Jorden. Virkeligheden på Mars er imidlertid en anden.

Under rejsen fra Jorden til Mars udsættes roveren for en betragtelig mængde radioaktiv stråling, især ved passagen gennem van Allen-bæltet. En sådan stråling vil være katastrofal for processoren i en moderne hjemmecomputer, fordi computerens transistorer er så små, at strålingen ødelægger dem. For at undgå denne ødelæggelse er Mars-rovernes processorer designet, så de kan modstå strålingen, og det koster regnekraft. Desuden skal alt udstyr, som sendes afsted på en rummission, være afprøvet ned til mindste detalje, og udviklingen af nye processorer til rummet går derfor langsomt.

De primære processorer, der er om bord på de to Mars-rovere, er RAD6000-processorer med clockfrekvenser på 20 MHZ. RAD6000 er en 'radiation-hardened single board computer', som minder om tidlige modeller af PowerPC processor-familien. Groft sagt er processoren en hærdet udgave af Intels 8086'er fra 1978. (Justesen & Høholdt; 2004b, side 5 i INTERVIEWS), (Kiely; 2005, side 39 i INTERVIEWS)

Den ringe processorkraft betyder, at der må stilles særligt høje krav med hensyn til hastigheds- og pladsoptimering for de programmer, der skal afvikles på en rovers primære computer. Roverens processor skal håndtere

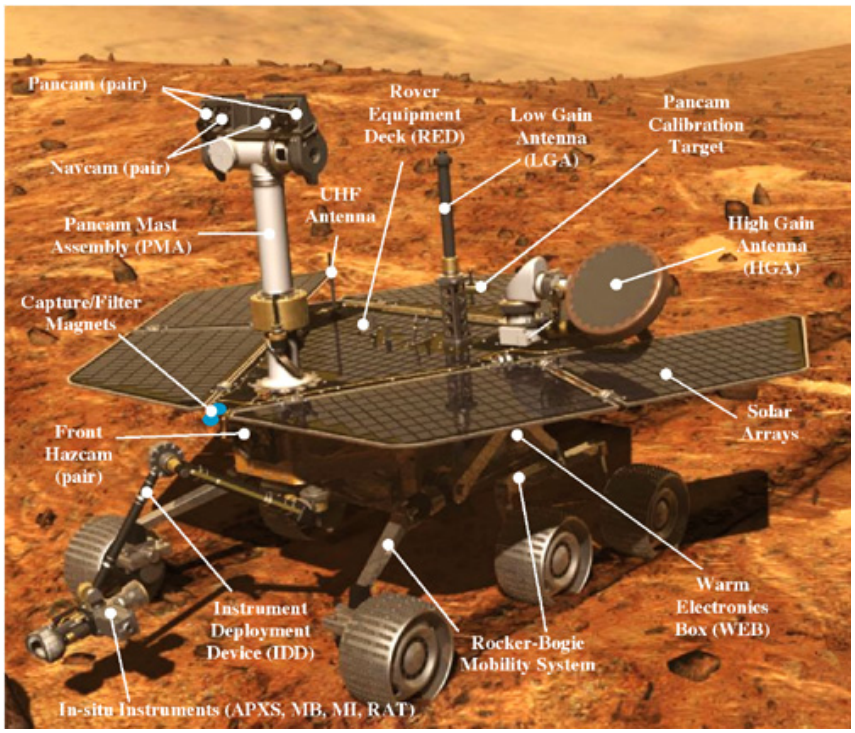
1 ICER står højest sandsynligt for noget i retning af *Image Compression for the Exploration Rovers*.

samtlige opgaver, og det at komprimere billeder må på grund af opgavens hyppighed ikke tage længere tid end allerhøjst nødvendigt. Aaron Kiely formulerede det i vores samtale med ham således:

> ... it really effects what they can do. If they have to stop and wait until the rover compresses all its images before it can move another several meters, that's time it's not doing anything except sitting there compressing images. (Kiely; 2005, side 39 i INTERVIEWS)

Betydningen af den ringe processorkraft bliver kun endnu mere udtalt i kraft af den begrænsede mængde tid og energi, der er til rådighed for roverne. Roverne er drevet af solenergi og befinder sig derfor i en dvaletilstand om natten. Derfor skal alle roverens opgaver udføres i løbet af Marsdøgnets dagtimer. Denne diskussion vender vi tilbage til i forbindelse med kanalkodningen i MER, som i endnu højere grad er underlagt energi-begrænsningerne.

Figur 5.1 illustrerer placeringen af diverse instrumenter på roverne, herunder kameraer, antenner og solpaneler.



Figur 5.1 Mars Exploration Rover. http://nssdc.gsfc.nasa.gov/planetary/image/mer_diagram.jpg

5.2 Overbliksbillede

For billeddata fra Mars og det ydre rum i det hele taget, vil man ofte være nødsaget til at vælge mellem, hvor mange billeder, man vil have ned, og hvor god billedernes kvalitet skal være. Dette 'trade-off' betyder, at en effektiv komprimeringsalgoritme spiller en hovedrolle, fordi man ikke bare kan sende alle billeder med tabsfri komprimering. Der er simpelthen ikke tid nok. Man må acceptere et vist tab. (Forchhammer; 2004, side 9 i INTERVIEWS)

Til komprimering af billeddata, for hvilke man er villig til at acceptere tab, anvendes JPLs egen internt udviklede algoritme *The ICER Progressive Wavelet Image Compression*, der er en waveletbaseret algoritme. ICER, der beskrives i det følgende, kan i princippet også benyttes til tabsfri komprimering, men man anvender i MER en anden algoritme til dette.

Den tabsfri algoritme er en modificeret udgave af den kommercielle algoritme *LOCO-I* (Low Complexity, Context-Based Lossless Image Compression Algorithm), som er udviklet af Hewlett-Packard Laboratories. Modificeringen er foretaget på JPL for at tilpasse algoritmen de særlige krav, der stilles til rumanvendelser. (Kiely; 2005, side 34 i INTERVIEWS)

5.2.1 ICER overordnet

ICER-algoritmen består ligesom mange andre teknikker til billedkomprimering af en forbehandling, en modellering og en entropikodning, omend denne opdeling ikke helt kan foretages så skarpt i ICERs tilfælde. Forbehandlingen i ICER består af en wavelettransformation, som egentlig ikke er én transformation, men en samling af transformationer, som en ICER-bruger frit kan vælge imellem. Fælles for disse transformationer er, som vi skal se, at de indeholder et prædiktivt element, hvilket betyder, at modelleringen i en vis grad også finder sted i selve transformationen. Et andet fælles træk er, at alle transformationerne er invertible. ICERs transformationer muliggør derfor tabsfri komprimering.

ICER gør brug af en særlig form for komprimering kaldet komprimering med *progressiv gengivelseskvalitet*[2]. Ved en sådan komprimering sendes først en lavkvalitets-tilnærmelse til billedet. Efterfølgende transmitteres bit, som gradvist øger billedets kvalitet. Når alle bit er transmitteret, er det rekonstruerede billede lig det oprindeligt sendte. Komprimering med tab opnås ved at stoppe bit-transmissionen, inden alle bit er sendt. På den måde understøtter ICER både komprimering med og uden tab.

ICER anvender en *kontekst-modellering*, som vedligeholder en statistisk model, der bruges til at estimere sandsynligheden for, at den næste bit som skal indkodes er nul.

Sandsynlighedsestimaterne fra kontekst-modelleringen anvendes i den efterfølgende entropi-indkodning, hvor bit alt efter deres tilhørende sandsynlighedsestimater placeres i forskellige 'beholdere'. Bit fra forskellige beholdere indkodes efterfølgende med forskellige koder. ICER anvender såkaldt *sammenflettet entropikodning*[3], som er en teknik, hvor adskillige komponent-kildekoder kombineres. ICERs komponent-koder er fortrinsvist Golomb-koder.

2 Fra engelsk: Progressive fidelity compression.
3 Fra engelsk: Interleaved entropy coding.

ICER benytter sig af såkaldt *fejlinddæmning*[4], som er en teknik, i hvilken man forsøger at tage højde for pakketab og -korruption under transmissionen. I og med at ICER er baseret på kontekst-modellering afhænger afkodningen af en given bit af den allerede foretagne afkodning. Fejlinddæmning begrænser afhængighederne i afkodningen til mindre segmenter af det oprindelige billede, hvorfor kun disse dele påvirkes af et eventuelt tab af pakker.

Den følgende gennemgang af de forskellige dele af ICER er, med mindre andet er angivet, baseret på Kiely & Klimesh (2003).

5.3 Forbehandling

ICER gør som nævnt brug af en hel samling af wavelettransformationer, som vi vil karakterisere i dette afsnit. Transformationerne er nemmest at forstå matematisk ved at sammenligne dem med Haar-transformationen, som er beskrevet i kapitel 3 i Indførelsen. Især lifting-tilgangen vil vi gøre brug af i det følgende. Matematisk set er den forbehandling, der foregår i ICER, ikke særlig elegant eller nem at beskrive. Den er valgt, fordi den er effektiv og nem at implementere, og således ikke valgt udfra matematisk æstetiske betragtninger. Derfor er det også svært at give en sådan kort beskrivelse af forbehandlingen.

ICERs transformation er en *sekventiel* transformation, hvilket betyder, at transformation af et to-dimensionalt billede foretages ved to på hinanden følgende en-dimensionale transformationer på rækker og søjler i billedet. Et billede repræsenteres ved en matrix, som ved transformationen afbildes i en anden matrix med samme dimension som den oprindelige.

For en matrix (et billede med) dimension[5] $N \times N$ beregnes for hver række x i billedet to vektorer l, d med længde $M = {}^{N}/_{2}$ givet ved

$$l(n) = \left\lfloor \frac{1}{2}\big(x(2n) + x(2n+1)\big) \right\rfloor, \tag{5.1}$$

$$d(n) = x(2n) - x(2n+1). \tag{5.2}$$

Ovenstående ses ved sammenligning at svare til en førstetrins Haar-transformation uden normering. De ${}^{N}/_{2}$ koordinater, som er indeholdt i l, svarer til at tage parvise gennemsnit af rækkens pixels, og d svarer til differenser. Data indeholdt i l kaldes for lavpas-uddata (*low-pass output*) for transformationen, og d kaldes højpas-uddata (*high-pass output*). Lavpas-uddata svarer til de koordinater i Haar-transformationen, som stammer fra de indre produkter (hvis man ser bort fra normering) mellem x og de lavfrekvente vektorer, mens højpas-uddata svarer til de koordinater, der stammer fra de højfrekvente vektorer. Dette er beskrevet i Indførelsen.

For højpas-uddata foretages endnu en operation. Først bestemmes

$$r(n) = l(n-1) - l(n). \tag{5.3}$$

Ovenstående benyttes til at beregne

$$h(n) = d(n) - \left\lfloor \alpha_{-1} r(n-1) + \alpha_0 r(n) + \alpha_1 r(n+1) - \beta d(n+1) + \frac{1}{2} \right\rfloor, \tag{5.4}$$

4 Fra engelsk: Error containment.

5 Alle MER-kameraer producerer 1024×1024 billeder med 12 bit per pixel. (Kiely & Klimesh; 2003, side 3)

hvor $\alpha_{-1}, \alpha_0, \alpha_1, \beta$ er konstanter, som tillader, at der kan skiftes mellem forskellige transformationer. De samlede uddata fra transformationen bliver så de $N/2$ koordinater stammende fra l, samt de $N/2$ koordinater stammende fra h. Hvis man studerer h nøje, ser man, at h indeholder informationer om både parvise gennemsnit for omkringliggende pixels samt om den foregående parvise pixeldifferens. Vektoren h angiver faktisk differensen mellem højpas-uddata $d(n)$ og en forudsigelse af disse uddatas størrelse (sidste led i ligning (5.4)). Transformationen indeholder altså en forudsigelse af pixelværdier, det vil sige, at der foregår en vis modellering allerede i forbehandlingen. (Said & Pearlman; 1993, side 665)

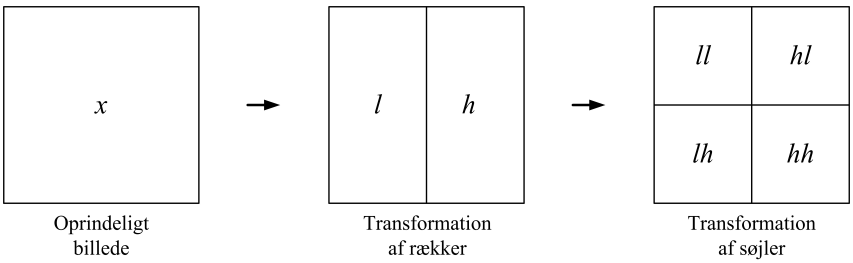
Valget af værdier for konstanter i bestemmelsen af h bestemmer, i hvor høj grad omkringliggende pixelværdier skal påvirke forudsigelsen af den nuværende højpas-værdi. Valgets konsekvenser for komprimeringen er undersøgt empirisk i blandt andet Said & Pearlman (1993) og Adams & Kossentini (2000). I praksis kan vælges mellem 7 forskellige konstant-sæt. Disse er specificeret i Kiely & Klimesh (2003).

Randen af et billede behandles på særlig vis. Denne behandling vil vi ikke inddrage, da den fortrinsvist imødekommer praktiske problemer.

Afrundingen i ligning (5.1) og ligning (5.4) foretages med det formål, at transformationen bliver af typen *heltal-til-heltal*, det vil sige, at transformation producerer uddata, der består af vektorer med heltal. Dette løser en masse praktiske problemer vedrørende plads til repræsentation af pixelværdier, fordi der kun skal repræsenteres heltallige værdier. Ydermere reduceres algoritmens kompleksitet væsentligt. Afrundingen betyder naturligvis, at ICERs transformation bliver ikke-lineær. Transformationen svarer dog tilnærmelsesvis til et lineært filter. (Kiely & Klimesh; 2003, side 2, 5), (Said & Pearlman; 1993, side 665)

Efter transformationen af rækkerne foretages en transformation af billedets søjler. Transformationen har dermed inddelt det oprindelige billede i fire områder, som typisk kaldes *subbands*. De fire subbands, som hver har dimensionen $N/2 \times N/2$ er vist på figur 5.2 og figur 5.3.

Fra definitionen af transformationen kan vi se, at subbandet ll fremkommer ved for hver 2×2 pixelblok i det oprindelige billede at finde den gennemsnitlige pixelværdi og give én pixel i ll denne værdi. Subbandet gengiver altså det oprindelige billede i en opløsning, der er halvt så god.



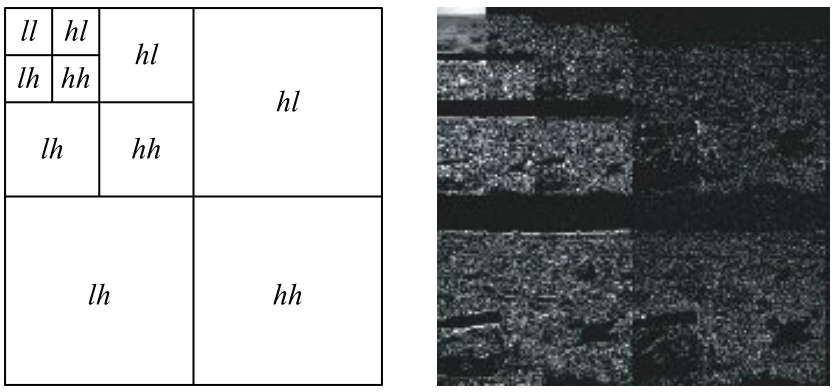
Figur 5.2 Figuren viser, hvordan den sekventielle transformation påføres et billede. Først transformeres rækker, derefter søjler. Vektoren l står for lavpas-uddata, og h står for højpas-uddata. (Said & Pearlman; 1993, side 666)

Figur 5.3 Til venstre: Originalt billede af Pathfinders rover Sojourner. I midten: Billedet efter en horisontal transformation. Til højre: Billedet efter både horisontal og vertikal transformation. (Kiely & Klimesh; 2003, side 8)

Lysintensiteten[6] for det transformerede billede giver et praj om energiens fordeling; jo større intensiteten er, jo højere er energien. Det transformerede billede i figur 5.3 illustrerer, at ICERs transformation koncentrerer energien i *ll*-subbandet.

Transformationen foretages nu igen på *ll*-subbandet. Ved gentagne gange at transformere det fremkomne *ll*-subband opnås en hierarkisk struktur, som er illustreret på figur 5.4. Strukturen, som for ICER kan have op til syv niveauer, kaldes en *pyramidestruktur*.

ICERs sekventielle pyramidestruktur betyder, at algoritmens transformation adskiller sig fra Haar-transformationen, for hvilken der kun foretages én horisontal transformation efterfulgt af én vertikal transformation. For ICERs transformation foretages del-transformationerne skiftevis. Denne skiften, samt afrundingsprocedurerne giver rent matematisk ICER-transformationerne nogle mindre pæne egenskaber. Udover ikke-lineariteten gælder, at det lineære filter, som transformationen tilnærmer, er ikke-ortogonalt. (Kiely & Klimesh; 2003, side 15)



Figur 5.4 Til venstre: Pyramidestrukturens subbands er illustreret. Til højre: Pyramidestrukturen for billedet af Sojourner. (Kiely & Klimesh; 2003, side 8)

6 En stor lysintensitet svarer til høje pixelværdier i billedets matrix-repræsentation. Pixelværdien 0 svarer til farven sort.

Når billedet skal rekonstrueres, skal alle skridt i ICERs transformation på grund af transformationens ikke-linearitet foretages i omvendt rækkefølge. Først bestemmes $r(n)$ ved hjælp af ligning (5.3). Dernæst kan $d(n)$ bestemmes ved at vende ligning (5.4) om. Så kan $l(n)$ bestemmes fra ligning (5.1), og tilsidst kan x bestemmes.

I INDFØRELSEN har vi i sætning 3.15 om *perfekt rekonstruktion* med vilje vist sætningen uden krav om ortonormalitet, fordi ICER ikke opfylder dette krav. At ICERs transformation giver perfekt rekonstruktion er ensbetydende med, at den er invertibel og dermed, at den kan benyttes både til komprimering med og uden tab. I det følgende vil vi diskutere, hvordan energi-koncentrationen i *ll*-subbandet fører til en simpel, men effektiv komprimeringsmetode.

5.4 Progressiv komprimering

Den overordnede strategi ved *progressiv* billedkomprimering er at transmittere en udvalgt del af billedets data og konstruere en tilnærmelse til billedet med denne delmængde som grundlag. Hvis tilnærmelsen ikke lever op til forventningerne kan mere data transmitteres, og tilnærmelsen kan forfines. Denne proces gentages indtil billedet møder de krav, der stilles.

Progressiv komprimering kan groft sagt deles i to grene: Komprimering med progressiv *opløsning* og komprimering med progressiv *gengivelseskvalitet*. Ved komprimering med progressiv opløsning bruges de transmitterede data til at genskabe billedet i en opløsning som forøges, hver gang mere data modtages. Det subband fra sidste afsnit, som blev betegnet *ll*, er netop en sådan lavopløsnings-tilnærmelse til det oprindelige billede.

Ved komprimering med progressiv gengivelseskvalitet komprimeres og transmitteres data, som gør det muligt at konstruere en tilnærmelse til billedet i fuld opløsning, men med en lav gengivelseskvalitet. Dette svarer egentlig til at transmittere et 'uskarpt' billede. De næste data, der modtages, bruges så til at øge gengivelsen eller 'stille skarpt'. Denne komprimeringsteknik udnyttes af ICER og er illustreret på figur 5.5.

Udfordringen ved komprimering med progressiv gengivelseskvalitet er for en given billedkvalitet at afgøre, hvilken del af de resterende data, der

Figur 5.5 Figuren viser, hvordan billedkvaliteten øges under komprimering med progressiv gengivelseskvalitet, når stadig flere data modtages. Fra venstre mod højre ses et billede af Sojourner med $0,125$; $0,25$ og $0,5$ bit per pixel. Læg mærke til hvordan detaljer på hjulene, i sandet og på stenen træder frem. (Kiely & Klimesh; 2003, side 2)

vil øge kvaliteten for det rekonstruerede billede mest muligt. Umiddelbart synes en diskussion af billedkvalitet noget subjektiv, men vi kan indføre et mål for tilnærmelsens *forvrængning*[7], som kan gøre denne diskussion mere matematisk.

Definition 5.1
Lad M være en matrix over $\mathbb{Z}$ og lad $\bar{M}$ være en tilnærmelse til M. Middelkvadratfejlen $\mathcal{M}$ *mellem M og $\bar{M}$ er da givet ved*

$$\mathcal{M} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |m_{ij} - \bar{m}_{ij}|^2.$$

Middelkvadratfejlen kan anvendes som et mål for den forvrængning af et billede, der opstår under komprimering. Jo mindre middelkvadratfejlen er, jo mindre er tilnærmelsens forvrængning, og jo bedre er kvaliteten derfor. Høj kvalitet er altså ensbetydende med, at M og $\bar{M}$ er næsten ens. Det gælder derfor om at komprimere og transmittere data, som giver den største reduktion af middelkvadratfejlen.

Det er ikke i praksis muligt at foretage en matematisk analyse, som giver en optimal løsning, da problemet i høj grad er afhængigt af det konkrete billede. Man kan give nogen udmærkede kvalitative retningslinier, men baserer typisk også sit valg på empiri. For ICER foregår den progressive del af komprimeringen som beskrevet i det følgende delafsnit.

5.4.1 Bitplaner

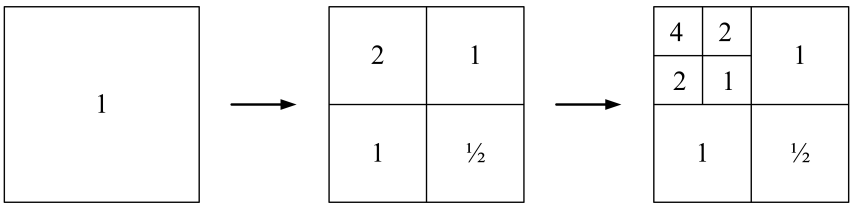
Som vi tidligere har nævnt, er størstedelen af et billedes energi (lysintensitet) typisk lokaliseret i det lavfrekvente subband *ll*. Dette subband indeholder altså de største pixelværdier og vil derfor bidrage mest til at reducere middelkvadratfejlen mellem det oprindelige og det tilnærmede billede. Det synes derfor naturligt at lade udgangspunktet for en tilnærmelse til billedet være *ll*-subbandet.

Som et indledende trin beregnes først middelværdien af de pixels, som befinder sig i subbandet *ll*. Dette er vist på figur 5.4. Denne middelværdi fratrækkes hver pixelværdi i subbandet, som derved kommer til at indeholde både positive og negative værdier. Herved kommer værdierne i dette subband til at ligne værdierne i de andre subbands. Middelværdien transmitteres separat.

For hver pixel i det transformerede billede hører et binært tal på formen $b = b_0 b_1 b_2 \dots b_k$, hvor b_0 angiver tallets fortegn og b_i, $i > 0$ angiver den i'te bit. Bitten b_1 kaldes den mest betydende bit (*most significant bit*), b_2 kaldes den næstmest betydende bit og så videre. For hvert subband hører en matrix

$$M = \begin{pmatrix} m_{00} & m_{01} & \dots & m_{0n} \\ m_{10} & m_{11} & \dots & m_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n0} & m_{n1} & \dots & m_{nn} \end{pmatrix},$$

7 Begrebet rate-forvrængning stammer fra Shannon (1959) og diskuteres indgående i Berger (1971).



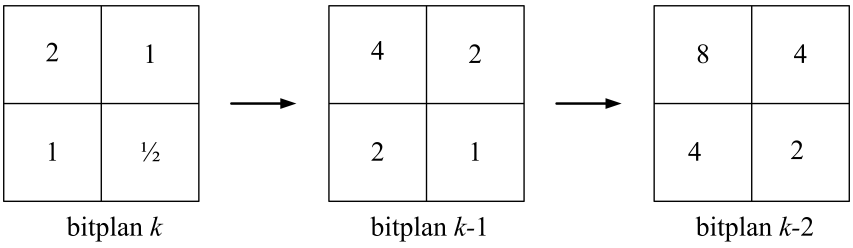
Figur 5.6 Figuren viser, hvordan transformationen og prioriteringen forløber. (Kiely & Klimesh; 2003, side 15)

hvor m_{ij} er et binært tal på ovenstående form. Lad B_ℓ være matricen, hvis i, j'te element er lig den ℓ'te mest betydende bit i m_{ij}. Da kaldes den binære matrix B_ℓ for det ℓ'te mest betydende bitplan for subbandet. Til hver matrix, som indeholder binære tal bestående af k bit, hører altså k bitplaner. Matricen B_1 indeholder de mest betydende bit.

I Said & Pearlman (1996) er de forskellige subbands relative effekter på det rekonstruerede billede undersøgt og på baggrund heraf er foretaget følgende prioritering af subbands: Som udgangspunkt tildeles pixels i billedet alle prioriteten 1. Efter hvert trin i transformationen multipliceres prioriteten for en pixel i *ll*-subbandet med 2. Pixelprioriteten i *hh*-subbandet multipliceres med en halv. Denne proces er illustreret på figur 5.6.

Målet med prioriteringen er at fastlægge hvilke værdier i den transformerede matrix, der skal transmitteres først. Man ønsker naturligvis at transmittere værdier, som har stor betydning for det rekonstruerede billede først. Subbandet *ll* prioriteres derfor højest.

Efter prioriteringen af de forskellige subbands foretages en prioritering af bit i disse subbands. Overordnet set prioriteres betydende bit højere end ikke-betydende bit. Betydningen for det rekonstruerede billede falder omtrent med en faktor to, når man bevæger sig fra det i'te mest betydende bitplan til det $i+1$'te mest betydende bitplan. Dette medtages i prioriteringen ved for bitplanet med de næst*mindst* betydende bit at multiplicere prioriteten med 2. Prioriteten for bitplanet med de tredje-*mindst* betydende bit multipliceres med 4 og så videre. Dette er vist på figur 5.7 for et billede med k bitplaner.



Figur 5.7 Figuren viser den relative prioritering af bitplanerne $k, k-1, k-2$ for et billede, som er transformeret én gang.

Figur 5.7 illustrerer, at det i'te bitplan i ll-subbandet på trin m prioriteres lige så højt, som det $i-1$'te bitplan i hl- og lh-subbandet og det $i-2$'te bitplan i hh-subbandet på trin m. Når komprimeringen påbegyndes, komprimeres altså først bit fra bitplanet med de mest betydende bit i ll-subbandet på højeste trin. Herefter vil der generelt blive komprimeret bit fra forskellige subbands og bitplaner på forskellige transformationstrin, men hele tiden kun de bit, som har den højeste prioritet. Prioriteringen af bit er faktisk ikke afhængig af billedet, men kun af antallet af transformationer, der foretages under opbygningen af den hierarkiske struktur.

De data, som transmitteres og løbende modtages, føjes til de allerede modtagne data, og der foretages en invers transformation, således at det tilnærmede billede træder frem.

5.4.2 Brug af progressiv komprimering i MER

De billeder, der tages af roverne, anvendes i mange forskellige sammenhænge, og der stilles derfor varierende krav til deres kvalitet. For en ikke-progressiv algoritme vil man være nødt til at estimere, hvor mange bit per pixel, der sikrer, at kvaliteten bliver god nok, og give besked om, at man ønsker billedet komprimeret til denne rate. For ikke at spilde dyrebar energi og operationstid ved en eventuel gentransmission er man nødt til at give et tilpas pessimistisk estimat. I mange tilfælde vil dette resultere i en kvalitet, som overstiger det egentlige krav, og dermed et ressourcespild.

For en algoritme, der muliggør komprimering med progressiv gengivelseskvalitet, er man *ikke* nødt til at foretage det nævnte estimat. I stedet kan man angive en grænse for middelkvadratfejlen, og bede om at få billedet komprimeret så dette kvalitetsmål er opfyldt. MER-roveren vil så komprimere billedet og begynde at transmittere data, indtil kvalitetsmålet er nået. På den måde sparer man dyrebar energi og tid, som kan anvendes til andre formål.

Roveren kan hele tiden 'se' både det oprindelige og det komprimerede billede, og det er den, der afgør, hvornår kvalitetsmålet er nået.

I MER-missionen angives faktisk både en øvre grænse for bit per pixel og en nedre grænse for forvrængning. Om dette fortalte Aaron Kiely:

> The scientists specify whether they want lossless compression through LOCO, whether they want ICER and if they want ICER then what bit rate and quality level they are going to require. [...] ... once you are able to meet this quality level then that is good for my scientific purpose and you don't need to spend any more bits. You can move along to the other image. On the other hand, I can't afford any more than say 4 bits per pixel so once it meets 4 bits per pixel it is time to stop and move on to the other image even if you haven't met the quality on it. So that provides some flexibility in being able to trade the rate and distortion here. (Kiely; 2005, side 34, 37 i INTERVIEWS)

Billeder taget af roverne har som udgangspunkt en opløsning på 1024×1024 pixels med 12 bit per pixel. Billederne er alle med undtagelse af dem fra panoramakameraet, i gråtoner. For hvert billede produceres en 64×64 'thumbnail' svarende til ll-subbandet efter fire på hinanden følgende ICER-

transformationer. Denne thumbnail komprimeres til omtrent 650 byte og transmitteres sammen med en 'header' på 682 byte, der indeholder anden information om billedet[8]. På Jorden udvælges så hvilke billeder, der ønskes transmitteret i højere kvalitet. (Kiely; 2005, side 34 i INTERVIEWS)

Vi vil nu vende tilbage til det transformerede billede og se på, hvad der sker under den efterfølgende modellering.

5.5 Kontekst-modellering

ICERs *kontekst-modellering* kan beskrives som en fortløbende proces, der foregår samtidig med selve entropi-indkodningen. Modelleringen består først af en *kategorisering* af pixels fra det wavelet-transformerede billede og dernæst af en *kontekst-klassificering* af bittene, som tilhører disse pixels. En bit, der skal indkodes, tildeles en af flere forskellige kontekster baseret på værdierne af tidligere indkodede bit. Altså modelleres det transformerede billede ved en Markov-kilde (se kapitel 4 i INDFØRELSEN). Ideen i ICERs modellering er at tildele kontekster således, at bit med samme kontekst har omtrent samme sandsynlighed for at være nul, og at entropikoderen udfra en bits kontekst vil kunne give et fornuftigt estimat af denne sandsynlighed.

Pixels fra det transformerede billede inddeles i fire *kategorier*, nummereret 0, 1, 2 og 3. Under entropi-indkodningen af bitplaner holder ICER styr på kategorien for hver pixel. En pixels kategori sammenfatter information om de allerede indkodede bit i denne pixel. Pixelværdier i ICER er repræsenteret ved en *fortegnsbit* og elleve *numeriske* bit, det vil sige bit, som angiver selve pixelværdien.

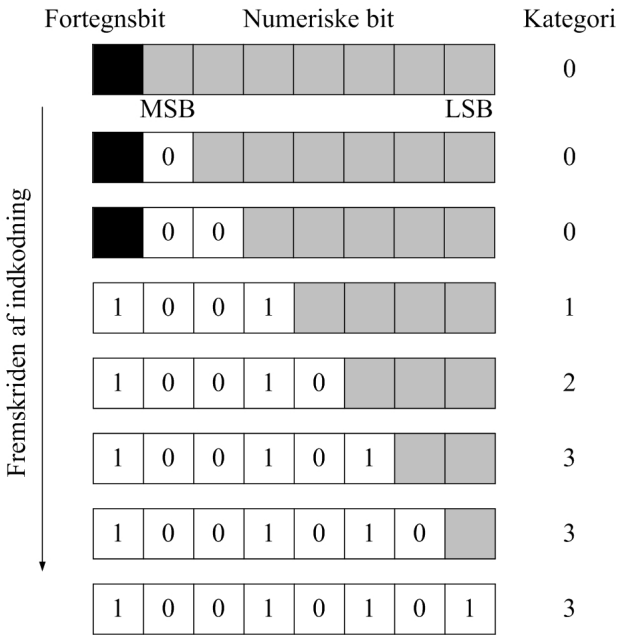
ICERs fire pixel-kategorier er et udtryk for, hvor betydningsfuld eller *signifikant* en pixel er. Hvis der i en pixel forefindes allerede indkodede numeriske bit forskellige fra nul, siges denne pixel at være signifikant. I modsat fald siges den at være *ikke-signifikant*. En pixel tildeles kategori 0, hvis den endnu ikke indeholder indkodede bit, eller hvis den endnu ikke er signifikant. Efter indkodning af en pixels første 1-bit stiger pixelens kategori til 1. Når den næste numeriske bit indkodes stiger kategorien til 2, og når den tredje numeriske bit indkodes stiger kategorien til 3. Det forbliver den herefter. Lad os se et eksempel.

Eksempel 5.2
Figur 5.8 viser, hvordan en pixel tildeles kategorier, efterhånden som pixelens fortegnsbit og numeriske bit indkodes.

Den viste pixel har værdien -21, og en 8-bit repræsentation er for overskuelighedens skyld antaget; én fortegnsbit (sort) efterfulgt af syv numeriske bit (grå). En bit, som endnu ikke er indkodet, er farvelagt. Mellem hver af de viste tilstande indkodes en numerisk bit som del af indkodningen af subband-bitplanen. ◇

Efter en pixels kategorisering finder kontekst-klassificeringen af denne pixels bit sted. Pixelens bit klassificeres i en af sytten forskellige kontekster, som er nummereret fra 0 til 16.

8 Headeren indeholder blandt andet middelværdien for pixels i *ll*-subbandet, som jo er fratrukket alle værdierne inden transmission. Dette er beskrevet tidligere.



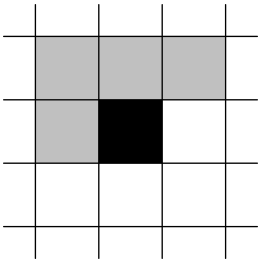
Figur 5.8 Figuren viser, hvordan en pixels kategorier ændrer sig, efterhånden som fortegnsbit og numeriske bit indkodes. En endnu ikke indkodet bit er farvelagt. Fortegnsbitten er sort, og numeriske bit er grå. MBS angiver 'Most Significant Bit' og LBS angiver 'Least Significant Bit'. (Kiely & Klimesh; 2003, side 17)

Konteksten for en bit i en pixel bestemmes udfra de allerede indkodede bit i denne pixel samt pixelens otte naboer fra samme del af subbandet. Dette er illustreret på figur 5.9. Med andre ord afhænger en bits kontekst altså af allerede indkodede bit i i alt ni pixels.

Konteksten af en bit bestemmes udfra kategorien af den pixel, den tilhører, samt af signifikansen og fortegnet af de otte nabopixels. Hvis ikke alle otte nabopixels er til rådighed, fordi pixelen, der skal indkodes, ligger på randen af subband-segmentet, betragtes de manglende pixels som værende ikke-signifikante.

Vi begynder med at beskrive konteksten af bit for pixels i kategori 0. Kontekst-klassificeringen sker på baggrund af tre parametre h, v og d. Hvis det subband, der indkodes, *ikke* er et hl-subband, sættes h lig antallet af signifikante *horisontale* nabopixels ($0, 1$ eller 2). Parameteren v sættes lig antallet af signifikante *vertikale* nabopixels ($0, 1$ eller 2), og d sættes lig antallet af signifikante *diagonale* nabopixels (0 til 4). For et hl-subband er rollerne af h og v byttet om, hvilket ombytter kontekst-skabelonen. Når h, v og d er fastlagt, tildeles den pågældende bit en kontekst. Hvis bitten tilhører en pixel i et ll-, lh- eller hl-subband tildeles en kontekst, der er bestemt som angivet i tabel 5.1. Hvis bittens pixel tilhører et hh-subband, tildeles en kontekst, som angivet i tabel 5.2. Bit, der tilhører pixels i kategori 0, tildeles altså kontekster fra 0 til og med 8.

Hvis en bit tilhører en pixel i kategori 1 tildeles den enten kategori 9 eller 10. Hvis ingen af dens horisontale eller vertikale nabopixels er



Figur 5.9 Når en bit i den sorte pixel skal indkodes, bestemmes konteksten af bitten udfra de allerede indkodede bit i de ni viste pixels. Numeriske bit fra det *pågældende* og de mere signifikante bit-planer (MSB-planer) er tilgængelige i de grå pixels. For de hvide pixels samt den sorte pixel er *kun* bit fra (MSB-planer) tilgængelige. (Kiely & Klimesh; 2003, side 16)

signifikante tildeles den kontekst 9 og ellers 10. En bit, der tilhører en pixel i kategori 2, tildeles altid kontekst 11.

Bit for pixels i kategori 3 regnes for ukomprimerbare, da estimaterne af disse bits sandsynlighed for at være nul oftest er meget tæt på $1/2$. Denne sandsynlighed vil vi herefter benævne $\mathscr{S}_0$. Bit i kategori 3 indkodes ikke af ICER. I stedet sendes de direkte til 'ukodet'-beholderen for entropikoderen, som beskrives i afsnit 5.6.

Kontekster 12-16 er forbeholdt fortegnsbit. Indkodning af fortegnsbit, som er beskrevet i (Kiely & Klimesh; 2003, side 17-18), er ikke væsentligt forskellig fra indkodning af numeriske bit, og vi vil derfor undlade at beskrive den.

	$h=0$			$h=1$		
d	$v=0$	$v=1$	$v=2$	$v=0$	$v>0$	$h=2$
$d=0$	0	3	4	5	7	8
$d=1$	1	3	4	6	7	8
$d\geq 2$	2	3	4	7	7	8

Tabel 5.1 Konteksttildeling for bit, som tilhører en pixel med kategori 0. Tildelingen gælder for *ll*-, *lh*- og *hl*-subbands og afhænger af h, v og d. (Kiely & Klimesh; 2003, side 17)

d	$h+v=0$	$h+v=1$	$h+v\geq 2$
$d=0$	0	1	2
$d=1$	3	4	5
$d=2$	6	7	7
$d\geq 3$	8	8	8

Tabel 5.2 Konteksttildeling for bit tilhørende pixels i kategori 0 for *hh*-subbands. Tildelingen afhænger af h, v og d. (Kiely & Klimesh; 2003, side 18)

5.5.1 Sandsynlighedsestimering

For hver kontekst fører ICER et regnskab over antallet af nul-bit samt det totale antal af bit, der har optrådt i denne kontekst. Forholdet mellem den totale optælling og optællingen af nul-bit repræsenterer $\mathscr{S}_0$-estimatet for den pågældende kontekst. For hver bit, der skal indkodes, modtager entropikoderen $\mathscr{S}_0$-estimatet i form af forholdet mellem optællingerne.

Hver konteksts optællinger tildeles fra begyndelsen værdier svarende til $\mathscr{S}_0 = 1/2$. Når en bit træffes i en kontekst forøges kontekstens totale antal, og hvis bitten er nul, forøges kontekstens antal af nul-bit. Optællingen finder først sted, når bitten er indkodet. Når det totale antal når en på forhånd specificeret værdi, omskaleres begge optællinger ved at dividere tallene med 2. Effekten af omskaleringen udmønter sig i $\mathscr{S}_0$-estimater, som giver højere vægt til nyligt indkomne bit, og som derfor til en vis grad imødekommer kontekst-statistikker, der ændrer sig efterhånden som komprimeringen skrider frem[9] – altså en form for adaptivitet.

For en følge af kilde-bit $b_1, b_2, \ldots$ producerer ICERs kontekst-model for hver bit b_i således et sandsynlighedsestimat $\mathscr{S}_{0_i}$, for at denne bit er nul. Entropikoderen anvender disse estimater til at producere en indkodet strøm af bit, fra hvilken den oprindelige følge af bit kan rekonstrueres. Dette skal vi se nærmere på i næste afsnit.

5.6 Entropikoderen

Vi skal her beskrive den entropikoder, som ICER benytter sig af til at komprimere fortegnsbit og numeriske bit.

ICER er baseret på en adaptiv entropikoder. En adaptiv indkodning giver et fordelagtigt samspil mellem entropikoderen og kontekst-modellen: Et sandsynlighedsestimat $\mathscr{S}_{0_i}$ for en bit b_i kan afhænge af værdierne af de foregående bit og ændre sig løbende med hver ny-indkodet bit. Dette bevirker, at kontekst-modelleringen hurtigt kan tilpasse sig de foranderlige statistikker og dermed gøre bedre brug af den umiddelbare kontekst, som en bit befinder sig i. En sådan tilgang stiller større krav til algoritmen, fordi der på dekomprimeringssiden skal konstrueres de samme estimater som på komprimeringssiden. Ved dekomprimeringen skal værdierne af de første $i - 1$ bit altså bestemmes førend den i'te bit kan afkodes.

Den adaptive entropikoder i ICER gør brug af en hele stribe af forskellige koder, som vi vil beskrive i det følgende.

5.6.1 Komponent-koder i ICER

ICERs entropikoder kombinerer adskillige komponent-kildekoder. Hver komponent-kode er en variabel-til-variabel længde binær kildekode, som afbilder *inddata*-kodeord i *uddata*-kodeord. Indkoderen opdeler følgen af inddata-bit til inddata-kodeord. Indkoderens uddata består af sammenfletningen (som forklares senere) af de tilhørende uddata-kodeord. Afkodning opnås ved at ombytte rollerne af inddata- og uddata-koderne og udføre samme procedure som ved indkodning.

9 I MER-implementeringen af ICER sættes de initiale optællinger af nul-bit til 2, de initiale totale optællinger sættes til 4, og en omskalering finder sted, når den totale optælling når 500.

Der gælder for komponent-koderne, at en følge af bit til entropikoderen kan opdeles til inddata-kodeord på en entydig vis, samt at bitfølgen er *øjeblikkeligt opdelelig*. Det vil sige, at den kan opdeles, så snart den sidste bit i et kodeord er modtaget. Denne egenskaber er sikret af, at komponent-koderne er prefix-koder og samtidig udtømmende koder (se kapitel 2 i INDFØRELSEN).

Størstedelen af komponent-koderne i ICERs entropikoder er Golomb-koder, da disse er effektive til lange følger af bit, der har næsten samme $\mathscr{S}_0$ nær 1. ICERs brug af Golomb-koderne afviger en smule fra den i INDFØRELSEN beskrevne, hvorfor vi kort skal gennemgå den her. Koden $\mathcal{G}_m$, for $m \geq 1$, har $m+1$ inddata-kodeord: $1, 01, 001, \ldots, 0^{m-1}1$ og 0^m (notationen 0^i angiver en følge af i nuller). Inddata-kodeord 0^m afbildes til uddata-kodeordet 1. Til beskrivelse af uddata-kodeordene for de resterende inddata-kodeord sættes[10]

$$\ell = \lceil \log_2 m \rceil \qquad \text{og} \qquad i = 2^\ell - m.$$

Uddata-kodeordet, der svarer til inddata-kodeordet 0^k1, er den ℓ-bit binære repræsentation af heltallet k, for $k < i$, og den $(\ell+1)$-bit binære repræsentation af heltallet $k+i$ ellers. Lad os se på et eksempel for den velkendte $\mathcal{G}_5$.

Eksempel 5.3
Den binære Golomb-kode $\mathcal{G}_5$ giver følgende indkodning:

$$\begin{aligned} 00000 &\mapsto 1 \\ 00001 &\mapsto 0111 \\ 0001 &\mapsto 0110 \\ 001 &\mapsto 010 \\ 01 &\mapsto 001 \\ 1 &\mapsto 000 \end{aligned}$$

Med denne kode opdeles den binære følge

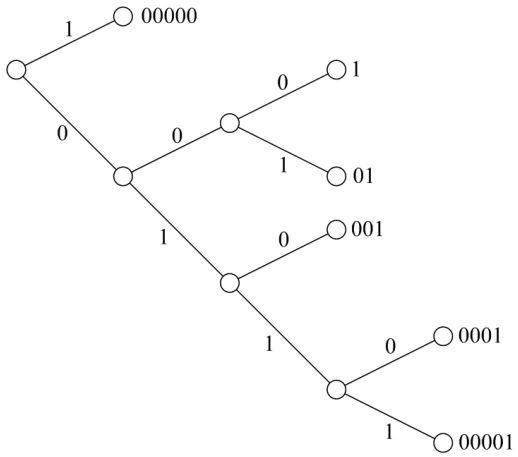
$$0100000001100001$$

til inddata-kodeordene 01, 00000, 001, 1, 00001, der indkodes til følgende uddata-kodeord: 001, 1, 010, 000, 0111. ◇

Den forholdsvis lille komprimering i eksemplet ovenfor skyldes, at koden er bedst til at komprimere følger af bit med mange nuller i træk (00000'er).

Golomb-koden $\mathcal{G}_5$ fra eksempel 5.3, vil have et binært afkodningstræ som det vist på figur 5.10. Hvert inddata-kodeord er tilknyttet en ekstern knude i træet. Et inddata-kodeord tilknyttes et uddata-kodeord, som fås ved at følge stien gennem træet fra roden og ned til den eksterne knude, der repræsenterer inddata-kodeordet. De bit, der mødes på vejen, giver uddata-kodeordet. For inddata-ordet 001 mødes bittene 0-1-0, som giver uddata-kodeordet 010.

10 ICER kunne have reduceret antallet af bit ved at anvende den repræsentation, som er beskrevet i INDFØRELSEN. At ICER benytter ovenstående repræsentation, er muligvis et spørgsmål om kompleksitet.



Figur 5.10 Afkodningstræ for $\mathcal{G}_5$. Uddata-bit er vist langs træets grene. Inddata-kodeord er vist ved træets eksterne knuder. (Kiely & Klimesh; 2003, side 23)

Der anvendes i Kiely & Klimesh (2003) en forkortet notation for afkodningstræet som giver en kompakt specifikation af komponent-koden. En *intern* knude k i træet repræsenteres ved (n_0, n_1), hvor n_0, n_1 er notationen for knuderne k_0, k_1 for enden af henholdsvis 0-grenen og 1-grenen for k. En *ekstern* knude repræsenteres ved dens inddata-kodeord. Notationen for den samlede kode er notationen for *roden* i kodens træ. For $\mathcal{G}_5$, der er vist på figur 5.10, er repræsentationen af roden $(n_0, 0^5)$. n_0 er repræsentationen af knuden k_0 for enden af 0-grenen. For enden af rodens 1-gren er en ekstern knude k_1, som repræsenteres ved dens inddata-kodeord 00000. Den interne knude k_0 har selv en notation, som findes på tilsvarende vis. For $\mathcal{G}_5$ bliver notationen

$$\mathcal{G}_5 \sim \Big(\big((1, 01), (001, (0^3 1, 0^4 1))\big), 0^5\Big).$$

Ovenstående notation kan anvendes til at angive alle ICERs komponentkoder. Det vil vi gøre brug af senere.

5.6.2 ICERs sammenflettede entropikoder

ICERs entropikoder komprimerer data med bitvise sandsynlighedsestimater ved at sammenflette uddata fra flere af de komponent-koder, der tidligere er beskrevet. Disse koder indkoder grupper af bit med ensartede sandsynlighedsestimater.

ICERs sammenfletning af Golomb-koder svarer til teknikken 'Block-Melcode', som er beskrevet i kapitel 5 i Indførelsen. ICERs sammenfletning af de øvrige prefix-koder svarer til 'udvidelsen af Block-Melcode til andre prefix-koder', som er beskrevet i samme kapitel.

Det kan uden tab af generalitet antages, at $\mathscr{S}_{0_i} \geq 1/2$ for hvert index i. Skulle det for visse $\mathscr{S}_{0_i}$ alligevel ikke være tilfældet kan bitten b_i inverteres før indkodning for at opnå et sandsynlighedsestimat tilhørende $[1/2; 1]$. Denne inversion kan duplikeres af afkoderen. Vi koncentrerer os derfor om

sandsynlighedsintervallet $[1/2; 1]$, som opdeles i mindre intervaller. Hvert interval tildeles en *beholder*, som har tilknyttet en komponent-kode. Hvis en bit har et $\mathscr{S}_0$, der tilhører intervallet, placeres den i beholderen. Alle bit, der placeres i beholderen, indkodes med beholderens komponent-kode. Dette illustreres bedst med et eksempel.

Eksempel 5.4
En af beholderne i ICERs entropikoder svarer til sandsynlighedsintervallet $[0,853; 0,881[$. Den tilknyttede kode er Golomb-koden $\mathcal{G}_5$ fra eksempel 5.3. For at få en indikation af, hvor godt denne kode klarer sig på intervallet, kan man evaluere den komprimering koden giver, når den anvendes på en hukommelsesløs binær kilde (symbolgenerator).

Lad $p \in [0,853; 0,881[$ være sandsynligheden for at et nul genereres. Den gennemsnitlige længde l_{ind} af inddata-kodeordene, som er givet i eksempel 5.3, bliver:

$$\begin{aligned} l_{\text{ind}} &= \sum_{i=1}^{6} p(s_i) l_i \\ &= (p^5) \cdot 5 + (p^4(1-p)) \cdot 5 + (p^3(1-p)) \cdot 4 + (p^2(1-p)) \cdot 3 \\ &\quad + (p(1-p)) \cdot 2 + (1-p) \cdot 1 \\ &= p^4 + p^3 + p^2 + p + 1. \end{aligned}$$

Den gennemsnitlige længde l_{ud} af uddata-kodeord som funktion af p bliver så:

$$\begin{aligned} l_{\text{ud}} &= p^5 \cdot 1 + (p^4(1-p)) \cdot 4 + (p^3(1-p)) \cdot 4 + (p^2(1-p)) \cdot 3 \\ &\quad + (p(1-p)) \cdot 3 + (1-p) \cdot 3 \\ &= -3p^5 + p^3 + 3. \end{aligned}$$

Den gennemsnitlige rate er da givet ved

$$r(p) = \frac{-3p^5 + p^3 + 3}{p^4 + p^3 + p^2 + p + 1} \text{ bit per kildebit.}$$

I Kiely & Klimesh (2003) defineres kodens *redundans* ved $r(p) - H_2(p)$, hvor $H_2(p)$ er den binære entropifunktion givet ved

$$H_2(p) = -\left(p \log_2 p + (1-p) \log_2 (1-p)\right).$$

Vi har i INDFØRELSEN defineret redundansen lidt anderledes, men de to udtryk for redundansen er faktisk tilsvarende. Dette kan vises ved en udregning, som er lidt besværlig, og som vi derfor vil undlade at give her.

Ved at udregne kodens redundans for forskellige værdier af p ses[11], at redundansen er mindst, når p tilhører intervallet $[0,853; 0,881[$. Koden egner sig altså *bedst* til indkodning af bit med et $\mathscr{S}_0$ tilhørende dette interval. For en uddybende beskrivelse af analysen af prefix-koder til de forskellige beholdere se Kiely & Klimesh (2001a). ◇

ICERs sammenflettede entropikoder benytter i alt sytten beholdere nummeret fra 1 til 17, som hver har tilknyttet et sandsynlighedsinterval

11 Se eventuelt (Kiely & Klimesh; 2003, figur 10, side 22).

j	z_j	$\mathcal{C}_j$
1	$0,539$	$(0,1)$ ikke-kodet
2	$0,570$	$(((((0^4 1, 1^4), 0^3 1), 001), 10), (01, (110, (0^5, 1^3 0))))$
3	$0,618$	$(((001, ((1101, 0^3 11), 1^3)), 10), (01, (0^4, (1100, 0^3 10))))$
4	$0,665$	$((0^3, 01), (10, (001, 11)))$
5	$0,724$	$(((010, (10^4, 110)), ((101, 011), ((10^3 1, 1^3), 1001))), 00)$
6	$0,765$	$((0^5, 1)((0^3 1, 001), (010, (0^4 1, 011))))$
7	$0,819$	$(0^3, ((001, 010), (100, (11, (011, 101)))))$
8	$0,853$	$(0^4, ((001, 01), (10, (0^3 10, (0^3 11, 11)))))$
9	$0,881$	$\mathcal{G}_5$
10	$0,899$	$\mathcal{G}_6$
11	$0,922$	$\mathcal{G}_7$
12	$0,950$	$\mathcal{G}_{11}$
13	$0,971$	$\mathcal{G}_{17}$
14	$0,985$	$\mathcal{G}_{31}$
15	$0,994$	$\mathcal{G}_{70}$
16	$0,998$	$\mathcal{G}_{200}$
17	1	$\mathcal{G}_{512}$

Tabel 5.3 ICERs sammenflettede entropikoder. Her angiver j indekset af beholderen med komponent-koden $\mathcal{C}_j$. En beholder har tilknyttet et sandsynlighedsinterval $[z_{j-1}, z_j[$ ($z_0 = 1/2$), som er angivet ved den øvre grænse z_j. (Kiely & Klimesh; 2003, side 24)

samt en komponent-kode $\mathcal{C}_j$. Disse koder en angivet i tabel 5.3. Hvis et givet interval og beholder anvender en Golomb-kode til indkodning, har vi i tabellen angivet dennes parameter. Hvis $\mathcal{C}_j$ ikke er en Golomb-kode, har vi angivet koden med den tidligere indførte kompakte notation.

Sandsynlighedsintervallet for beholder j er specificeret ved $[z_{j-1}, z_j[$, ($z_0 = 1/2$). Små værdier af j svarer til sandsynlighedsintervaller tæt på $1/2$ og store værdier af j svarer til intervaller tæt på 1.

Vi vil i det følgende beskrive, *hvordan* indkodning af bit i de forskellige beholdere foregår.

Indkodning

- Indkoderen grupperer bit fra samme beholder j i et inddata-kodeord c_{ind}. For at sikre at afkodning er mulig skal uddata-kodeordene c_{ud} fra de forskellige komponent-koder sammenflettes i den rette rækkefølge. Indkoderen sørger for dette ved at vedligeholde en liste L med inddata-kodeord c_{ind} samt delvist dannede inddata-kodeord c^*_{ind} i en buffer.
- Når en kilde-bit b_i ankommer, placeres den i en beholder, hvis sandsynlighedsinterval indeholder $\mathscr{S}_{0_i}$. Indkoderen kontrollerer om L indeholder et c^*_{ind} for denne beholder. Er dette tilfældet tilføjes b_i til dette c^*_{ind}. I modsat fald tilføjes et nyt delvist dannet inddata-kodeord, som kun indeholder b_i. Dette ord tilføjes sidst på listen.
- Når det *første* ord på L udgør et afsluttet inddata-kodeord c_{ind} for komponent-koden $\mathcal{C}_j$, producerer indkoderen kodens tilhørende uddata-kodeord, hvorefter c_{ind} slettes fra L. Hvis bit-følgen i det *nye* første ord på L ligeledes udgør et afsluttet inddata-kodeord c_{ind},

foretages indkodning igen. Dette fortsættes indtil første ord på L er et ufuldstændigt inddata-kodeord (c^*_{ind}).

- I tilfælde af at bufferen, som indeholder L, bliver fyldt op, føjes en eller flere såkaldte 'flush-bit' til det delvise inddata-kodeord c^*_{ind}, som står først på L, for at danne et afsluttet c_{ind}, der kan indkodes. Flush-bit tilføjes, så koden producerer det kortest mulige uddata-kodeord. Flush-bit bruges i øvrigt også til at fuldende alle tilbageværende c^*_{ind} i L, når følgen af inddata-bit ender.

Afkodning

- Afkodningen er simplere end indkodningen. Afkoderen holder styr på et c^*_{ind} for hver beholder i entropikoderen og hvert sådan ord er et suffix af et c_{ind} for den beholder. Kodeord rekonstrueres i afkoderen i samme rækkefølge, som de stod på indkoderens liste L.
- Kilde-bit afkodes i rækkefølge. For at afkode bit b_i beregner afkoderen det tilknyttede sandsynlighedsestimat $\mathscr{S}_{0_i}$. Denne beregning er generelt baseret på værdierne af de netop afkodede kilde-bit $\ldots, b_{i-2}, b_{i-1}$. Givet $\mathscr{S}_{0_i}$ anvender afkoderen samme procedure som indkoderen til bestemmelse af den beholder j, som b_i skal knyttes til. Hvis afkoderen har et c^*_{ind} for denne beholder, udskiftes den første manglende bit i dette c^*_{ind} med den afkodede bit fra b_i. Hvis ikke, må afkoderen rekonstruere et c_{ind} fra beholderens komponent-kode $\mathcal{C}_j$. Dette gøres ved at oversætte et c_{ud} fra de næste tilgængelig bit i den indkodede bitstrøm til et tilsvarende c_{ind}. Et prefix bestående af bit fra dette kodeord kommer til at udgøre de afkodede bit, og afkoderen husker det hængende kodeordssuffix.
- For korrekt afkodning må afkoderen identificere og slette flush-bit, så de ikke fejlagtigt opfattes som kilde-bit. Dette kan gøres ved en ligefrem procedure. Under afkodningen holder afkoderen styr på antallet af rekonstruerede kodeord. Hver gang et kodeord er rekonstrueret, gemmer afkoderen det umiddelbare antal sammen med kodeordets suffix. Når differensen mellem det nuværende antal og værdien af et givet kodeords suffix overstiger størrelsen af indkoderens buffer, må samtlige bit i dette suffix være flush-bit, og de skal dermed ikke tages i betragtning.

5.7 Fejlinddæmning

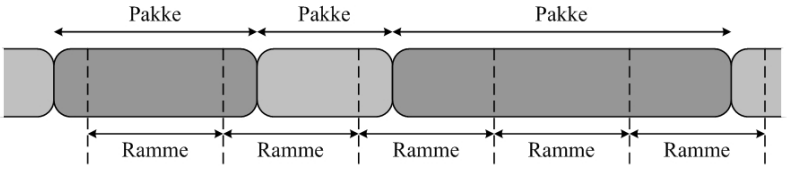
Brug af kontekst-modelleringen øger komprimeringens effektivitet, men til gengæld skabes en afhængighed mellem de indkodede bit. Denne afhængighed betyder, at hvis bit b_i går tabt under en transmission, vil bittene $b_{i+1}, b_{i+2}, \ldots$ ikke kunne afkodes korrekt.

Fejlinddæmning[12] er en teknik, i hvilken et billede opdeles i mindre segmenter med det formål at begrænse en eventuel fejlafkodning til de enkelte segmenter. Hvert segment komprimeres uafhængigt af de andre, hvorfor tab eller korruption af data i et segment ikke påvirker rekonstruktionen af data i de andre segmenter.

Data, der transmitteres over deep-space kanalen, er ordnet i *pakker*. En pakke er en blok af data, hvis størrelse kan variere. Pakkerne trans-

12 Fra engelsk: Error containment.

mitteres via *rammer*, der er data-blokke af fast længde. Da pakkestørrelser er variable og rammelængder faste vil pakker og rammer oftest ikke være sammenfaldende – dette er illustreret på figur 5.11.



Figur 5.11 Illustration af at pakker som regel ikke er sammenfaldende med rammer. (Kiely & Klimesh; 2003, side 25)

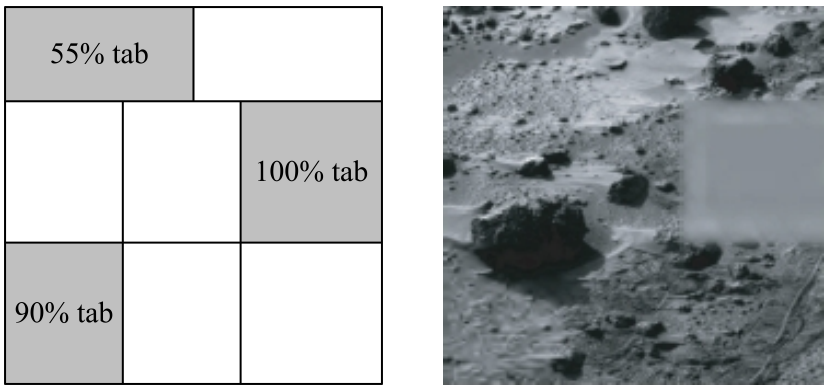
5.7.1 Fejlinddæmning i ICER

Da tabet af en enkelt ramme kan påvirke flere pakker, sørges der i ICER for, at komprimerede uddata arrangeres sådan, at pakker med data fra samme billedsegment følger efter hinanden. Dette reducerer sandsynligheden for, at et rammetab påvirker mere end et enkelt segment. For at sikre at rammerne kan dekomprimeres uafhængigt af hinanden, begynder hvert segment med en header, som indeholder informationer så som segment-indeks, billedets dimensioner og ICERs parametre. Da hvert segment også komprimeres uafhængigt af andre segmenter, vedligeholder ICER separate kontekst-modeller og entropi-indkodningsdata for hvert segment.

ICERs segmentering foregår efter wavelet-transformationen. Hver pixel i det transformerede billede bliver koblet til et segment. Selv om segmenter defineres i det transformerede domæne, svarer hvert segment til en rektangulær del af det oprindelige billede. Hvis der ønskes en opdeling i k segmenter, deles ll-subbandet ind i k rektangulære segmenter og derefter overføres denne opdeling til de andre subbands. Pixels fra forskellige subbands, som har samme rumlige placering i billedet, vil på den måde tilhøre det samme segment.

En konsekvens af at foretage segmenteringen på det transformerede billede er, at grænserne mellem segmenterne i det oprindelige billede bliver uskarpe. Når pixels nær sådanne grænser rekonstrueres, kombinerer den inverse transformation data fra nabosegmenter. Effekten af dette er, at et datatab i ét segment kan påvirke nabosegmentet i det rekonstruerede billede. Det vil sige, at pixels nær grænsen vil forekomme udtværede. En fordel ved denne tilgang er, at man undgår de såkaldte 'blocking artifacts' (se kapitel 3 i Indførelsen), der ellers vil komme til udtryk ved, at segmentinddelingen er synlig på det rekonstruerede billede.

Da hvert segment komprimeres progressivt, vil en vis mængde fejlinddæmning automatisk forekomme indenfor selve segmenterne. En tabt pakke vil blot betyde, at et segment af billedet vil have en dårligere gengivelseskvalitet end resten af billedet. Selvom data, der følger efter et tab, ikke kan anvendes i dekomprimeringen vil et pakketab derfor ikke ødelægge et helt billede. Dette er illustreret på figur 5.12, hvor tre af billedets segmenter har oplevet et pakketab.



Figur 5.12 Eksempel på fejlinddæmning i et billede, som er komprimeret til 1 bit per pixel. Der er simuleret pakketab, som påvirker tre af de otte billedsegmenter. (Kiely & Klimesh; 2003, side 28)

Vi vil ikke her komme nærmere ind på omstændighederne for ICERs fejlinddæmning. Den interesserede læser henvises til Kiely & Klimesh (2003).

5.8 ICERs komprimeringsresultater

I februar måned 2004 fremkom en rapport, (Kiely & Klimesh; 2004), i hvilken de foreløbige komprimeringsresultater for billeder taget af roverne var beskrevet frem til den 7. februar. Vi vil kort gengive rapportens hovedresultater.

Langt størstedelen af de transmitterede billeder var på dette tidspunkt blevet komprimeret med ICER. I alt 5.625 regulære billeder, det vil sige billeder i fuld størrelse[13], var i februar 2004 blevet transmitteret. Af disse var 5.132 komprimeret af ICER. Sammenlagt blev 2.867,9 Mpixel komprimeret af ICER til en datamængde på 387,3 Mbyte[14]. Dette svarer til en gennemsnitlig rate på 1,13 bit per pixel, eller en komprimeringsfaktor på cirka 11. Denne komprimering er ikke et udtryk for ICERs maksimale komprimeringsevne, men derimod et udtryk for de kvalitetskrav, der er udstukket af JPL. (Kiely & Klimesh; 2004, side 3)

For alle billederne, der er nævnt i resultatrapporten, er der foretaget en forbehandling med fire transformationstrin, sådan at *ll*-subbandet på fjerde trin har været en 64×64 pixel gengivelse af det oprindelige billede, svarende til den thumbnail, der er blevet transmitteret. For stort set alle billeder er filteret, som i Kiely & Klimesh (2003) kaldes E blevet anvendt. Filterets koefficienter er givet ved $\alpha_{-1} = 0$, $\alpha_0 = {}^3\!/_{16}$, $\alpha_1 = {}^8\!/_{16}$ og $\beta = {}^6\!/_{16}$.

Ved en sammenligning af rate-forvrængninger ses, at ICERs komprimeringsegenskaber fuldt tilsvarer JPEG-2000s, og at begge disse i øvrigt klarer sig langt bedre end JPEG. For uddybning se (Kiely & Klimesh; 2003, side 40).

13 Næsten udelukkende 512×512 pixels eller 1024×1024 pixels.
14 En Mpixel er 10^6 pixels, og en Mbyte er 2^{20} byte.

5.9 Kort om LOCO-I

Den tabsfri billedkomprimeringsteknik LOCO-I består ligesom ICER af en modelleringsdel og en entropikodningsdel, men adskiller sig ved ikke at have en forbehandling, hvilket er typisk for tabsfri komprimeringsteknikker. LOCO-I, der er baseret på Huffman-kodning, udmærker sig ved sin lave kompleksitet og er derfor også at foretrække frem for ICER til tabsfri komprimering i MER.

Modellen i LOCO-I opererer traditionen tro med betingede sandsynligheder for pixelværdierne i det billede, der ønskes komprimeret. Ligesom ICER gør LOCO-I brug af kontekst-modellering af inddata, før de entropiindkodes. Sandsynlighedstildelingen kan generelt foretages således:

1. Et 'forudsigelses'-skridt i hvilket en værdi $\bar{x}_{i+1}$ gættes for den næste pixel x_{i+1} på basis af en delmængde af de tilgængelige tidligere følger $x^i = x_1, x_2, \ldots, x_i$.
2. Bestemmelsen af en *kontekst*, i hvilken pixel x_{i+1} optræder – det vil sige en funktion af en tidligere delfølge.
3. En sandsynlighedsmodel for *fejl-signalet*, $e_{i+1} = x_{i+1} - \bar{x}_{i+1}$, betinget af konteksten af x_{i+1}.

Fordelingen af fejl-signalet i et 'continuous-tone' billede[15] kan tilnærmes ved en Laplace-fordeling, det vil sige en to-sidet eksponentielt aftagende fordeling centreret i nul. Det antages derfor i LOCO-I, at fejl-signalet kan antage værdier ε:

$$\frac{\alpha}{2} \leq \varepsilon \leq \frac{\alpha}{2} - 1, \tag{5.5}$$

hvor $\alpha = 2^\beta$ er størrelsen af inddata-alfabetet. (Weinberger et al.; 1996, side 1-4)

I en ét-gennemløbsalgoritme kan indkoderen ikke vedligeholde en Huffman-tabel for hver mulig fordeling af fejl-signalerne, eftersom indkodning skal ske øjeblikkeligt. Adaptiv konstruktion af optimale tabeller umuliggøres af begrænsningerne på kompleksiteten. I stedet vælger indkoderen for hver kontekst på adaptiv vis den bedste kode i en begrænset mængde af Huffman-koder svarende til sandsynlighedsfordelinger baseret på tidligere indkodning. (Weinberger et al.; 1996, side 4)

For at udnytte Golomb-kodernes optimalitet i forhold til geometriske fordelinger af ikke-negative heltal til at indkode det Laplace-fordelte fejl-signal, foretages følgende indkodning af negative tal: ε'er som givet i ligning (5.5) afbildes til værdier $M(\varepsilon)$, $0 \leq M(\varepsilon) \leq \alpha - 1$, ved afbildningen

$$M(\varepsilon) = \begin{cases} 2\varepsilon & \text{hvis } \varepsilon \geq 0, \\ 2|\varepsilon| - 1 & \text{hvis } \varepsilon < 0. \end{cases}$$

Afbildningen $M(\varepsilon)$ ordner fejl-signaler ved at sammenflette negative værdier med positive i følgen $0, -1, 1, -2, 2, \ldots$. Hvis værdierne ε følger en Laplace-fordeling centreret i nul vil fordelingen $M(\varepsilon)$ tilnærmelsesvis være den geometriske fordeling. Denne kan indkodes med en velvalgt Golomb-Rice kode (se kapitel 5 i INDFØRELSEN). (Weinberger et al.; 1996, side 5-6)

15 For et 'continuous-tone' billede gælder, at hver pixel benytter mindst en byte for hver af dets røde, grønne og blå værdier.

For en mere udtømmende beskrivelse af de ovenfor gennemgåede aspekter af LOCO-I se Weinberger et al. (1996).

En af grundene til at vi kort har valgt at behandle LOCO-I er, som vi skal se i næste afsnit, at den i løbet af MER-missionen har fået en langt mere fremtrædende rolle.

5.10 Valg af komprimering i MER

Generelt kan siges, at komprimeringsteknikker i rumanvendelser i ringe grad er underlagt standarder, CCSDS-standarderne er i højere grad væsentlige for kanalkodningen. CCSDS *har* en standard for satellitkommunikation, men når det gælder deep-space anvendelser, er der ikke angivet nogen standarder, hverken for komprimering med eller uden tab. Aaron Kiely, der er medlem af komprimeringsudvalget i CCSDS, kommenterede dette forhold således:

> For deep space our needs are a little bit specialized in that our data rates are typically much lower and so whereas for something that's in a lower orbit or something that can be collecting data and transmitting them in a much, much faster rate it becomes more critical that you have something that can operate really, really fast in hardware whereas for, you know, ICER on Mars we are not transmitting in such a high rate so having something that runs in software might be more reasonable. The considerations in those two applications may lead to a very different algorithm. (Kiely; 2005, side 43 i INTERVIEWS)

Valget af komprimeringsteknik er altså i ringe grad underlagt eksisterende standarder, men i højere grad præget af missionernes krav til billedkvalitet og kompleksitetsbegrænsninger for algoritmerne.

Målet med udviklingen af ICER har ifølge Kiely & Klimesh (2003) været at skabe en teknik med 'state of the art' komprimeringsresultater kombineret med en lav kompleksitet og gode muligheder for at inddæmme fejl. I Kiely & Klimesh (2003) hedder det således:

> Our software implementation of the coder in ICER has particularly low complexity, and so is well-suited for space applications and other applications where encoding speed can be of critical importance. (Kiely & Klimesh; 2003, side 20)

Desuden ønskede man en progressiv algoritme af grunde, som er diskuteret tidligere. Da MER-missionen blev planlagt, var den dominerende standard inden for billedkomprimering JPEG-standarden, som ikke levede op til kravene, og man var derfor nødt til at udvikle noget nyt. Udviklingen af ICER kom til at foregå sideløbende med udviklingen af JPEG-2000-standarden, der nu har afløst JPEG som dominerende komprimeringsteknik. JPEG-2000 er ligesom ICER en progressiv waveletbaseret teknik. Matthew Klimesh forklarer lighederne mellem de to således:

> Actually our development of ICER happened about the same time that the JPEG-2000 standards were being developed. There are a lot of similarities between ICER and JPEG-2000. Some of them are because we decided to make the same deci-

> sions they did and some of them because we thought 'I wonder what JPEG-2000 is doing on this' so we looked to see what they were doing. (Klimesh; 2005, side 45 i INTERVIEWS)

Da ICER er udviklet til deep-space anvendelser er der naturligvis også forskelle. En af de væsentligste forskelle er, at ICER anvender en sammenflettet entropikoder som 'motor', mens JPEG-2000 benytter sig af en beregningsmæssigt tungere aritmetisk entropikoder. Denne forskel skyldes hovedsageligt de krav til lav kompleksitet, der er stillet i MER-missionen. Kontekst-modelleringen i ICER er næsten den samme som i JPEG-2000, dog er der foretaget en række justeringer, som har sænket kompleksiteten[16]. Søren Forchhammer beskrev ICER således efter at have læst i artiklen Kiely & Klimesh (2003):

> De grundlæggende teknikker er de samme. Det er det, der foreligger op til og med JPEG-2000. Og noget af det de gør, er i virkeligheden at gå lidt længere tilbage, de skræller noget af det nyere i JPEG-2000 omkring de her kontekst-modeller og aritmetisk kodning af. Så det er relativt gennemprøvet. (Forchhammer; 2004, side 10 i INTERVIEWS)

5.10.1 Mars Odysseys rolle

En mulig strategi for billedtransmission er at komprimere billederne med ICER og derefter transmittere direkte fra roverne til Jorden. I stedet for at benytte denne strategi, kan man inddrage Mars Odyssey i transmissionen. Mars Odyssey har været i kredsløb om Mars siden 2001. Kredsløbsmissionen skulle egentlig have været afsluttet i 2004, men har fået forlænget fristen til september 2006. Den anden transmissionsstrategi går ud på at transmittere fra roverne til Mars Odyssey, som så kan videretransmittere data til Jorden. Fordelen ved denne relæ-kommunikation er, at roverne kun skal transmittere over nogle hundrede kilometer. Den korte afstand betyder, at data kan sendes mere eller mindre uden brug af kodning. Herved 'spares' en masse redundans, hvilket mindsker mængden af data, som skal transmitteres.

For kildekodningen har dette rent faktisk betydet, at man har kunnet benytte tabsfri komprimering med LOCO-I til transmissionen fra roverne. Da Mars Odyssey slet ikke har de samme begrænsninger i forhold til tid og energi som roverne, har man opnået en transmissionsvej, som er langt mere effektiv, og tabsfri komprimering har derfor været det bedste. Jacob Matijevic fortalte om denne alternative transmissionsstrategi:

> I think maybe the message from Mars is, that when you are in a situation where you can do this kind of relay-communication it's much less of a problem than when you are doing it over these long distances. [...] We intended to use the ICER algorithm ... but it turns out that the relay-communication work so well that most of the time we send back our data with lossless compression and very little compression-artifacts put into the system. (Matijevic; 2005, side 53)

16 Se eventuelt (Kiely & Klimesh; 2003, side 19)

Relæ-transmission har tidligere været benyttet blandt andet i Pathfinder-missionen, som benyttede Mars Global Surveyor som relæ, men effektiviteten af strategien ser ud til at have overrasket folkene på JPL.

5.11 Afrunding

Komprimering af billeder i MER har fortrinsvist foregået ved hjælp af algoritmen ICER, som er udviklet på JPL. Målet med udviklingen af ICER har været at opnå en progressiv algoritme med gode komprimeringsegenskaber. Da algoritmens virkefelt udelukkende skulle være deep-space anvendelser var lav kompleksitet og gode muligheder for fejlinddæmning ligeledes krav, der skulle opfyldes.

Udviklingen af ICER er dels en konsekvens af ovenstående krav og dels en konsekvens af, at JPEG-2000 endnu ikke var udviklet da MER-missionen blev planlagt. ICERs komprimeringsresultater modsvarer i øvrigt resultaterne for JPEG-2000 på trods af ICERs lavere kompleksitet.

Undervejs i missionen har man fundet ud af, at billedtransmissionen kan effektiviseres ved i stedet for at transmittere direkte til Jorden, at transmittere til Mars Odyssey. På den måde kan man benytte tabsfri komprimering gennem LOCO-I.

6 Kanalkodning i MER

> Space communication is unique in that, well for deep space communication there is a very high premium on the complexity of what's put on the spacecraft and no premium on the ground. (Hamkins; 2005, side 47 i Interviews)

I del II af Indførelsen har vi defineret og forklaret Reed-Solomon koder og foldningskoder. I dette kapitel beskrives, hvordan disse koder anvendes i MER-missionens kodningssystem.

Vi vil begynde kapitlet med at nævne de særlige omstændigheder, der gør sig gældende ved kommunikation mellem MER-roverne på Mars' overflade og Jorden, samt forklare hvilke problemer, der i den forbindelse skal løses.

Derefter vil vi give en overordnet beskrivelse af MER-missionens *kodningssystem* herunder en forklaring af de konkrete valg, der er foretaget. Følgeligt vil vi i større detalje beskrive kodningssystemets forskellige dele. Egentlig er der i MER-missionen ikke tale om et enkelt fast kodningssystem, men om flere mulige systemer, der kan skiftes imellem. Det system, vi har valgt at beskrive, er den mulighed, som oftest benyttes. Afslutningsvis vil vi diskutere valget af koderne i MERs kodningssystem. Kapitel er baseret på forskellige kilder, som vil blive nævnt, når de anvendes.

6.1 Særlige omstændigheder for roverne

Til al transmission af data kræves energi. Jo mere energi man har til rådighed, desto stærkere et signal kan man transmittere. Et stærkt signal er mindre sårbart overfor forstyrrelser end et svagt signal, og man er derfor interesseret i at få et signal, der er så stærkt som muligt.

For MER-rovernes vedkommende er mængden af energi, der er til rådighed til kommunikation, yderst begrænset. Al den energi, en rover har til rådighed, stammer fra rovernes solceller. Med denne energi skal roveren kunne bevæge sig rundt på Mars' overflade, tage billeder, foretage og analysere prøver samt transmittere data. Desuden skal roveren kunne 'holde varmen' om natten. Dette kræver også energi. Energi til transmission er altså stærkt begrænset, hvilket betyder, at signalet, som roveren udsender, er yderst sårbart blandt andet overfor stråling i rummet. Jon Hamkins fra JPL fortalte:

> We have very limited power, something less than a Christmas tree light bulb to transmit from there to here so we have humongous, seventy-meter deep space antennas trying to piece together the signal that is coming down. (Hamkins; 2005, side 47 i Interviews)

Et andet forhold, som gør transmissionen sårbar er, at beskeder fra roveren komprimeres inden de sendes (jævnfør kapitel 5). Dekomprimeringsalgoritmer er ofte meget følsomme, og en enkelt fejl kan derfor gøre stor skade. Af samme årsag udgør kodningsteorien en af grundpillerne i rumindustriens kommunikationssystemer, hvor man altså ikke bare kan 'skrue op for energien'.

6.2 Kodningssystemet overordnet

For at opnå en tilstrækkelig lav sandsynlighed for afkodningsfejl, benyttes i MER-missionen et sammensat system bestående af en *ydre* Reed-Solomon kode og en *indre* foldningskode. Betegnelserne indre og ydre skal forstås således:

Indkodning

- Data indkodes først med Reed-Solomon koden. De kodeord, der kommer fra denne indkodning, inddeles herefter af en *blok-opdeler* i datarammer, som indkodes af foldningskoden. Denne indkodning producerer en række koderammer, som transmitteres. 'Indre' skal altså forstås som tættest på kanalen.

Afkodning

- De transmitterede koderammer afkodes til datarammer ved foldningskodens afkoder. Datarammerne sammensættes herefter i en *blok-sammensætter* til kodeord for Reed-Solomon koden og benyttes som inddata til dennes afkoder. Når datarammerne er afkodet er transmissionen tilendebragt.

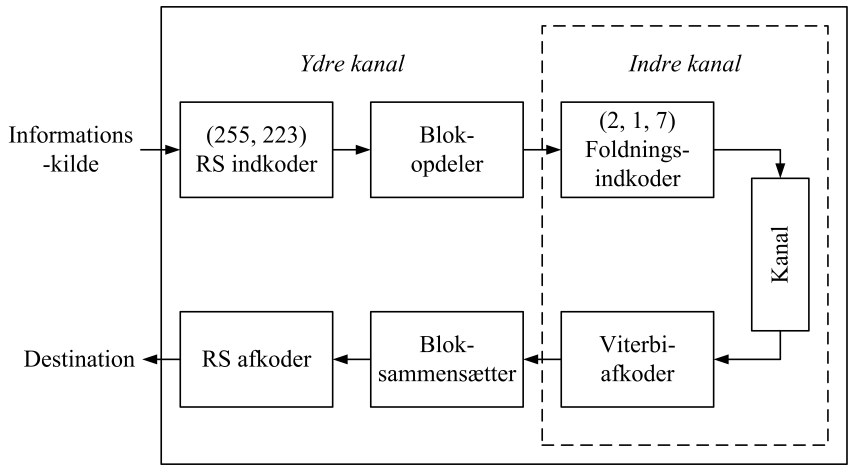
Ved benyttelse af den nævnte kodningsstrategi opnår man i praksis et system, hvor den indre indkodning og afkodning sammen med den binære kanal kan opfattes som en *indre* kanal. Et system, der gør brug af en sådan strategi, kaldes et *konkateneret* system. Systemet er skitseret på figur 6.1.

Afkodning af foldningskoden foregår ved hjælp af en algoritme kaldet *Viterbi*-algoritmen. Denne algoritme har en tendens til at én afkodningsfejl følges af flere, det vil sige en tendens til at producere bursts under afkodningen. I det konkatenerede system skal uddata fra Viterbi-afkoderen imidlertid afkodes endnu en gang gennem Reed-Solomon koden, som netop har korrigering af bursts som sit store fortrin. De to ind- og afkodninger komplementerer derfor hinanden.

Med det konkatenerede system opnår man i praksis en situation, i hvilken man transmitterer en besked med en Reed-Solomon kode over en kanal (her den indre kanal), der har en tendens til at producere burstfejl.

For at undgå at et meget langt burst ødelægger Reed-Solomon kodens muligheder for fejlkorrigering benyttes endnu en procedure ved indkodning kaldet *sammenfletning*[1]. Proceduren, som beskrives senere i kapitlet, bevirker, at et langt burst spredes ud i flere mindre bursts, som Reed-Solomon koden kan håndtere. Man opnår så at sige et 'burst af bursts'.

1 Denne procedure er forskellig fra den, der blev beskrevet i sidste kapitel.



Figur 6.1 CCSDS-konkateneringsstandard med en indre $(2, 1, 7)$-foldningskode og en ydre $(255, 223)$-Reed-Solomon kode. (Costello, Jr. et al.; 1998, side 2536)

6.2.1 CCSDS-standarden

MER-missionen benytter et kodningssystem, som anbefales af *Consultative Committee for Space Data Systems* i (CCSDS; 2002). Systemet er et konkateneret system bestående af en ydre Reed-Solomon kode med parametre $(255, 223)$ og sammenfletning til dybde $1, 2, 3, 4, 5$ eller 8 (forklares i afsnit 6.3.4), samt en indre foldningskode med parametre $(2, 1, 7)$ og maximum-likelihood Viterbi-afkodning. Afkodning af Reed-Solomon koden er ikke nævnt i standarden.

I det følgende vil vi forsøge at matematisere det beskrevne system. Vi har valgt at beholde opdelingen med et indre og et ydre system, da det giver en mere overskuelig beskrivelse af koderne.

6.3 Det ydre system

I beskrivelsen af det ydre system opfatter vi det indre system som én kanal, der producerer burstfejl. Vi begynder med den ydre kode.

6.3.1 $(255, 223)$-Reed-Solomon koden

Den anbefalede Reed-Solomon kode er defineret over legemet $\mathbb{F}_{256}$ konstrueret på baggrund af det primiske binære polynomium

$$F(x) = x^8 + x^7 + x^2 + x + 1.$$

Kodens generatorpolynomium er defineret over $\mathbb{F}_{256}$ og givet ved

$$g(x) = \prod_{j=112}^{143} (x - \alpha^{11j})$$

hvor α er rod i F. Den beskrevne kode har parametrene:

$$k = 223, \qquad n = 255, \qquad d_{\min} = 33.$$

Disse parametre giver en rate på $^{223}/_{255} \sim 0,87$ og minimumsafstanden er 33, så der kan korrigeres op til 16 fejl i et kodeord.

Det specifikke valg af α og faktoren 11 i potensen for α skyldes overvejelser angående optimering af hardware-implementering af indkoderen, overvejelser, som vi ikke vil komme ind på denne rapport. Den interesserede læser kan finde en forklaring i (JPL; 1982).

I implementeringen af Reed-Solomon koden repræsenteres hvert element i $\mathbb{F}_{256}$ ved en byte. Et kodeord bestående af 256 symboler repræsenteres altså af 256 byte svarende til 2.040 bit. Umiddelbart betyder dette at et burst af 121 eller færre korrumperede bit ifølge sætning 8.10 i INDFØRELSEN højest vil ødelægge $1 + \lceil (121-1)/8 \rceil = 16$ symboler i det tilsvarende kodeord. Ifølge diskussionen i samme kapitel kan denne fejl korrigeres ved afkodning.

6.3.2 Systematisk indkodning

Indkodning af dataord foregår ved *systematisk indkodning*. For systematisk indkodning udgøres de første k symboler i kodeordet af dataordets symboler, mens de sidste $n-k$ udgøres af paritetssymbolerne.

Ved indkodning knyttes til hvert datapolynomium $a(x)$ et kodepolynomium $c(x)$ givet ved

$$c(x) = x^{32}a(x) + r(x),$$

hvor polynomiet $r(x)$, som indeholder alle paritetssymboler, er givet ved

$$r(x) = R_{g(x)}[x^{32}a(x)].$$

I ovenstående betegner $R_{g(x)}$ reduktion modulo $g(x)$. At $c(x)$ er et kodeord indses let:

$$\begin{aligned} R_{g(x)}[c(x)] &= R_{g(x)}\left[x^{32}a(x) + r(x)\right] \\ &= R_{g(x)}\left[R_{g(x)}[x^{32}a(x)] + R_{g(x)}[r(x)]\right] \\ &= R_{g(x)}\left[r(x) + r(x)\right] \\ &= 0. \end{aligned}$$

Dette betyder, at $g(x)|c(x)$, så $c(x)$ er et kodepolynomium. I ovenstående har vi udnyttet, at $R_{g(x)}[b(x) + d(x)] = R_{g(x)}[b(x)] + R_{g(x)}[d(x)]$, hvilket kan vises ved en simpel udregning.

Ved den angivne indkodning opnås et kodeord på formen

$$c = a_1 a_2 \ldots a_k r_1 r_2 \ldots r_{32},$$

hvoraf tydeligt ses, at kodeordet er på systematisk form.

6.3.3 Afkodning

Reed-Solomon koder udmærker sig ved at kunne afkodes ved få beregninger. I dette afsnit vil vi angive en afkodningsalgoritme. Afkodning er ikke en del af CCSDS-standarden, så man kan afkode som man lyster. Jon Hamkins fra JPL foreslog i en elektronisk brevveksling, at vi beskrev en standardalgoritme, for eksempel Berlekamp-Masseys algoritme eller Euklids algoritme. Begge algoritmer er overkommelige at forstå og samtidigt effektive. Vi har valgt en algoritme kaldet *Sugiyamas* algoritme, som

bygger på Euklids udvidede algoritme for polynomier kendt fra grundlæggende algebra.

Selvom algoritmen er nem at opstille og gennemprøve, kræver det en del forarbejde at forklare i detaljer, hvorfor den virker. Vi har derfor valgt at indlede beskrivelsen med en kort forklaring af det helt grundlæggende princip. Derefter springer vi frem til en præsentation af algoritmen og giver tilsidst et eksempel på en afkodning.

Udgangspunktet for fremstillingen er en Reed-Solomon kode over $\mathbb{F}_q$ med ω som primitivt element. Generatorpolynomiet g for koden er givet ved

$$g(x) = \prod_{j=b}^{b+2t} (x - \omega^j),$$

så minimumsafstanden er $2t + 1$.

For transmission af et kodeord $c(x)$ vil vi beskrive n fejl i transmissionen ved et polynomium $e(x)$ med n komponenter forskellige fra nul. Det modtagne ord $v(x)$ er da givet ved

$$v(x) = c(x) + e(x).$$

Eksempel 6.1
Betragt koden fra eksempel 8.9 i INDFØRELSEN. Lad 0012276 være det transmitterede kodeord, $c(x) = x^4+2^3+2x^2+7x+6$, og lad det modtagne ord være 0000276, $v(x) = 2x^2 + 7x + 6$. Så er fejlen e givet ved

$$e(x) = v(x) - c(x) = x^4 + 2x^3.$$

◇

Lad $v(x)$ være det modtagne ord og lad $c(x)$ være det tilhørende kodeord. Lad $j < 2t$. Da $c(x)$ er et kodeord gælder, at $c(\omega^{b+j-1}) = 0$, så

$$v(\omega^{b+j-1}) = c(\omega^{b+j-1}) + e(\omega^{b+j-1}) = e(\omega^{b+j-1}).$$

Definition 6.2
Det j'te frekvensdomæne-syndrom S_j*, eller bare* syndrom*, defineres ved*

$$S_j = \hat{v}_{b+j-1}.$$

Da

$$\hat{v}_{b+j-1} = v(\omega^{b+j-1}) = c(\omega^{b+j-1}) + e(\omega^{b+j-1}) = e(\omega^{b+j-1}) = \hat{e}_{b+j-1}$$

gælder altså

$$S_j = \hat{e}_{b+j-1}.$$

For at forsimple den følgende diskussion vil vi gøre den antagelse, at $b = 1$. Antagelsen ødelægger ikke diskussionen, som med få modifikationer kan overføres til det generelle tilfælde (Blahut; 2003, side 160).

For $b = 1$, er $S_j = e(\omega^j)$. Antag, at e indeholder $\nu \leq t$ komponenter, $i_1, \ldots i_\nu$, forskellige fra nul. Så er

$$S_1 = e(\omega) = e_{i_1}\omega^{i_1} + \ldots + e_{i_\nu}\omega^{i_\nu}$$

og

$$\begin{aligned} S_j &= e(\omega^j) = e_{i_1}(\omega^j)^{i_1} + \ldots + e_{i_\nu}(\omega^j)^{i_\nu} \\ &= e_{i_1}(\omega^{i_1})^j + \ldots + e_{i_\nu}(\omega^{i_\nu})^j. \end{aligned}$$

Lad $Y_\ell = e_{i_\ell}$ og $X_\ell = \omega^{i_\ell}$. Da kan S_j'erne opskrives som

$$\begin{aligned} S_1 &= Y_1X_1 + Y_2X_2 + \ldots + Y_\nu X_\nu \\ S_2 &= Y_1X_1^2 + Y_2X_2^2 + \ldots + Y_\nu X_\nu^2 \\ &\vdots \qquad \vdots \\ S_{2t} &= Y_1X_1^{2t} + Y_2X_2^{2t} + \ldots + Y_\nu X_\nu^{2t}. \end{aligned}$$

Hvis vi kender X_ℓ, så kan vi bestemme i_ℓ, og hvis vi kender Y_ℓ, kan vi bestemme e_{i_ℓ} og dermed $c(x)$, det vil sige afkode. Udfordringen er altså nu at løse ovenstående ligninger med henblik på at finde X_ℓ'erne og Y_ℓ'erne.

At løse ovenstående ligninger med få udregninger er ikke ligetil, men kan faktisk lade sig gøre. Vi vil springe en stor del af udledningen af løsningen over og gå direkte til resultatet. Lad

$$\begin{aligned} S(x) &= \sum_{j=1}^{2t} S_j x^j, \\ \Lambda(x) &= \prod_{\ell=1}^{\nu}(1 - xX_\ell) \quad \text{og} \\ \Gamma(x) &= \sum_{i=1}^{\nu} xY_iX_i \prod_{j\neq 1}(1 - X_jx). \end{aligned}$$

Hvis S, Λ og Γ opfylder

$$\Gamma(x) \equiv \Lambda(x)S(x) \mod x^{2t+1}, \tag{6.1}$$

samt at $\text{grad}(\Gamma) \leq t$ og $\text{grad}(\Lambda) \leq t$, så opfylder X_ℓ'erne og Y_ℓ'erne det førnævnte ligningssystem. Endvidere er løsningen entydig. Dette udledes og diskuteres i (Blahut; 2003, kapitel 6 og 7). Inden vi præsenterer algoritmen, som finder Λ og Γ giver vi endnu et resultat som ikke bevises. For et bevis se (Blahut; 2003, side 196-197).

Sætning 6.3 (Forney)
Y_ℓ er givet ved

$$Y_\ell = -X_\ell \frac{\Gamma(X_\ell^{-1})}{\Lambda'(X_\ell^{-1})}.$$

Sætningen, som egentlig kaldes *Forneys algoritme*, gør os i stand til at bestemme Y_ℓ'erne når vi kender Λ og Γ. Polynomiet Λ' er den formelle afledte af Λ.

Sugiyamas afkodningsalgoritme bygger på Euklids udvidede algoritme. Algoritmen finder for to polynomier $s(x)$ og $t(x)$ den største fælles divisor samt polynomier $b(x)$ og $d(x)$, der opfylder

$$\gcd(s(x), t(x)) = b(x)s(x) + d(x)t(x).$$

Algoritmen er velkendt og vi vil derfor blot benytte den som et resultat. For en mere grundig analyse se eksempelvis (Blahut; 2003, side 217-221).

Algoritme 6.4 (Udvidet Euklid)
Inddata*: Polynomier $s(x)$ og $t(x)$.*

1. *Sæt $s^{(0)}(x) = s(x)$, $t^{(0)}(x) = t(x)$ og $A^{(0)} = I$.*
2. *Bestem den r'te kvotient $Q^{(r)}$ af s og t som det polynomium, der opfylder*
$$s^{(r)}(x) = Q^{(r)}(x)t^{(r)}(x) + b(x),$$
med $\text{grad}(b) < \text{grad}(t)$.
3. *Sæt*
$$A^{(r+1)}(x) = \begin{pmatrix} 0 & 1 \\ 1 & -Q^{(r)}(x) \end{pmatrix} A^{(r)}(x).$$
4. *Sæt til sidst*
$$\begin{pmatrix} s^{(r+1)}(x) \\ t^{(r+1)}(x) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -Q^{(r)}(x) \end{pmatrix} \begin{pmatrix} s^{(r)}(x) \\ t^{(r)}(x) \end{pmatrix} = A^{(r+1)}(x) \begin{pmatrix} s(x) \\ t(x) \end{pmatrix}.$$

Trinene 2-4 gentages indtil det skridt R, hvor $t^{(R)} = 0$. Så er $\gcd(s(x), t(x))$ *givet ved $s^{(R)}(x)$.*
Uddata*: $\gcd(s(x), t(x))$ samt polynomier $b(x)$ og $d(x)$, der opfylder*

$$\gcd(s(x), t(x)) = b(x)s(x) + d(x)t(x).$$

Euklids algoritme opfylder, at $\text{grad}(t^{(r+1)}) < \text{grad}(t^{(r)})$, så algoritmen terminerer på et tidspunkt (Blahut; 2003, side 219).

I afkodningsalgoritmen, som vi definerer om lidt, skal Euklids algoritme ikke føres til ende. For det r'te trin i algoritmen gælder, at

$$\begin{pmatrix} s^{(r)}(x) \\ t^{(r)}(x) \end{pmatrix} = \begin{pmatrix} A_{11}^{(r)}(x) & A_{12}^{(r)}(x) \\ A_{21}^{(r)}(x) & A_{22}^{(r)}(x) \end{pmatrix} \begin{pmatrix} s(x) \\ t(x) \end{pmatrix}. \tag{6.2}$$

Heraf følger, at

$$t^{(r)}(x) = A_{21}^{(r)}(x)s(x) + A_{22}^{(r)}(x)t(x),$$

og dermed

$$t^{(r)}(x) \equiv t(x)A_{22}^{(r)} \mod s(x). \tag{6.3}$$

Hvis vi sætter $s(x) = x^{2t+1}$ og $t(x) = S(x)$ (syndrom-polynomiet) kan vi benytte Euklids udvidede algoritme til at bestemme Γ og Λ.

Algoritme 6.5 (Sugiyama)
Inddata*: Syndrompolynomium $S(x)$.*

1. *Sæt $t^{(0)}(x) = S(x)$, $s(x) = x^{2t+1}$ og lad $A^{(0)}$ være 2×2-enhedsmatricen.*
2. *Bestem $Q^{(r)}, A^{(r)}(x)$ og $\begin{pmatrix} s^{(r)}(x) \\ t^{(r)}(x) \end{pmatrix}$, som i Euklids algoritme.*
3. *Stop algoritmen, når* $\text{grad}(t^{(r)}(x)) \leq t$.

Uddata*: Polynomier* Γ *og* Λ*, som opfylder*

$$\Gamma(x) \equiv S(x)\Lambda(x) \mod x^{2t+1},$$

samt at $\text{grad}(\Gamma), \text{grad}(\Lambda) \leq t$.

Sætning 6.6
Lad R *være det skridt algoritmen stopper ved. Sæt*

$$\begin{aligned} \Gamma(x) &= t^{(R)}(x), \\ \Lambda(x) &= A_{22}^{(R)}. \end{aligned}$$

Da er Γ, Λ *en løsning til*

$$\Gamma(x) \equiv S(x)\Lambda(x) \mod x^{2t+1},$$

som opfylder $\text{grad}(\Gamma), \text{grad}(\Lambda) \leq t$.

Bevis
Lad R være trinnet algoritmen stopper ved. Så er

$$\text{grad}(\Gamma) = \text{grad}(t^{(R)}(x)) \leq t$$

ifølge algoritmebeskrivelsen. Desuden er $\Gamma(x) \equiv S(x)A_{22}^{(R)}(x) \mod x^{2t+1}$ ifølge ligning (6.3). Vi mangler at vise, at $\text{grad}(A_{22}^{(x)}) \leq t$. For at vise dette, er vi nødt til at se på matricen $A^{(r)}$. Matricen kan skrives

$$A^{(r)}(x) = \prod_{j=0}^{r-1} \begin{pmatrix} 0 & 1 \\ 1 & -Q^{(j)}(x) \end{pmatrix}. \tag{6.4}$$

Heraf følger ved udregning, at

$$\begin{aligned} \text{grad}(A_{12}^{(r)}) &= \sum_{j=0}^{r-1} \text{grad}(Q^{(j)}) \text{ og} \\ \text{grad}(A_{22}^{(r)}) &= \sum_{j=0}^{r} \text{grad}(Q^{(j)}) \geq \text{grad}(A_{12}^{(r)}). \end{aligned}$$

Da $s^{(r)}(x) = t^{(r-1)}(x)$ ifølge trin 4 i algoritme 6.4 er

$$\text{grad}(s^{(r)}) = \text{grad}(t^{(r-1)}) > (t^{(r)}).$$

Denne observation vender vi tilbage til om lidt. Fra ligning (6.4) ses, at $\det(A^{(r)}(x)) = (-1)^r$, så $A^{(r)}(x)$ er invertibel. Multipliceres ligning (6.2) med den inverse $\left(A^{(r)}(x)\right)^{-1}$ fås

$$\begin{pmatrix} s(x) \\ t(x) \end{pmatrix} = (-1)^r \begin{pmatrix} A_{22}^{(r)}(x) & -A_{12}^{(r)}(x) \\ -A_{21}^{(r)}(x) & A_{11}^{(r)}(x) \end{pmatrix} \begin{pmatrix} s^{(r)}(x) \\ t^{(r)}(x) \end{pmatrix}.$$

Der gælder altså

$$s(x) = (-1)^r (A_{22}^{(r)}(x)s^{(r)}(x) - A_{12}^{(r)}(x)t^{(r)}(x)).$$

Kombineret med den foregående observation har vi da, at

$$\text{grad}(s(x)) = \text{grad}(A_{22}^{(r)}(x)) + \text{grad}(s^{(r)}(x)).$$

Da $s^{(r)}(x) = t^{(r-1)}(x)$ og $\text{grad}(t^{(r-1)}(x)) > t$ (ellers var algoritmen jo stoppet) er

$$\begin{aligned} \text{grad}(A_{22}^{(R)}) &= \text{grad}(s(x)) - \text{grad}(t^{(R-1)}(x)) \\ &\leq 2t + 1 - (t-1) = t, \end{aligned}$$

hvilket var det, der skulle vises. Algoritmens uddata opfylder altså de betingelser, som blev opstillet. □

At algoritmen giver en afkodning illustrerer vi med et eksempel.

Eksempel 6.7
Lad $\mathcal{C}$ være Reed-Solomon koden givet i eksempel 8.9 i INDFØRELSEN med $d_{\min} = 2t + 1 = 5$. For denne kode er b lig 0. Lad det transmitterede kodeord være 0012276 svarende til polynomiet $c(x) = x^4+2x^3+2x^2+7x+6$ og lad det modtagne ord være givet ved $v(x) = x^2 + 7x + 6$. Der er altså fejl på to positioner, og det modtagne ord skulle da kunne afkodes korrekt. Syndromerne S_j for $j = 1, 2, 3, 4$ bliver

$$\begin{aligned} S_1 &= v(2^0) = 3, \\ S_2 &= v(2^1) = 0, \\ S_3 &= v(2^2) = 3, \\ S_4 &= v(2^3) = 6, \end{aligned}$$

og syndrompolynomiet er så givet ved

$$S(x) = 6x^4 + 3x^3 + 3x.$$

Sugiyamas algoritme giver i første trin

$$\begin{aligned} Q^{(0)}(x) &= 2x + 1. \\ A^{(1)}(x) &= \begin{pmatrix} 0 & 1 \\ 1 & 2x+1 \end{pmatrix}. \\ \begin{pmatrix} s^{(1)}(x) \\ t^{(1)}(x) \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 1 & 2x+1 \end{pmatrix} \begin{pmatrix} x^5 \\ 6x^4+3x^3+3x \end{pmatrix} \\ &= \begin{pmatrix} 6x^4+3x^3+3x \\ 3x^3+6x^2+3x \end{pmatrix}. \end{aligned}$$

I andet trin fås

$$\begin{aligned} Q^{(1)}(x) &= 2x + 5. \\ A^{(2)}(x) &= \begin{pmatrix} 0 & 1 \\ 1 & 2x+1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 2x+5 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 2x+1 \\ 2x+5 & 4x^2+5x+4 \end{pmatrix}. \\ \begin{pmatrix} s^{(2)}(x) \\ t^{(2)}(x) \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 1 & 2x+5 \end{pmatrix} \begin{pmatrix} 6x^4+3x^3+3x \\ 3x^3+6x^2+3x \end{pmatrix} \\ &= \begin{pmatrix} 3x^3+6x^2+3x \\ 2x^2+x \end{pmatrix}. \end{aligned}$$

Herefter terminerer algoritmen, da $\text{grad}(t^{(2)}) \leq 2$. Polynomierne Γ og Λ er da givet ved

$$\begin{aligned} \Gamma(x) &= 2x^2 + x, \\ \Lambda(x) &= 4x^2 + 5x + 4. \end{aligned}$$

Polynomiet Λ kan bringes på den ønskede form ved at dividere polynomiet med 4. Herved opnår vi løsningen Λ^*, som kan skrives[2]

$$\begin{aligned} \Lambda^*(x) &= x + 2x + 1 \\ &= (x+7)(x+5) \\ &= (5x+1)(7x+1) \\ &= (1-5x)(1-7x), \end{aligned}$$

hvoraf følger, at $X_1 = 5^{(-1)} = 7$ og $X_2 = 7^{(-1)} = 5$. Så er $X_1 = 2^4 = 2^{i_1}$ og $X_2 = 2^3 = 2^{i_2}$. Der er altså fejl på pladserne $i_1 = 4$ og $i_2 = 3$. Fejlenes størrelse beregnes ved hjælp af Forneys algoritme (sætning 6.3) til

$$Y_1 = 5 \cdot \frac{\Gamma(7)}{\Lambda'(7)} = 5 \cdot \frac{6}{4} = 1,$$

$$Y_2 = 7 \cdot \frac{\Gamma(5)}{\Lambda'(5)} = 7 \cdot \frac{4}{3} = 2.$$

I ovenstående er $\Lambda'(x) = 5x+5$. Fejlen opstået ved transmissionen er altså givet ved

$$e(x) = Y_1 x^{i_1} + Y_2 x^{i_2} = 1x^4 + 2x^3 = 2x^4 + x^3.$$

Det afkodede ord bliver således

$$c(x) = v(x) + e(x) = x^4 + 2x^3 + 2x^2 + 7x + 6,$$

som ses at være identisk med det transmitterede ord. Eksemplet viser, at afkodningen er beregningsmæssigt billig. ◇

Den afkodning vi her har beskrevet, fungerer kun korrekt så længe antallet af fejl er mindre end det antal fejl t, som koden garanteret kan korrigere. I praksis anvender man ofte algoritmer, som afkoder på trods af, at afstanden fra det modtagne ord til nærmeste kodeord er større end t, det vil sige, selvom flere end t symbolfejl er opstået. Vi har for simplicitetens skyld valgt ikke at beskrive en sådan algoritme. Det skal dog nævnes, at den før omtalte Berlekamp-Massey-algoritme kan udvides, så den foretager en sådan afkodning (Blahut; 2003, side 213).

6.3.4 Sammenfletning

Sammenfletning (*interleaving*) af kodeord benyttes som tidligere nævnt for at undgå, at lange burstfejl ødelægger en transmission.

Lad der været givet m dataord $a_1, a_2 \ldots, a_m$ hver af længde k svarende til km datasymboler. Hvert dataord a_i indkodes og giver et tilhørende

2 Faktoriseringen kan foretages ved en *Chien-Søgning*. (Blahut; 2003, side 164)

kodeord $c_i = c_i^1 c_i^2 \dots c_i^n$. Disse m kodeord flettes nu sammen til kodestrømmen

$$c_1^1 c_2^1 \dots c_m^1 c_1^2 c_2^2 \dots c_m^2 \dots c_1^n c_2^n \dots c_m^n,$$

som består af nm symboler. Det ses, at de første m symboler i kodestrømmen svarer til det første symbol i hvert af kodeordene, de næste m symboler svarer til det andet symbol i kodeordene og så videre. Den beskrevne indkodning kaldes *sammenfletning* til dybde m.

Eksempel 6.8
Sammenfletning vil vi illustrere ved hjælp af koden fra eksempel 8.9 i INDFØRELSEN. Lad c_1, c_2, c_3 være de tre sidste kodeord fra denne kode, nemlig

$$\begin{aligned} c_1 &= 7456044, \\ c_2 &= 7460603, \\ c_3 &= 7472475. \end{aligned}$$

Ved sammenfletning til dybde 3 giver disse kodeord tilsammen kodeordet

$$7774445676020644074 35.$$

Det ses, at rækkefølgen af symboler i ovenstående svarer til rækkefølgen man opnår, hvis man opfatter de tre kodeord som en matrix, og læser én søjle ad gangen. Opskrevet binært er kodestrømmen:

$$111111111100100100101110111110000010000110100100000111100011101.$$

Antag nu, at der ved transmission forekommer et burst, som ødelægger bittene 27-42, altså 16 bit. Den modtagne strøm kunne da være

$$111111111100100100101110110101110100111101100100000111100011101,$$

hvor understregningen angiver korrumperede bit. For modtagne ord over $\mathbb{F}_8$ svarer dette til

$$\begin{aligned} c_1 &= 7455\underline{7}44, \\ c_2 &= 7466\underline{5}03, \\ c_3 &= 74\underline{6}4475. \end{aligned}$$

Da Reed-Solomon koden ifølge eksempel 6.7 kan korrigere to fejl resulterer ovenstående i en korrekt afkodning. Uden sammenfletning vil et burst af længde 16 resultere i $1 + \lceil (16-1)/3 \rceil = 6$ ødelagte symboler, hvilket med sikkerhed vil give en afkodningsfejl. ◇

Eksemplet fører os frem til følgende sætning:

Sætning 6.9
For Reed-Solomon koden med parametre $(255, 223)$ *gælder, at koden kan korrigere bursts af længde mindre end eller lig*

$$1 + 8(16m - 1) = 128m - 7,$$

når der anvendes sammenfletning til dybde m.

Bevis
Ligesom i beviset for sætning 8.10 i INDFØRELSEN tager vi udgangspunkt i det værst tænkelige burst. Dette burst ødelægger den sidste bit i et symbol i kodeordet c_i og de efterfølgende $8(16m-1)$ bit. Den sidste del af burst'et ødelægger $16m-1$ symboler begyndende i kodeordet c_{i+1}. Dette forårsager fejl i 16 symboler i hvert af ordene $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_m$ samt 15 fejl i c_i, som har 1 fejl i forvejen. Burst'et giver altså alle kodeord 16 symbolfejl, hvilket ikke resulterer i en afkodningsfejl. □

Bemærk, at for $m = 1$ svarende til ingen sammenfletning giver sætningen, at der kan korrigeres et burst på op til 121 bit. Dette er i overensstemmelse med det tidligere beskrevne. Ved sammenfletning til dybde 8 – den største dybde i CCSDS-anbefalingen – kan bursts af længde op til 1.017 korrigeres.

6.4 Det indre system

Det indre system består af den CCSDS-standardiserede $(2, 1, 7)$-foldningskode samt deep-space kanalen. I modsætning til det ydre system, der er designet til at korrigere burstfejl opstået ved den indre afkodning af foldningskoden, sørger det indre system for at korrigere fejl på enkelte bit opstået som følge af stråling i rummet. Foldningskoderne anvendes, da de er særdeles velegnede til at korrigere for enkelte spredte fejl i den modtagne kodefølge. Det er netop denne type fejl, som støj på deep-space kanalen oftest giver anledning til.

6.4.1 $(2, 1, 7)$-foldningskoden

Indkodningen af foldningskoden kan foregå ved hjælp af dennes generatormatrix som beskrevet i kapitel 9 i INDFØRELSEN. De specifikke generatorpolynomier for CCSDS-$(2, 1, 7)$-foldningskoden er givet ved:

$$\begin{aligned} g_0 &= 1 + D + D^3 + D^4 + D^6, \\ g_1 &= 1 + D^3 + D^4 + D^5 + D^6. \end{aligned}$$

Generatormatricen for koden er da givet ved:

$$\begin{aligned} G(D) &= \begin{pmatrix} 1 + D + D^3 + D^4 + D^6 & 1 + D^3 + D^4 + D^5 + D^6 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 + \dfrac{D + D^5}{1 + D + D^3 + D^4 + D^6} \end{pmatrix}. \end{aligned}$$

Kodefølgen genereres ved at multiplicere datafølgen med generatormatricen. Koden har en fri Hammingafstand på $d_{\text{fri}} = 10$. (CCSDS; 2002, side 2.2), (Costello, Jr. et al.; 1998, side 2535)

6.4.2 Viterbi-afkodning

I MER-missionen anvendes en afkodningsprocedure kaldet *maximum-likelihood Viterbi-afkodning* til foldningskoden. Afkodningsproceduren tager udgangspunkt i et sandsynlighedsmål i stedet for Hammingafstanden eller den euklidiske afstand imellem to koderammer[3]. Viterbi-algoritmen, som

3 For en beskrivelse af hvordan Hammingafstand kan oversættes til euklidisk afstand se (Blahut; 2003, side 336-337).

er grundlaget for foldningskodens afkodning, er en procedure, der udforsker trellis-strukturen af en foldningskode. Algoritmen bygger på teknikker hentet fra dynamisk programmering.

I og med at en foldningskodefølge i princippet er uendelig lang, må afkoderen, som har endelig kompleksitet, begynde med at foretage valg på baggrund af et endeligt udsnit af den modtagne følge.

Betragt for en (n, k, m)-foldningskode en følge af k-dimensionale datarammer givet ved

$$a = a_0, a_1, \ldots, a_{L-1},$$

som i alt indeholder kL datasymboler. Datafølgen a indkodes til en følge af n-dimensionale koderammer

$$c = c_0, c_1, \ldots, c_{L+m-1},$$

som i alt indeholder $N = n(L + m)$ kodesymboler. Kodefølgen c udsættes for modulation og transmitteres over en kanal. Efter demodulation ved destinationen haves den modtagne følge

$$v = v_0, v_1, \ldots, v_{L+m-1}.$$

En afkoder producerer et estimat $\tilde{c}$ af kodefølgen c baseret på den modtagne følge v. En maksimum-likelihood afkoder vælger $\tilde{c}$ som den følge c, der maksimerer log-likelihood funktionen $P(v|c)$. For den betragtede kanal har vi følgende:

$$P(v|c) = \prod_{i=0}^{L+m-1} P(v_i|c_i), \tag{6.5}$$

hvor $P(v_i|c_i)$ er en kanal-overgangssandsynlighed, altså en specifik symbolsandsynlighed for en specifik kanal. Når sandsynligheden af alle kodefølger er ligeligt fordelt, giver maksimum-likelihood afkodning en minimal fejlsandsynlighed. Af ligning 6.5 følger, at

$$\log P(v|c) = \sum_{i=0}^{L+m-1} \log P(v_i|c_i). \tag{6.6}$$

log-likelihood funktionen $\log P(v|c)$ kaldes for *metrikken* associeret med stien c og angives $M(v|c)$. Leddene $\log P(v_i|c_i)$ i ligning (6.6) kaldes for *gren-metrikker* og angives $M(v_i|c_i)$. Sti-metrikkerne $M(v|c)$ kan derfor opskrives som

$$M(v|c) = \sum_{i=0}^{L+m-1} M(v_i|c_i).$$

En *partiel* metrik bestående af de første j grene af en sti kan da defineres ved

$$M([v|c]_j) = \sum_{i=0}^{j-1} M(v_i|c_i).$$

Viterbi-algoritmen for v finder stien gennem trellis-diagrammet med den største metrik, det vil sige maximum-likelihood-stien. Algoritmen behandler v på iterativ manér. I hvert skridt sammenligner algoritmen metrikkerne af samtlige indgående stier til hver tilstand og gemmer stien

med den største metrik, kaldet *overleveren* $\tilde{c}$, sammen med dennes metrik. Den formulering af Viterbi-algoritmen (algoritme 6.10), som gives nedenfor, stammer fra (Lin & Costello, Jr.; 1983). og inddrager foldningskodens hukommelse m.

Viterbi-algoritmen tager generelt udgangspunkt i de tilstande, S_ι, der indgår i trellis-beskrivelsen af en foldningskode (se kapitel 9 i INDFØRELSEN). For en binær foldningskode med tvungen længde ν og symboler fra $\mathbb{F}_q$ findes der q^ν forskellige tilstande, det vil sige $0 \leq \iota \leq q^\nu - 1$. Til hver tilstand S_ι er der q^k indgående stier, hvor k er længden af datarammen. Da foldningskoden, der anvendes i MER, er binær, haves $q^\nu = 2^\nu$, og da $k = 1$ er antallet af indgående stier til en tilstand $2^1 = 2$.

Algoritme 6.10 (Viterbi)
Inddata*: Den modtagne følge v.*

1. *Begyndende til tid $j = m$: Beregn for hver tilstand S_ι den partielle metrik $M([v|c]_j)$ for den indgående sti c. For hver S_ι gem $\tilde{c}$ og dens metrik, $M(v|\tilde{c})$.*
2. *Forøg j med 1. Beregn $M([v|c]_j)$ for samtlige indgående stier, c, til en tilstand ved at addere den indgående gren-metrik, $M(v_i|c_i)$ for tilstanden med metrikken af den overlever fra forrige tidsskridt, som den skal forbindes med. Gem $\tilde{c}$ sammen med $M(v|\tilde{c})$, og slet samtlige andre stier for hver tilstand, S_ι.*
3. *Hvis $j < L + m$, gentag 2. Ellers stop.*

Uddata*: Afkodet følge.*

Det kan vises, at den sidste overlever af Viterbis algoritme netop er den sti, der har den største metrik, det vil sige maximum-likelihood-stien.

Sætning 6.11
Den sidste overlever $\tilde{c}$ af Viterbis algoritme er maximum-likelihood-stien, det vil sige

$$M(v|\tilde{c}) \geq M(v|c) \text{ for } \forall c \neq \tilde{c}.$$

Bevis
Antag, at sidste gren i maximum-likelihood-stien s_{ml} lige er blevet slettet af algoritmen til tid j. Dette betyder, at den partielle metrik af overleveren på dette tidspunkt overstiger den for s_{ml}. Tilføjes nu den tilbageværende del af s_{ml} til overleveren til tid j, vil den totale metrik for denne sti overstige den totale metrik for s_{ml}. Men dette er i modstrid med definitionen af s_{ml} som den sti med den største metrik. Altså kan s_{ml} ikke slettes af algoritmen og må dermed være den sidste overlever. □

Viterbi-algoritmen er derfor optimal i den forstand, at den altid finder maximum-likelihood-stien gennem en trellis. (Lin & Costello, Jr.; 1983, side 316-318), (Forney, Jr.; 1974a, side 228-229)

Viterbi-algoritmen illustreres bedst gennem et simpelt eksempel. Vi skal give et sådant for den binære symmetriske kanal (se figur 6.3) og derefter relatere algoritmen til deep-space kanalen. Antag, at vi har en

binær symmetrisk kanal med overgangssandsynlighed[4] $p < 1/2$ samt at den modtagne følge v er binær ($q = 2$). Da bliver log-likelihood funktionen

$$\begin{aligned} \log P(v|c) &= d(v,c)\log p + (N - d(v,c))\log(1-p) \\ &= d(v,c)\log\frac{p}{1-p} + N\log(1-p), \end{aligned}$$

hvor $d(v,c)$ er Hammingafstanden mellem v og c. Da $\log p/(1-p) < 0$ og da $N\log(1-p)$ er en konstant for alle c, vælger en maximum-likelihood afkoder for en binær symmetrisk kanal den følge $\tilde{c}$, der minimerer Hammingafstanden til v, det vil sige minimerer udtrykket

$$d(v,c) = \sum_{i=0}^{L+m-1} d(v_i, c_i).$$

Når man således anvender Viterbis algoritme på den binære symmetriske kanal bliver $d(v_i, c_i)$ en gren-metrik, og algoritmen finder den sti gennem trellis-diagrammet, som har den mindste metrik, det vil sige stien tættest på v målt i Hammingafstand. Algoritmen er præcis den samme som tidligere bortset fra, at Hammingafstanden nu udgør metrikken i stedet for log-likelihood funktionen, og at overleveren i hver tilstand nu er stien med den mindste metrik. (Lin & Costello, Jr.; 1983, side 10, 320-321)

Viterbi-afkodningen kan altså lige så vel illustreres for en binær symmetrisk kanal som for en anden kanal. Principperne er de samme, blot er udregningerne for den binære symmetriske kanal mere simple, fordi vi i dette tilfælde har at gøre med diskrete Hammingafstande i stedet for reelle sandsynligheder. Lad os se et eksempel.

Eksempel 6.12
Vi tager udgangspunkt i trellis-diagrammet fra eksempel 9.25 i INDFØRELSEN. Vi har en foldningskode med $n = 2$, $k = 1$, tvungen længde $\nu = 2$ og generatorpolynomier

$$\begin{aligned} g_0(D) &= 1 + D^2, \\ g_1(D) &= 1 + D + D^2. \end{aligned}$$

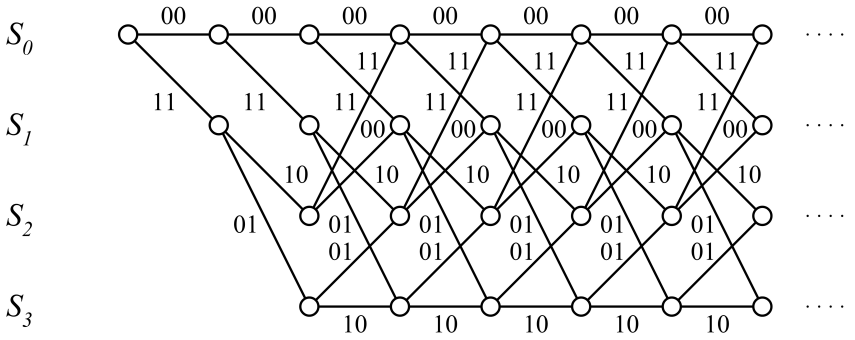
Trellis-diagrammet, hvormed koden er blevet indkodet, er afbilledet på figur 6.2.

Vi anvender i nærværende eksempel et afkodningsvindue eller en afkodningsforsinkelse af længde 15. Koden kommer ind i vinduet fra højre og bevæger sig ud af vinduet mod venstre. Antag, at den modtagne følge for en ikke nærmere specificeret værdi af L er givet ved:

$$v = 01\ 11\ 10\ 10\ 01\ 01\ 00\ 00\ 00\ 00\ldots.$$

Som det fremgår af figur 6.3 påbegyndes eksemplet i iteration 3. Ved begyndelsen af iteration r kender afkoderen den korteste sti (overleveren) til hver af knuderne i den $(r-1)$'te ramme, det vil sige til de $q^\nu = 2^2 = 4$ tilstande. Overleveren bestemmes som dén af de $q^k = 2^1 = 2$ til hver

4 En binær symmetrisk kanal med $p > 1/2$ kan altid konverteres til en binær symmetrisk kanal med $p < 1/2$ ved at ombytte uddataene.



Figur 6.2 Trellis-diagram for foldningskoden i eksempel 9.25 i INDFØRELSEN.

tilstand indgående stier, der har den korteste Hammingafstand til v. Figur 6.2 viser til en given iteration de fire overlevere.

Lad os eksemplificere udregningerne af Hammingafstandene ved iteration 4. Kodefølgerne svarende til iteration 4's fire stier kan aflæses af trellis-diagrammet i figur 6.2. Understregningerne angiver de bit, hvorpå stierne i hver af de fire tilstande afviger fra den modtagne følge v.

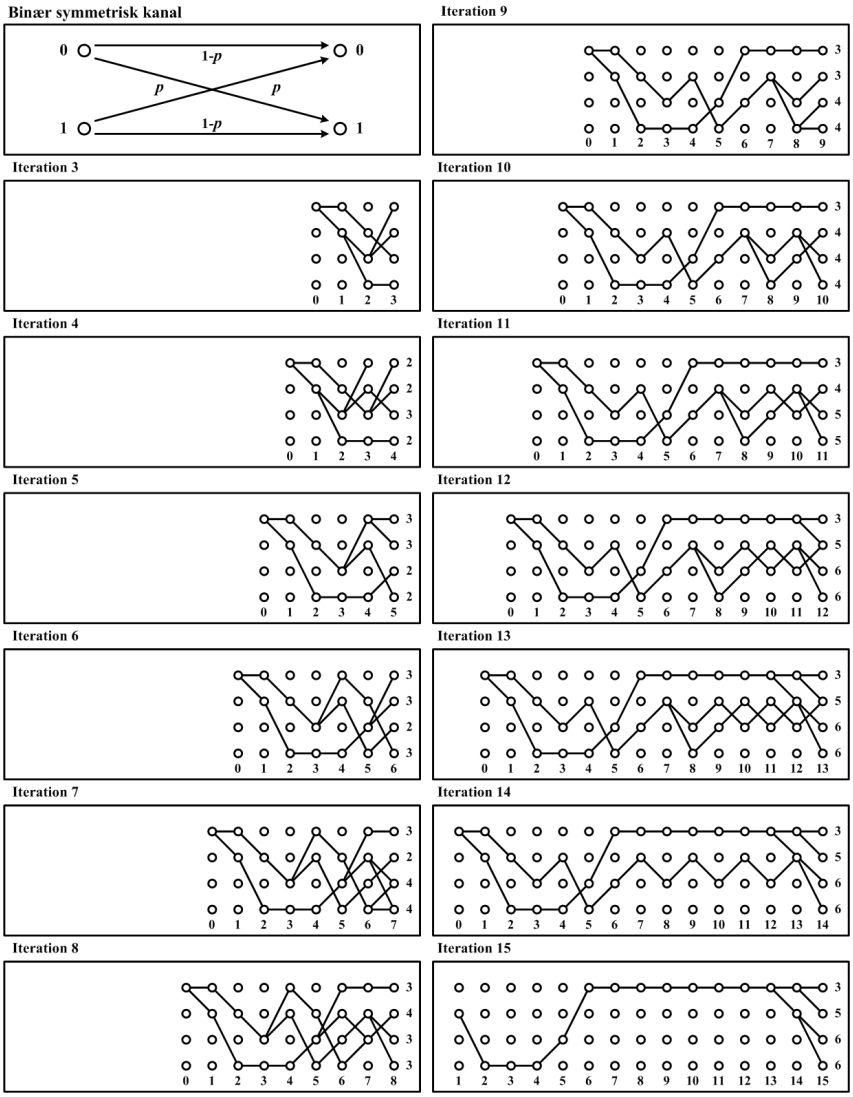
$$\begin{array}{lll} v & 01\,11\,10\,10 & \\ S_0 & \underline{0}0\,11\,10\,\underline{1}1 & d = 2, \\ S_1 & \underline{0}0\,11\,10\,\underline{0}0 & d = 2, \\ S_2 & \underline{1}1\,\underline{1}0\,\underline{0}0\,10 & d = 3, \\ S_3 & \underline{1}1\,\underline{0}1\,10\,10 & d = 2. \end{array}$$

I sidste vindue er den modtagne følge v foreløbig blevet afkodet til

$$11\,01\,10\,10\,01\,11\,00\,00\,00\,00\,\ldots,$$

som har Hammingafstand lig 3 fra v. ◇

Viterbi-afkoderen for den binære symmetriske kanal er et eksempel på en minimumsafstand-afkoder. Ydermere er der tale om et eksempel på en *hard-input* Viterbi-afkoder, fordi den anvender Hammingafstanden som dens metrik. En modtaget følge v bestående af elementer fra kodealfabetet kaldes for en *hårdt* modtaget følge. En modtaget følge bestående af reelle eller komplekse tal kaldes for en *blødt* modtaget følge. Mens komponenterne i en hårdt modtaget følge er elementer i et eller andet legeme, kan komponenterne i en blødt modtaget følge opfattes som symbolvise troværdighedsindikatorer. I forhold til diskussionen af demodulation i kapitel 3 bør det bemærkes, at blødt modtagne følger er udelukket, når man anvender hard-decision demodulation.



Figur 6.3 Øverste venstre vindue: Illustration af den binære symmetriske kanal: p angiver sandsynligheden for at en bit (afbildet til venstre) under transmissionen skifter til den modsatte bit (afbildet til højre). Andet øverste venstre vindue og følgende vinduer: Afkodning med Viterbis algoritme af den modtagne følge $v = 01\ 11\ 10\ 10\ 01\ 01\ 00\ 00\ 00\ 00\ldots$. Til hver iteration vises i vinduet overleveren til hver af trellis-diagrammets tilstande. Hammingafstande er angivet til højre i vinduerne og tidsskridt er angivet i bunden. (Blahut; 2003, side 342)

En *soft-input* Viterbi-afkoder anvender i stedet for Hammingafstand en anden metrik eksempelvis den ovenfor beskrevne log-likelihoods funktion eller euklidisk afstand. Foretager vi for en binær blok- eller foldningskode følgende afbildning

$$\mathbb{F}_2^n \to \{-1, +1\}^n,$$

kan vi tale om en euklidisk afstand frem for Hammingafstanden. Den euklidiske afstand mellem to følger $c, c' \in \mathbb{R}^n$ er da defineret ved

$$d(c, c') = \sqrt{\sum_{i=0}^{n-1} (c_i - c'_i)^2}.$$

Definition 6.13
Den euklidiske minimumsafstand for en kode i $\mathbb{R}^n$ er givet ved den mindste euklidiske afstand mellem to vilkårlige følger, som tilhører koden.

Størrelserne $d_{\min}$ og d_{fri} for henholdsvis blok- og foldningskoder kan oversættes til lignende begreber i euklidisk afstand, idet de i euklidisk afstand blot er givet ved det halve. (Blahut; 2003, side 336-337, 343)

Det kan vises, at for en Gaussisk kanal som deep-space kanalen er maksimering af log-likelihood funktionen det samme som at minimere den euklidiske afstand (se (Blahut; 2003, side 338-339)). Afkodningen med den mindste euklidiske afstand giver den bedste afkodning, hvorfor Viterbi-algoritmen med maksimering af log-likelihood funktionen for deep-space kanalen også giver den bedste afkodning.

6.5 Valg af koder

Gennem rumforskningens historie har transmission af billeder fået en mere og mere fremtrædende rolle, og datakomprimering er derfor blevet en væsentlig del af missionerne. For ICERs vedkommende vil en bitfejl i en transmission muligvis give en forkert pixelværdi i det transformerede billede. Under rekonstruktionen af billedet vil denne bitfejl blive spredt ud i et segment af billedet (jævnfør afsnit 5.7.1). En enkelt fejl kan således forårsage væsentlige ændringer i det rekonstruerede billede. Hvis man ønsker at anvende billedkomprimering, er man derfor nødt til at sænke transmissionens BER væsentligt. Valget af kanalkodning er altså stærkt afhængigt af valget af datakomprimering. Det konkatenerede system har vist sig, at kunne sikre de krævede værdier af BER.

En væsentlig faktor er altså, at det konkatenerede system har kunne opfylde de krav, som har været stillet i missionerne. En anden meget væsentlig faktor, som har influeret valget af koder i MER, er kodestandarderne. Det konkatenerede system har været en del af CCSDS-standarden siden midten af firserne, og af den årsag er det et utroligt velafprøvet system, som er nemt at overføre mellem missionerne. Desuden findes hardwareimplementeringer af systemets afkodere, som er særdeles dyre at skifte ud. For MER-missionens vedkommende har man derfor valgt at benytte CCSDS-standarden.

6.5.1 Andre kodningsstrategier i MER

Foldningskoden, der indgår i det konkatenerede system, er ikke den eneste foldningskode i brug i MER. Udover $(2, 1, 7)$-koden anvendes også en

$(6, 1, 15)$-foldningskode, altså en kode med rate $1/6$ og hukommelse 15. Koden kendes også som *den store foldningskode*. (JPL; 2002, side 8.9)

For en BER på 10^{-6} giver den store foldningskode, som oprindeligt blev udviklet til Galileo-missionen i 1989, en forbedring af transmissionen på omtrent 2 dB. Afkodning er dog ikke en smal sag. Implementeringskompleksiteten for Viterbi-afkodning vokser eksponentielt med kodens tvungne længde, og trellis-diagrammet for den store foldningskode er derfor enormt med sine $2^{14} = 16.384$ tilstande i forhold til den mindre foldningskode, hvis trellis har $2^6 = 64$ tilstande. Om brugen af denne foldningskode forklarede Jørn Justesen:

> Vi andre mener, at det skyldes, at der var en eller anden skør ph.d.-studerende, som fik for meget frit løb, og lavede det der. Sammenlignet med forskellige alternativer, som er meget enklere at implementere, så ser det mærkeligt ud, at det der Storm P.-apparat står der og overhovedet nogensinde har fået lov til at blive lavet. Så, ja, spørgsmålet om hvorfor det lige er netop den kode der er valgt, det er til dels et godt spørgsmål. Der er to mærkelige svar: Det ene er, at det stammer fra tidernes morgen, hvor der stadig væk ikke eksisterede nogen teknik; og det andet er, at det skyldes en eller anden skør studerende. (Justesen & Høholdt; 2004b, side 4 i INTERVIEWS)

Jon Hamkins fra JPL bekræftede i en elektronisk brevveksling, at den store foldningskode er taget i brug i MER, men gav ingen begrundelse herfor. En mulig begrundelse fik vi imidlertid af Ralph Kötter fra MIT, der udtrykte sig på følgende bramfrie vis:

> Oh, they screwed up! There was some problem... they didn't get enough energy or enough power from some of the solar panels, I think. [...] They screwed up; they actually changed the code in flight. They first had a low memory convolutional code, but that code didn't give them the gains they needed, so they changed the code in flight and made up for it. (Kötter; 2005, side 29 i INTERVIEWS)

Som nævnt benytter man i stor stil transmission til Mars Odyssey i stedet for direkte transmission til Jorden. Dette har ændret i kodningsstrategien ved at flytte den egentlige kodning fra roverne og over på Mars Odyssey. På den måde slipper man for at bruge rovernes tid og energi. Belastningen kan i stedet overføres til Odyssey, for hvilken tid og energi ikke er en mangelvare i samme omfang. Odyssey anvender samme kanalkodning som MER.

6.6 Afrunding

For rovernes vedkommende er energi en mangelvare, og effekten af det signal, der kan transmitteres er derfor stærkt begrænset. For overhovedet at være i stand til at kommunikere med Jorden er kanalkodning således en nødvendighed. Når kildekodning anvendes i en mission, er kanalkodning endnu vigtigere, fordi komprimerede data er meget følsomme overfor transmissionsfejl.

Det kanalkodningssystem, som anvendes i MER, sikrer en BER, der er tilstrækkelig for missionens kildekodning. Kanalkodningen udgøres af et konkateneret system, som består af en indre $(2, 1, 7)$-foldningskode (eller en $(6, 1, 15)$-foldningskode) og en ydre $(255, 223)$-Reed-Solomon kode. Systemet er en del af CCSDS-standarden for kanalkodning i deep-space anvendelser. Afkodning af foldningskoden foregår i overensstemmelse med CCSDS anbefaling ved maximum-likelihood Viterbi-afkodning. Afkodning af Reed-Solomon koden er ikke omfattet af CCSDS anbefaling. Vi har valgt at illustrere afkodningen ved Sugiyamas algoritme.

7 Bag om matematikken i Mars-missionerne

> Everything is a cost-benefit analysis. The whole space system is a cost-benefit analysis. (Folkner; 2005b, side 70 i INTERVIEWS)

I dette kapitel skal vi diskutere spørgsmålene i problemformuleringen. De to første spørgsmål i problemformuleringen er udelukkende rettet mod kilde- og kanalkodning, men mange af de observationer, vi gør i forbindelse med disse, vil vi videreføre til diskussionen af problemformuleringens tredje spørgsmål. Spørgsmål 2 og 3 i problemformuleringen omtaler de *aktuelle* Mars-missioner. Vi vil kort præcisere hvilke missioner, der hører under denne betegnelse.

De aktuelle missioner til Mars omfatter for det første Mars Pathfinder, som var den første mission i tyve år til at lande på Mars' overflade efter 70'ernes succesfulde Viking-missioner. Pathfinder var en såkaldt 'low-cost planetary discovery mission', som omfattede både en lander og den lille rover Sojourner. Pathfinder-missionen var forgængeren for MER-missionen, som i et vist omfang er baseret på erfaringerne fra Pathfinder. MER er naturligvis også inkluderet i de aktuelle missioner.

Udover de to rover-missioner har NASA opsendt et par orbitere, nemlig Mars Global Surveyor (MGS), som gik i kredsløb om Mars i september 1997, samt 2001 Mars Odyssey, der nåede frem i oktober 2001. NASAs sidst opsendte orbiter er Mars Reconnaissance Orbiter (MRO), som vil nå Mars i marts 2006. Orbiter-missionerne er ligeledes 'low-cost' missioner og en del af de aktuelle missioner.

Udover de nævnte missioner, vil vi også lejlighedsvist inddrage fremtidige missioner, eksempelvis Mars Science Laboratory (MSL), som formentlig skal opsendes i 2009. Denne består af en større og mere avanceret rover, som skal foretage videnskabelige eksperimenter på Mars. Yderlige information om missionerne findes i appendiks A.

7.1 Kilde- og kanalkodningens betydning for MER

I (JPL; 2002) angives en række succeskriterier for MER-missionen. For rovernes vedkommende er der tale om otte kriterier, hvoraf to lyder:

1. At køre roverne til samlet otte forskellige lokaliteter og anvende det videnskabelige udstyr til at undersøge Mars' geologi ved disse lokaliteter. Det skal så vidt muligt forsøges at maksimere forskelligheden af disse lokaliteter for at opnå stor diversitet i undersøgelserne. Dog kommer missionens sikkerhed i første række.

2. At én af roverne skal foretage en overfladekørsel med en samlet længde på mindst 600 meter og helst 1.000 meter. (JPL; 2002, side 2-1,2-2)

To af missionens succeskriterier afhænger altså af rovernes evne til at bevæge sig på Mars' overflade. Vi talte med Aaron Kiely om, hvordan en (hurtig) komprimeringsalgoritme påvirker denne evne. Han fortalte:

> A lot of the navigation is done based on people on the ground and things like that. At some point if you are just getting back all your images they need to be compressed and largely what they are doing is sending down images to learn about whether there is water on Mars. If the compression is running slowly it limits how many images they can get down and then it limits how far they can drive. (Kiely; 2005, side 39 i INTERVIEWS)

Under kørsel på overfladen tages billeder af landskabet omkring roverne. Disse billeder transmitteres til Jorden. Her anvendes billederne til dels at vurdere, om der i landskabet er interessante genstande (for eksempel sten), som kræver nærmere undersøgelse, og dels til at vurdere, om videre fremfærd bringer rovernes sikkerhed i fare. I de tilfælde, de transmitterede billeder viser potentielle farer for roverne i landskabet, bliver der brugt lang tid på at vurdere sikkerheden af den videre fremfærd, og i de tilfælde betyder tid spildt på transmission af billeder ikke så meget. I de tilfælde, roverne befinder sig i et sikkert terræn og bare skal videre, vil man til gengæld gerne have billederne ned så hurtigt som muligt for at kunne afstikke en kurs til roverne, så de kan køre videre. Evnen til at bevæge sig hen over overfladen hænger altså i høj grad sammen med evnen til hurtigt at kunne transmittere billeder til Jorden.

For billeder, der skal bruges til navigation, er komprimeringen med tab særligt anvendelig, da billederne gerne skal hurtigt ned, mens det er mindre vigtigt om kvaliteten er helt i top. En simpel udregning illustrerer vigtigheden: Spirit og Opportunity forlod deres landingsmoduler og kørte ud på Mars' overflade den 15. og den 29. januar 2004. Den 7. februar, altså efter omtrent 34 dages sammenlagt operationstid, havde roverne transmitteret 5.132 billeder (Kiely & Klimesh; 2004, side 1). Disse billeder var komprimeret med ICER til $1,13$ bit per pixel, hvilket betød en datamængde på 3.098 Mbit eller $387,3$ Mb. For MERs transmissionsrate på 168.000 bit per sekund fås en transmissionstid på 5 timer og 22 minutter eller cirka $9,5$ minutter om dagen i 34 dage. Hvis data i stedet var blevet sendt ukomprimeret (12 bit per pixel i stedet for $1,13$ bit per pixel), havde den samlede datamængde været 4.112 Mb. Denne datamængde ville med samme rate kræve en transmissionstid på godt 57 timer svarende til over 100 minutter om dagen i de 34 dage. I forhold til ukomprimeret transmission vinder man altså med ICER over halvanden time om dagen, i hvilke roverne er fri for at skulle transmittere.

Matthew Klimesh fortalte os, at roverne (på interview-tidspunktet) havde taget mere end 60.000 billeder, som var sendt til Jorden (Klimesh; 2005, side 46 i INTERVIEWS). Hvis transmission skulle have foregået uden komprimering var en stor del af disse billeder sandsynligvis blevet nedprioriteret og udeladt. Dette havde uden tvivl betydet et mindre udbytte af missionen.

Overordnet set er billedkomprimering altså en forudsætning for at få så meget mission for pengene som muligt. Hvis roveren skal bruge meget tid og energi på billedtransmission, når den simpelthen ikke andet. Billedtransmission må altså opfattes som en flaskehals i missionen, hvis succes derfor i høj grad afhænger af en effektiv algoritme til billedkomprimering.

Som beskrevet i kapitel 3 skal den energi, der er nødvendig for at sikre pålidelig transmission *uden* kanalkodning, være af en størrelsesorden, som er helt urealistisk i en mission som MER. Kanalkodning må altså siges at være en af de helt grundlæggende forudsætninger for kommunikation og dermed, at missionen overhovedet kan gennemføres. Hvis kildekodning skal kunne komme på tale, har man brug for yderligere reduktion af sandsynligheden for transmissionsfejl for at sikre, at dekomprimering kan foretages med tilfredsstillende resultater. Hvis man et øjeblik skal se ud over MER, kan man med stor rimelighed sige, at kanalkodning er vigtigt for alle rummissioner, der er afhængige af kildekodning, eller som ikke har meget energi til rådighed.

Udover at være en forudsætning for kildekodning er der et andet forhold, som gør kanalkodning vigtig. Dette skal vi kort se på i det følgende.

En af de helt store udgifter, der er forbundet med en mission som MER, er udgifter i forbindelse med opsendelsen. William Folkner anslog disse udgifter til at udgøre mellem en fjerdedel og en tredjedel af de samlede udgifter (Folkner; 2005b, side 73). Opsendelsen må altså betegnes som en anden flaskehals for missionen. For opsendelsen betyder teknologi og viden om raketter næsten alt. Folkner fortalte om udviklingen inden for dette videnskabelige felt:

> ... space is up against – not physics – but chemistry. We have to get mass into space, and it costs energy to get mass into space, and velocity to get mass into space, which means that for everything that goes up we have to send something down. So there has been no fundamental advancement in the physics or the chemistry if you like for sending things into space for forty years probably. (Folkner; 2005a, side 67 i Interviews)

Den ikke-eksisterende udvikling i kemien bag raketterne har betydet, at udgifter til opsendelsen har været mere eller mindre konstante igennem rumindustriens (senere) historie.

Som nævnt i kapitel 2 har den generelle trend hos NASA i de senere år været at udføre mange mindre 'low-cost' missioner. For de missioner, der ikke har enorme pengesummer til rådighed, udgør omkostninger forbundet med opsendelsen en væsentlig del af de samlede udgifter. Den mest effektive metode til at reducere en missions omkostninger er derfor at reducere den masse, som skal opsendes. Alle tiltag, der kan bidrage hertil, er væsentlige for en mission.

Som nævnt i kapitel 3 giver de dB, man vinder ved kanalkodning, mulighed for at reducere eksempelvis antennestørrelser. Sådanne reduktioner af materiel betyder naturligvis reduktion af masse. Kanalkodning gør det altså muligt at reducere massen af det opsendte fartøj og dermed indirekte muligt at reducere omkostningerne ved opsendelsen. Det samme gælder faktisk også for kildekodningen. Folkner forklarede:

> ... if you can preprocess the data and always send the good

> ten percent back instead of sending it all back and editing on the ground then that is more expensive in terms of processing, but you save so much in terms of the mass of the communication system that you have to fly. You can have a much more capable instrument for the same telemetry then. So data compression onboard and data processing onboard is very important. (Folkner; 2005b, side 73 i INTERVIEWS)

Kildekodning kan altså anvendes til at reducere to størrelser i en rummission: Transmissionstiden og massen af kommunikationssystemet. Disse to størrelser hænger sammen. Hvis man kan tolerere en stor transmissionstid, kan man klare sig med et mindre (lettere) kommunikationssystem og omvendt. I praksis er der altså et såkaldt 'trade-off' mellem massen af kommunikationssystemet og den tid, systemet skal bruge til transmission. Kildekodning gør det muligt at forbedre dette 'trade-off'.

Den tidligere omtalte ændring af kommunikationen mellem roverne og Jorden, der er foretaget undervejs i missionen – transmission via Mars Odyssey – er særdeles vigtig i denne sammenhæng. Masseproblemet gælder nemlig ikke kun opsendelsen fra Jorden, men også opbremsningen ved ankomsten til Mars. Hvis man kan flytte en del af den masse, som egentlig burde være om bord på roverne i form af kommunikationsudstyr, ud i kredsløb om Mars, opnår man en stor massebesparelse i opsendelsen. Umiddelbart kan dette lyde mærkeligt, da massen af kommunikationsudstyret stadig skal løftes ud af Jordens tyngdefelt. Imidlertid skal massen ikke bremses op i Mars' atmosfære, hvilket betyder, at man sparer en del brændstof til de bremseraketter, som affyres kort inden landing på Mars' overflade. Massen af denne mængde brændstof slipper man for at medbringe, hvilket resulterer i en massebesparelse ved opsendelsen. Alternative kommunikationsstrategier som den nævnte ser da også ud til at være på vej til at blive en fast del af rover-missionerne.

Historisk set har kilde- og kanalkodning betydet en væsentlig forøgelse af rumfartøjernes evne til at transmittere. Som det fremgår af appendiks A, er transmissionsraten vokset enormt de sidste fyrre år. Mariner 4 fra 1964, som hverken gjorde brug af kilde- eller kanalkodning, kunne således transmittere med en rate på $8,33$ bit per sekund. I forhold til denne tidlige rummission har MER en rate, der er omtrent 20.000 gange så stor. I McEliece (2004a) opsummeres en undersøgelse af kilde- og kanalkodningens bidrag til denne forøgelse. Det konkluderes, at de to teoriers bidrag til forøgelsen udgør omtrent 37 procent af den vundne rate. De resterende 63 procent stammer fra de samlede fremskridt, der er gjort inden for fysikken og teknologien i rummet. (McEliece; 2004a, side 4-5)

Overordnet set er de to teorier, som vi har beskæftiget os med, altså særdeles vigtige for MER. Sammen giver anvendelsen af teorierne fordele, som betyder besparelser af både tid og masse. Begge disse besparelser kan mere eller mindre direkte omregnes til penge. Citatet, som indleder dette kapitel, illustrerer, hvorfor økonomiske besparelser skulle gøre teorierne særligt relevante.

I de følgende tre afsnit skal vi diskutere problemformuleringens spørgsmål 2. For at kunne besvare dette spørgsmål er vi nødt til at præcisere hvilken del af udviklingen inden for kilde- og kanalkodning, vi vil interessere os for. Da vi beskæftiger os med rummissioner, hvis planlægning er

foregået fra starten af 1990'erne og frem, synes vi, at det er rimeligt at se på udviklingen i samme periode.

7.2 JPLs betydning for kildekodningens seneste udvikling

På MIT talte vi med Gilbert Strang om udviklingen inden for kildekodning. Han fremhævede følgende ved denne udvikling:

> I think it was remarkable that wavelets made it into JPEG standards because a change of standards is such a big deal that it really has to be better. I'm sure that's true also with NASA, I'm sure they have codes in which they've invested thousands of man-years. If someone comes along with a better idea, it has to be ten times better. (Strang; 2005, side 32 i INTERVIEWS)

Ovenstående er interessant for os fordi både kilde- og kanalkodning er underlagt standarder, eksempelvis CCSDS- og JPEG-standarderne. Som beskrevet i kapitel 6 har især CCSDS-standarderne betydet en vis fastlåsning af kanalkodningen i rumanvendelser. Styrken ved standarderne er, at alle gør brug af dem, men dette forhold gør dem samtidigt meget dyre og besværlige at udskifte.

Gilbert Strang siger egentligt, at hvis man gerne vil identificere vigtige udviklinger inden for kilde- og kanalkodning, så skal man blot se på, hvornår de etablerede standarder bliver ændret. Denne idé vil vi benytte os af her.

Strang nævner selv, at wavelets har betydet en ændring af JPEG-standarderne. På de institutioner, vi har foretaget interviews, var der da også en generel enighed om, at wavelets har drevet kildekodningen videre, og vi finder det derfor rimeligt at lade den seneste udvikling i kildekodning være ensbetydende med indførelsen og brugen af wavelets i teorien.

Muligheden for at udskifte kildekodning i rumanvendelser er generelt god, fordi kildekodningen implementeres i software, som kan ændres fra mission til mission. Den konkrete processor, som skal afvikle denne software, *er* jo på forhånd godkendt til brug i rummet. Umiddelbart kan kildekodningen i Mars-missionerne altså godt inkludere udvikling, som har en generel matematisk indflydelse på teorien.

MER er den første rummission, som benytter en waveletbaseret komprimeringsalgoritme, og vi vil derfor koncentrere os om MERs algoritme ICER. ICER er udviklet, fordi det kommercielle produkt JPEG, som var til rådighed, da MER skulle planlægges, simpelthen ikke var godt nok. Ved første øjekast er ICER da også et splinternyt produkt, som benytter helt nye teknikker.

Når man har beskæftiget sig et stykke tid med ICER, får man dog efterhånden dannet sig et noget andet billede af algoritmen. Overordnet set må ICER siges at være en algoritme, som er sammensat af en masse forskellige teknikker, som allerede *var* udviklet, da udformningen af algoritmen blev påbegyndt. ICERs heltalstransformationer er ganske vist blevet analyseret så sent som i artiklen Adams & Kossentini (2000), men denne analyse er ikke JPL-relateret. For modelleringens og entropikodningens vedkommende gælder ligeledes, at de teknikker, der anvendes, er relativt nye teknikker, som alle er udviklet *uden for* JPL. Betragter

man imidlertid teknikkerne i en mere generel matematisk ramme, så er der ingen nyudviklinger. Som eksempel på dette kan nævnes ICERs entropikoder. Den konkrete sammenflettede entropikodning, som benyttes i ICER, stammer fra slutningen af 1990'erne, men ideen om sammenflettede koder er omkring ti år ældre, og selve Golomb-koderne stammer fra 1966. Matematikken i ICER *var* altså allerede på plads, da algoritmen skulle udvikles.

Vi talte med Aaron Kiely om at arbejde med kildekodning på JPL. Han kommenterede:

> A lot of the data compression stuff that we are doing is largely just the same you would do for an image compression algorithm for the Internet. So there are very big chunks that are the same as for industry. We didn't invent wavelets here at the JPL because we weren't really forced to it. (Kiely; 2005, side 42 i Interviews)

Der er altså ikke den store forskel på at udvikle en komprimeringsalgoritme til rummet og til industrien. Man har derfor på JPL kunnet holde øje med industriens algoritmer og plukke de teknikker herfra, som blev skønnet bedst egnet. De omstændigheder, der er særlige for rummet, er stort set kun kravet om lav beregningskompleksitet. Vi spurgte Kiely, om det er rimeligt at karakterisere udviklingen af ICER som »giving already established theories a small twist«, og det syntes han (Kiely; 2005, side 42 i Interviews). ICER er rent faktisk blevet udviklet sideløbende med JPEG-2000, og algoritmerne har derfor mange af de samme træk. Dog har man for ICERs vedkommende fravalgt JPEG-2000's aritmetiske kodning for at undgå at introducere for stor beregningskompleksitet i indkodningen. Matthew Klimesh fortalte os, at man under udviklingen af ICER havde skævet en del til udviklingen af JPEG-2000, men at det på grund af deadlines ikke var muligt at vente på færdiggørelsen af JPEG-2000 (Klimesh; 2005, side 45 i Interviews). Kiely kommenterede også dette:

> By the time we gave them the algorithm, JPEG-2000 was just a committee that talked about things. There were no existing official JPEG-2000 that we could have given to them. If the mission had landed five years later we might just have given them JPEG-2000. (Kiely; 2005, side 40 i Interviews)

Ovenstående tyder kraftigt på, at der ikke er noget grundlæggende i ICER, som ikke findes i JPEG-2000. Om ICER vil få en betydning for fremtidens kildekodning i rumanvendelser er et åbent spørgsmål. For orbitere, hvis transmission i ringe grad er begrænset af tid eller energi, er det ikke sandsynligt, at ICER kommer i anvendelse. I tidligere orbiter-missioner har man ikke benyttet komprimering med tab, og der er ingen grund til at tro, at man vil gøre det i fremtidige missioner. Mars Reconnaissance Orbiter, der er opsendt i 2005 benytter da heller ikke ICER, men algoritmen FELICS (*Fast, Efficient, Lossless Image Compression System*)[1]. For Mars Science Laboratory med planlagt opsendelse i 2009, mente Kiley dog, at ICER vil blive brugt til komprimering af billeder fra roverens 'hazard avoidance' kameraer. Anvendelse af ICER til andet en navigationsbilleder

1 Denne er beskrevet i Howard & Vitter (1993).

forekommer ikke at være realistisk. Vores samtale med Matijevic afslørede, at ICER var på vej ud af de videnskabelige anvendelser, for hvilke tabsfri komprimering kombineret med den omtalte relækommunikation via Mars Odyssey er mere effektiv.

For kort at opsummere er ICER altså en algoritme, der er udviklet ved at stykke eksisterende teknikker sammen, og som samtidigt har været inspireret af JPEG-2000. Sandsynligheden for at udviklingen af ICER skulle have betydet – eller vil komme til at betyde – noget for den generelle udvikling inden for kildekodning, mener vi derfor er meget ringe. Vi spurgte Aaron Kiely, om han opfattede sit arbejde som matematisk arbejde eller som software-udvikling. Han svarede:

> I'm really doing both. I don't know if it is always that easy to draw a clear distinction between if what is being done in one case is software but not mathematics. We spent a while in algorithm development coming up with an entropy encoder that is fairly efficient for the coding that is done in ICER which I think is a lot closer to pure mathematics but we also spent quite a while making ICER run faster. (Kiely; 2005, side 35 i Interviews)

Lad os et øjeblik overveje dette. At Kiely karakteriserer sammenstykning af en entropikoder som *ren* matematik, synes vi umiddelbart er ejendommeligt. Selv vil vi nok nærmere kalde det en *anvendelse* af matematik. Kielys opfattelse skyldes muligvis, at sammenstykning af en entropikoder, er det tætteste, han kommer på matematik i sit daglige arbejde, og derfor opfatter han arbejdet som *ren* matematik. Vi mener, at udtalelsen forstærker indtrykket af JPL som et foretagende, der i høj grad anvender eksisterende matematik, men kun i ringe grad udvikler det selv. En diskussion af, hvad ren matematik egentlig *er*, ville nok være på sin plads her. Diskussionen er imidlertid kompleks, og vi vil derfor ikke direkte indlade os på den i dette speciale.

I stedet vil vi gå videre til diskussionen af JPLs betydning for kanalkodningens udvikling. For at være bedre rustet vil vi først præsentere to klasser af koder, som er i fokus i diskussionen.

7.3 Kort om turbokoder og LDPC-koder

Inden for kanalkodning er der siden starten af 1990'erne sket to spændende udviklinger, som er relevante for vores diskussion. Den ene af disse udviklinger er opdagelsen af en ny klasse af koder, kaldet *turbokoder*, og den anden er en ny forståelse af en gammel klasse af koder, som kaldes *LDPC*-koder.

Da vi ikke har omtalt disse koder tidligere vil vi meget kort præsentere dem her.

7.3.1 Turbokoder

Turbokoderne så første gang dagens lys i 1993 i artiklen *Near Shannon limit error-correcting coding and decoding: Turbo codes* af Berrou, Glavieux og Thitimajshima.

En turbokode er kort fortalt en lineær blokkode, som er konstrueret fra en termineret systematisk indkodet foldningskode. Den overordnede ide er for en datastrøm at foretage to forskellige indkodninger og beholde paritetssymbolerne for begge indkodninger. En sådan indkodning kaldes parallel indkodning.

Definition 7.1 (Turbokode)
Lad $g(x)$ være et polynomium af grad r. En turbokode *er en lineær blokkode med en generatormatrix på formen*

$$G(x) = \begin{pmatrix} I & (g(x)) & \prod(g(x)) \end{pmatrix},$$

hvor $(g(x))$ er matricen, hvis rækker består af cykliske skift af koefficienterne for $g(x)$, og $\prod$ er en $k \times k$ permutationsmatrix.

Lad $a(x)$ være et datapolynomium, og lad $c(x)$ være indkodningen af $a(x)$. Så er $c(x)$ givet ved

$$c(x) = a(x)G(x) = \begin{pmatrix} a(x) & a(x)g(x) & a(x)\prod g(x) \end{pmatrix}.$$

Vi kan opfatte $a(x)\prod g(x)$ som $a'(x)g(x)$, hvor $a'(x)$ er en permutation af datasymbolerne. Indkodningen af en datastrøm $a(x)$ kan altså skrives som

$$c(x) = \begin{pmatrix} a(x) & a(x)g(x) & a'(x)g(x) \end{pmatrix}.$$

Dette illustrerer, hvorfor turbokoden siges at være konstrueret fra en foldningskode. Lad $\mathcal{C}$ være foldningskoden med generatormatrix $\begin{pmatrix} I & g(x) \end{pmatrix}$. For en datastrøm $a(x)$ består turbokodens paritetssymboler så af $\mathcal{C}$'s paritetssymboler, samt de paritetssymboler, der stammer fra $\mathcal{C}$'s indkodning af $a'(x)$.

Det har ved eksperimentelle studier vist sig, at der findes turbokoder, som for en BER på omkring 10^{-6} giver en kodning, der ligger tæt ved Shannon-grænsen. Man har dog ikke endnu fundet nogen generel opskrift, som gør det muligt at konstruere sådanne gode koder. De eksisterende gode turbokoder er fundet gennem eksperimenter og simuleringer.

Afkodning af en turbokode kan foretages ved en iterativ[2] process, i hvilken turbokoden opfattes som bestående af to foldningskoder. Den første afkodning foretages på baggrund af den del af kodeordet, som svarer til $\begin{pmatrix} a(x) & a(x)g(x) \end{pmatrix}$. Denne afkodning fungerer så som et estimat for troværdigheden af hvert symbol i datastrømmen $a(x)$ repræsenteret ved en *troværdighedsvektor*. Afkodning nummer to, der foretages på baggrund af $\begin{pmatrix} a(x) & a'(x)g(x) \end{pmatrix}$, opdaterer troværdighedsvektoren og afgiver denne vektor til afkodning nummer et. Denne procedure gentages[3], indtil man opnår en tilfredsstillende afkodning. (Blahut; 2003, side 329)

Udover at give en kodning, som i nogen tilfælde ligger tæt op ad Shannon-grænsen, er der et yderligere forhold, som gør turbokoder til et virkeligt godt bud på fremtidens kanalkodning. Jon Hamkins fortalte:

> When turbo codes came it was just a tremendous improvement after what had been decades of usage of these other codes. Nobody believed that it was true because the codes jumped right

2 Et eksempel på en iterativ turboafkodning, som tager udgangspunkt i Viterbi-algoritmen, er givet i Justesen & Høholdt (2004a).
3 Eksperimentelle studier tyder på, at ti til tyve gentagelser er passende.

> over the cut-off rate. The cut-off rate is... it was widely believed that without exponentially increased complexity of your decoding operation you couldn't exceed a certain performance limit called the cut-off rate. These turbo codes just jumped right over it and this led to a renaissance of channel coding research... the first thing everybody did was to rush home to simulate these turbo codes and indeed they performed much better than the other codes even though they were composed of the same old convolutional codes that everybody had been looking at all along. (Hamkins; 2005, side 48 i INTERVIEWS)

Turbokoderne er altså nær-optimale i forhold til Shannon-grænsen og samtidigt simple at afkode. For den rejse i kanalkodningens univers, som blev indledt kort tid efter Shannons 1948-artikel, er turbokoderne således tæt på endestationen. Shannon-grænsen er jo en 'virkelig' matematisk grænse, som ikke kan overskrides uden en fundamental ændring i den måde, der kommunikeres på. Store forbedringer i forhold til turbokoder skal altså ikke komme i form af bedre kodningsresultater, men snarere i form af lavere kompleksitet.

7.3.2 LDPC-koder

Low-Density Parity-Check koder, i daglig tale *LDPC*-koder, er en del af en større klasse af koder, der blev udviklet af Robert Gallager i 1962, og som derfor kaldes Gallager-koder.

Gallager-koder er blokkoder, der helt ekstraordinært er defineret ved deres paritetscheckmatrix.

Definition 7.2 (Gallager-kode)
En (r, s) Gallager-kode *er en lineær kode, for hvilken der findes en paritets-check-matrix* H*, der opfylder, at hver søjle har* r *1-taller, og hver række har* s *1-taller.*

Gallager-koder er mest interessante, når bloklængden n er stor, og r og s er små i forhold til n, fordi dette gør koderne lette at afkode. Når en Gallager-kode opfylder $r \leq \log_2 n$ kaldes den en LDPC-kode.

For en LDPC-kode kontrolleres hvert kodesymbol af r ligninger, som hver især summerer s kodesymboler. Afkodning af LDPC-koderne kan foretages ved hjælp af en såkaldt *flipping*-algoritme, som i nogen tilfælde giver en meget effektiv kodning[4]. Jon Hamkins fortalte om koderne:

> ... in some cases LDPC-codes are outperforming even turbo codes which nobody appreciated before. Now it turns out that turbo codes and LDPC codes if they're designed properly are comparable in performance but the advantage of LDPC codes is that they are much less complex to decode than turbo codes and they have the advantage that the particular structure of the decoder is parallelizable. That meant that even though they were more complex you would still want them because of the fact that you can distribute the computations. (Hamkins; 2005, side 48 i INTERVIEWS)

4 Se eventuelt (Blahut; 2003, side 354-355).

LDPC-koderne giver altså i nogen tilfælde de ovenfor nævnte forbedringer i afkodningskompleksiteten i forhold til turbokoderne samtidigt med kodning i nærheden af Shannon-grænsen.

Besvarelsen af spørgsmål 2 af problemformuleringen involverer *udviklingen* inden for kanalkodning. Dette vil vi konkretisere her ved at lade *udvikling* betyde *udviklingen af turbo- og LDPC-koder*. De forskere, vi talte med på både MIT og JPL, har på spørgsmålet om, hvilke interessante udviklinger, der er sket inden for kanalkodning i de seneste år, enstemmigt peget på udvikling af disse koder. Som vi skal diskutere nedenfor, er koderne tillige ved at blive indført i CCSDS-standarderne, hvilket, som vi tidligere har nævnt, tyder på, at de markerer en vigtig udvikling i kanalkodning. Vi mener derfor, at man med rimelighed kan besvare kanalkodningsdelen af problemformuleringens spørgsmål 2 ved at se på de aktuelle Mars-missioners bidrag til udviklingen af disse koder.

7.4 JPLs betydning for kanalkodningens seneste udvikling

Mars Global Surveyor, Pathfinder, Mars Odyssey og MER-missionen har alle gjort brug af en kodningsstrategi baseret på et konkateneret system bestående af forskellige indre foldningskoder og en ydre Reed-Solomon kode med parametre $(255, 223)$. Det konkatenerede system, der anvender foldningskoden med parametre $(2, 1, 7)$ og den nævnte Reed-Solomon kode, er et gammelt system, som første gang blev beskrevet og afprøvet i 1970 (McEliece & Swanson; 1994, side 29). Selve konkateneringsideen stammer fra Forneys arbejde i 1966. Viterbi-algoritmen, der afkoder foldningskoden, stammer fra 1967, og Sugiyamas algoritme til Reed-Solomon koden stammer fra 1975 (Berlekamp-Masseys algoritme er ældre endnu). Den vigtigste bestanddel i Sugiyamas algoritme er Euklids algoritme, som jo om noget er gammel. Den første anvendelse af det konkatenerede system fandt sted i 1985 under Voyager 2's transmission af billeder fra Uranus. Den store $(6, 1, 15)$-foldningskode, som har været om bord på Pathfinder (og MER), har været kendt siden slutningen af 1980'erne. (Wicker; 1998, side 2138-2140), (Blahut; 2003, side 226)

Missionerne har altså gjort brug af kodningsstrategier, som ikke involverer hverken turbo- eller LDPC-koder, og som blev udviklet for flere årtier siden. Vi spurgte Jon Hamkins om, hvorfor man ikke har valgt at anvende turbokoder i MER. Han fortalte om valget af kodning:

> The process of flight qualification is very long actually. Even though turbo codes was 1993 it was less than a couple of years ago that turbo codes were flying in space. The process from invention to demonstrating it in a laboratory hardware like we have here to flight qualifying them on radiation hardened parts... and you know missions that are signing up for a very complicated space craft... they are out to minimize risks so they want stuff that has been flown in previous missions, they don't want something new. It's kind of contrary to the spirit of exploration. They don't want risks even though we are confident that it works... it is a risk to a mission if it hasn't flown before. (Hamkins; 2005, side 49 i INTERVIEWS)

Valget af kodningssystemerne i Pathfinder og MER er dybest set be-

grundet i, at disse systemer er sikre valg. Som Hamkins svar antyder, er det i realiteten ikke forskerne ved JPL, der vælger hvilken kanalkodning, der skal anvendes, men personer højere oppe i hierarkiet, for hvem risikominimering spiller en større rolle end optimering af kodningen i missionen. Rummissionerne er enorme prestigeprojekter, som for alt i verden ikke må slå fejl, hvilket sandsynligvis betyder, at matematikken i missionerne i højere grad er styret af politiske faktorer end af videnskabelige.

CCSDS-standardiseringen af koderne har betydet, at der er investeret store summer i implementeringer af kodningssystemerne i teknologi. Denne teknologi er efterhånden velafprøvet og udgør således et langt mindre usikkerhedsmoment end nyudviklede koder og dertil hørende teknologi ville gøre. Hamkins svar illustrerer i øvrigt, at den tidsskala, der arbejdes med i forbindelse med at få nyudviklet matematik eller teknologi ind i missionen, er så stor, at turbokoderne først for nyligt er nået ind i rumanvendelserne. Det skal her nævnes, at turbokoderne i en vis forstand er lidt af et eksperiment i hvert fald matematisk set. De koder, man har fundet, kan man naturligvis anvende, men det er faktisk ikke på nuværende tidspunkt afklaret, *hvorfor* turbokoderne er så gode, som de er.

Den første Mars-mission, der benytter turbokoder, er Mars Reconnaissance Orbiter, som blev opsendt i august 2005. MRO anvender en turbokode med parametre $k = 8920$ og $n = 53544$, hvilket giver en rate på $1/6$. Vi spurgte Jon Hamkins om JPLs arbejde med turbokoderne:

> JPL's involvement in the development of turbo codes was on the standardization of the particular codes that are used now in deep space communications. The Mars Reconnaissance Orbiter mission, which is launching in 2005 is using turbo. (Hamkins; 2005, side 48 i Interviews)

JPL *er* altså rent faktisk involveret i arbejde med turbokoder. Arbejdet er koncentreret omkring at standardisere konkrete koder til fremtidigt brug i rummet. Dette arbejde har betydet, at turbokoderne er blevet en del af CCSDS-standarderne. CCSDS omtaler i alt tyve forskellige turbokoder: Fem værdier for informationsbloklængden k; 1738, 3568, 7136, 8920 og 16384 med hver fire tilhørende værdier for bloklængden n, som giver rater på $1/2$, $1/3$, $1/4$ og $1/6$. (CCSDS; 2002, side 4.3)

JPLs arbejde har stor betydning for CCSDS-standarderne og vil på lang sigt få en kæmpe betydning for, hvilken kanalkodning, der anvendes i rummet, fordi koderne vil blive implementeret i hardware. For kanalkodning opfattet som en matematisk teori har dette arbejde dog ikke den store betydning. JPL har ikke været involveret i opdagelsen af koderne, som faktisk har betydet et mindre paradigmeskift i kanalkodningen, hvor konkatenerede systemer tidligere har domineret. Ralph Kötter fra MIT afviste, at rumindustrien skulle have haft nogen betydning for dette skift:

> The space programs didn't think different than other coding theorists. They also had the concatenated coding schemes consisting of an algebraic code and a convolutional code. (Kötter; 2005, side 26 i Interviews)

JPL er altså først blevet involveret i arbejdet med turbokoderne efter at disse koder er slået igennem, og det synes ikke sandsynligt, at JPLs arbejde har influeret udviklingen af kanalkodning.

For LDPC-kodernes vedkommende er genopdagelsen af koderne først sket senere i 1990'erne blandt andet med MacKay & Neal (1996), og LDPC-koder har derfor ikke fundet anvendelse i rummet endnu. Jon Hamkins fortalte dog, at JPL også arbejder på standardiseringen af disse koder, der ligesom turbokoderne skal i spil i fremtidige missioner (Hamkins; 2005, side 48 i Interviews). Hamkins fortalte videre:

> For the ones [missions] where power is really a critical thing, which it often is, they're going to be using either turbo codes or LDPC-codes. Each has it's niche of where it's applicable. (Hamkins; 2005, side 50 i Interviews)

På sigt kommer JPLs arbejde med kanalkodning altså til at få en stor betydning for *anvendelser i rummet*, fordi de koder, der er i færd med at blive standardiseret, vil blive implementeret i hardware og derfor af økonomiske årsager blive svære at fortrænge. De tidligere omtalte ringe muligheder for at opnå forbedringer vil yderligere betyde, at det vil være svært at retfærdiggøre udskiftninger, hvorfor man godt kan regne med at turbo- og LDPC-koderne kommer til at dominere i mange år ud i fremtiden.

For kanalkodning forstået som en matematisk teori har JPLs arbejde med turbo- og LDPC-koder ikke haft stor betydning. Ralph Kötter fortalte om kanalkodningens udvikling:

> If Gallager had invented his codes perhaps not so early in coding theory... I think the problem was what to do with computers. Gallager's codes had the same potential as turbo codes; he could have produced the same curves. I don't think he was guided by anything in space missions. He had coding schemes and they were coding schemes. Maybe if he had the computers to evaluate them... he could have produced the same curves and coding theory would have taken quite a different path, it would never have gone down that byroute as much as it did. I don't think that the space missions have been the driving factor here. And I doubt very much that the French guys had the space missions in mind when they invented the turbo codes. (Kötter; 2005, side 26 i Interviews)

Ralph Kötter mente i øvrigt, at udviklingen inden for kanalkodning i langt højere grad har været drevet af kodning til eksempelvis harddiske og CD-teknologi, altså industrielle problemstillinger (Kötter; 2005, side 27 i Interviews). Den seneste udvikling inden for kanalkodning er altså i meget ringe grad blevet påvirket af arbejde foretaget på JPL. Mars-missionerne inden MRO har slet ikke gjort brug af turbokoder, og LDPC-koderne er slet ikke nået ind i missionerne endnu. JPLs arbejde med standardisering af koderne vil på lang sigt få betydning for kodernes anvendelse i rummet, men for den matematiske teori kanalkodning vil arbejdet ikke betyde noget særligt.

I forhold til denne diskussion er det interessant at bemærke, at stort set alle anvendelser af kanalkodning er relateret til binære koder, og at anvendt kanalkodning derfor i realiteten kun udgør en lille del af den generelle kodningsteori. Den drivkraft for kanalkodningen, som Ralph Kötter nævner, er faktisk også kun relateret til anvendelser og dermed binære

koder. Fremstillingen af kanalkodning, som er givet i INDFØRELSEN, viser, at kanalkodning kan indføres i en mere generel matematisk ramme.

7.5 Generelt om matematikken i de aktuelle Mars-missioner

Til besvarelsen af problemformuleringens spørgsmål 3 skal vi forsøge at generalisere den foregående diskussion og se på *matematikken* i de aktuelle Mars-missioner.

7.5.1 Matematik udviklet til lejligheden

For kilde- og kanalkodningens vedkommende har vi set, at det arbejde, der foregår fra mission til mission, sjældent involverer udvikling af matematik, som ikke findes i forvejen.

Mars-missionerne har forskellige formål og introducerer derfor forskellige problemstillinger, som skal løses på forskellig vis. Det arbejde, der foretages på JPL, er ikke koncentreret om at finde nye og bedre løsninger på problemerne. I stedet vælges løsninger, som allerede er udviklede og afprøvede. Dette gælder både for kilde- og kanalkodning. Man kan umiddelbart indvende, at ICER er udviklet af JPL til lejligheden, og at algoritmen derfor adskiller sig væsentligt fra eksempelvis Reed-Solomon koden, som er blevet udviklet for mange år siden. Vi opfatter imidlertid ikke ICER som matematik, men som en *anvendelse af matematik*. Hvis man ser på ICERs bestanddele, får man det billede, vi beskriver. Matematikken i algoritmen (wavelets, Golomb-koder og så videre) var allerede udviklet, da ICER skulle stykkes sammen. JPLs udviklere behøvede 'bare' at vælge de teknikker, som de fandt bedst egnede, og implementere disse i software.

Der er en lang række faktorer, som vi mener, gør det rimeligt at generalisere ovenstående billede af kilde- og kanalkodning til *matematikken* i de aktuelle Mars-missioner generelt.

En meget væsentlig faktor er, at matematik udviklet til lejligheden og implementeret i teknologi introducerer en række *usikkerhedsmomenter*. Der er for det første usikkerheden om, hvornår arbejdet gøres færdigt. Matematisk grundforskning er besværligt at passe ind under fastlagte deadlines, fordi det ofte er svært at vurdere, hvor lang tid det tager at løse et givet problem. For et sted som JPL, hvor deadlines er så vigtige og opsendelsesvinduerne for missionerne så små, kan forsinkelser ikke tolereres. Et andet usikkerhedsmoment har at gøre med afprøvning. Selvom man afprøver et nyt stykke teknologi med dertil indlejret matematik inden missionens opsendelsesfase, så ved man ikke, om alting fungerer korrekt, før missionen er i gang. Vores samtaler med Jon Hamkins bekræfter dette generelt. Matematik og teknologi, som har været om bord på en tidligere mission, er ikke nær så usikker og er derfor et mere attraktivt valg.

En anden væsentlig faktor er økonomien. 'Low-cost' missionerne er nødt til at genbruge fra mission til mission, fordi hele rumsystemet, som Folkner udtrykker det, er »én stor cost-benefit analyse«. JPLs samlede budget til udvikling af roverne har været 800 millioner dollars[5], så der er

5 Se http://pr.caltech.edu/periodicals/CaltechNews/articles/v37/flights.html

en hel størrelsesorden til forskel fra Vikings budget på 8 milliarder dollars. Den økonomiske faktor er derfor meget vigtig.

En tredje faktor er et mere menneskeligt aspekt. De personer, der kommer til JPL, kommer der for at gennemføre rummissioner og ikke for at udvikle ny matematik (se kapitel 2). Motivationen bag det daglige arbejde er altså ikke at bedrive matematisk grundforskning. Dertil kommer krav fra ledelsens side om at vælge sikre, billige løsninger. Vi mener, at disse faktorer er med til at forstærke billedet af JPL som en matematik-forbruger frem for en matematik-udvikler.

I kapitel 2 har vi nævnt nogle konkrete matematiske problemstillinger, som optræder i MER-missionen. Disse problemstillinger mener vi ligeledes understøtter ovenstående. Både Kalmanfilteret og 'Lost in Space' problemet har været kendt og behandlet siden 1960'erne. At konstruere software, som implementerer Kalmanfilteret, og få den software til at fungere om bord på et rumfartøj er en kæmpe ingeniørmæssig udfordring, men det er ikke en *matematisk* udfordring. Det samme gælder for 'Lost in Space'-problemet.

Kontrolteorien i missionen er endnu et af de eksempler, som vi har nævnt i kapitel 2. Vores samtaler med Jacob Matijevic og William Folkner antyder, at der ikke er udviklet noget kontrolteori specielt til MER-missionen. William Folkner antyder endda (se citatet på side 18), at kontrolteorien snarere er på vej ud af missionen end ind i den.

Hohmann-banekurven har været kendt siden 1925 og er den banekurve, der altid benyttes til Mars-missionerne. Banekurven er et godt eksempel på matematik, som bare skal pilles ned fra hylden, når der er brug for den. At de *behov*, der er for matematik i missionerne, i en vis grad allerede er dækket, betyder naturligvis også, at der ikke foretages nyudvikling. Som følgende kommentar fra Folkner illustrerer, har rover-missionernes matematiske behov i øvrigt ikke ændret sig særligt meget:

> It is very similar. In terms of design of the trajectory, doing the computations for finding out what time of day it is and where to steer the antenna, which direction to put the arm in, it's very similar. I don't think the mathematics has changed very much. (Folkner; 2005b, side 76 i INDFØRELSEN)

Umiddelbart ser det altså ikke ud til, at der er meget at komme efter med hensyn til matematik udviklet specifikt til MER. Der er heller ikke noget, der tyder på, at genbrugen af matematikken skulle være specielt for MER, men nærmere typisk for de aktuelle missioner. Vi spurgte Folkner, om det forhold, at roverne om bord på Pathfinder og MER var mobile enheder, havde introduceret ny matematik i missionerne:

> I don't consider that a big mathematical difference. It is a huge technological difference, but not a big mathematical – I don't think there is a different class of mathematics invented to make it happen. (Folkner; 2005b, side 76 i INDFØRELSEN)

Det billede, vi har tegnet af JPL og Mars-missionerne, viser altså en institution, som i meget høj grad anvender matematik, men som samtidig i meget ringe grad selv udvikler denne.

7.5.2 Den skjulte matematik

I kapitel 2 har vi forsøgt at give et generelt billede af JPLs mennesker, af de matematiske problemstillinger, der optræder i MER, samt af hvordan disse typisk løses på JPL. Dette billede vil vi vende tilbage til her for at diskutere skjultheden af matematikken i Mars-missionerne.

Vores oprindelige opfattelse af den i indledningen omtalte skjulthed var, at når man betragtede MER udefra, som medierne gør, så var matematikken ikke til at få øje på. Vi havde måske lidt naivt forestillet os, at når man kørte ind på JPLs område, så var matematikken i missionen pludseligt meget mere synlig. Vores empiriske arbejde har imidlertid vist, at skjultheden ikke kun er et spørgsmål om at betragte missionen udefra eller indefra.

Skjultheden forekommer på flere niveauer. For den almindelige offentlighed er matematikken skjult i en masse teknologi, hvis virkemåde sjældent diskuteres i det offentlige rum. Denne skjulthed er muligvis betinget dels af mediernes angst for at frastøde læsere eller seere ved at formidle 'kedelig' viden om missionerne og dels af de besværligheder, der normalt er forbundet med at formidle abstrakt matematik til ikke indviede. Denne skjulthed er ikke så interessant at diskutere for os, fordi den er så åbenlys.

Mere interessant er det, at matematikken i et vist omfang også er skjult inden for JPLs mure. Som vi har omtalt i kapitel 2 er en stor del af den matematik, der udgør grundlaget for at løse missionernes mange problemstillinger, indlejret og dermed skjult i software-pakker. Disse software-pakker er i vid udstrækning kommercielle pakker, der er udviklet uden for JPL. For ansatte på JPL er det vigtigt at vide, hvordan pakkerne benyttes på korrekt vis, og det kræver sandsynligvis et vist matematisk kendskab, men den specifikke matematik, der indgår i pakkerne, er det ikke for JPL-folkene nødvendigt at forstå i detaljer.

Det er altså ikke urimeligt at sige, at matematikken i en vis forstand er udliciteret, og at implementeringen i pakker medvirker til at usynliggøre matematikken for JPLs ansatte. Som samtalen mellem William Folkner og Miguel san Martin på side 12 illustrerer, betyder dét at matematikken er usynlig, at den også i et vist omfang fortrænges fra bevidstheden.

Et andet forhold i denne sammenhæng, som er værd at nævne, er at JPLs projekter er så store, at sondringen mellem at være indenfor og udenfor til dels er meningsløs. En ansat, som arbejder med en særlig del af missionen, har naturligvis en *overordnet* opfattelse af, hvilken matematik, der er involveret i andre dele af missionen, men som nedenstående uddrag af en af vores samtaler med William Folkner viser, er detaljerne i høj grad skjulte. Samtalens indledende bemærkning er egentlig en vittighed fra Folkners side, som skal illustrere, at det er nemmere at afdække den matematik, der ikke finder anvendelse i missionerne end omvendt.

– We don't do any map theory for instance. Four color map problem. You know that? We don't do any of that. We don't really use any abstract algebra, group theory and that sort.

Except in the channel coding.

– They use abstract algebra and group theory in that?

The Reed-Solomon codes are based on Galois Fields.

– That's news to me. I didn't know that, all right, interesting. (san Martin & Folkner; 2005, side 62 i INTERVIEWS)

Det er værd at bemærke, at Folkner siger »*They* use abstract algebra and group theory in that?« Dette antyder, at samhørigheden mellem de forskellige afdelinger på JPL ikke er så stor. Hvis man slår op på side 57 i INTERVIEWS er det i øvrigt tydeligt, at Folkner og san Martin møder hinanden for første gang under vores interview.

Pointen med disse observationer er at anskueliggøre, at matematikken i MER også er skjult fra afdeling til afdeling. De ansatte kender oftest kun dem/det, de arbejder med til daglig. Kendskabet til personer og matematik uden for det daglige arbejde er begrænset.

Den sidste form for skjulthed, vi vil nævne, er den, der er relateret til klassificering, altså det bevidste forsøg på at skjule. Generelt var åbenheden på JPL stor, men vi oplevede et par gange at få svar, som var betinget af klassificering (se side 18 i kapitel 2). Motiverne bag klassificeringen har vi ikke undersøgt nærmere. Ikke desto mindre er klassificeringen med til at skjule matematikken for offentligheden og for os.

7.5.3 Spørgsmålet om grundforskning

Den diskussion, vi har ført i dette kapitel, mener vi i et vist omfang sætter os i stand til at afvise, at institutionen JPL skulle bidrage til matematisk grundforskning. Der er dog et par årsager, som gør, at vi ikke kan give et absolut svar.

For det første er der den form for skjulthed, som er relateret til klassificering. Klassificering foretages højest sandsynligt af hensyn til eventuel patentering og er nok hovedsageligt møntet på software, men det er også muligt, at der er nyudviklet matematik skjult her. Den mulige klassificerede matematik er svær for os at vurdere. Umiddelbart synes det realistisk at antage, at omfanget af 'skjult grundforskning' i hvert fald må være begrænset.

Et andet forhold, som gør diskussionen kompleks, er at JPL som institution hører under CalTech. Vi mener ikke, at bidrag til grundforskning udelukkende kan afgøres på baggrund af, hvem der financierer hvilket arbejde. For at afgøre spørgsmålet er man også nødt til at undersøge relationer mellem folkene på JPL og folkene på CalTech. Det synes som udgangspunkt plausibelt, at den matematiske grundforskning foretages på CalTech, og at arbejdet på JPL handler om anvendelserne, men det synes også plausibelt, at der må være en vis påvirkning institutionerne imellem. JPL rekrutterer mange forskere fra CalTech og for MERs vedkommende er fem af missionens ti ledere tidligere CalTech-studerende[6].

Vi kan altså ikke totalt afvise, at JPL bidrager til matematisk grundforskning, men vi kan sige, at der er mange faktorer, der peger på, at det ikke er tilfældet.

6 Se http://pr.caltech.edu/periodicals/CaltechNews/articles/v37/flights.html

7.6 Afrunding

Vores diskussion skulle gerne have gjort det klart, at både kilde- og kanalkodning er vigtige for MER-missionen. To af missionens succeskriterier er direkte koblet til rovernes overfladekørsel, som afhænger af evnen til hurtigt at transmittere navigationsbilleder til Jorden. Kildekodning forøger denne evne væsentligt.

Brugen af kildekodning i en mission betyder, at vigtigheden af kanalkodning opgraderes væsentligt. Desuden er kanalkodning på grund af rovernes energibegrænsninger en forudsætning for overhovedet at kunne kommunikere mellem Mars og Jorden. Samlet giver kilde- og kanalkodning en forøgelse af transmissionsraten. Denne forøgelse betyder, at massen til kommunikationsudstyret kan reduceres og giver dermed en økonomisk gevinst i opsendelsesfasen.

Vores undersøgelse peger på, at MER-missionen og de andre aktuelle Mars-missioner i høj grad er forbrugere af kilde- og kanalkodning, men at missionerne giver meget lidt tilbage i form af grundforskning til teorierne. Der er flere årsager til dette forhold.

For det første er Mars-missionerne prestigeprojekter med dertil hørende bevågenhed fra alle kanter, hvilket udmønter sig i en modvilje mod at introducere usikkerhedsmomenter i form af nyudviklet matematik. For det andet er de aktuelle Mars-missioner såkaldte 'low-cost' missioner, som er nødt til at genbruge allerede eksisterende løsninger på de problemer, der altid skal løses – herunder kommunikation over store afstande med meget lidt energi til rådighed. Kanalkoderne er desuden implementeret i kostbar hardware og underlagt CCSDS-standarder, hvilket gør udskiftning besværlig.

Vores interviews antyder, at forholdene inden for kilde- og kanalkodning i høj grad gør sig gældende i mange andre af missionens områder. Matematikken synes altså at være noget, der tages ned fra hylden efter behov og er i øvrigt ofte skjult for JPLs ansatte. Vores vurdering af de aktuelle Mars-missioner og af institutionen JPL er, at bidraget til matematisk grundforskning er yderst begrænset.

Det billede af Mars-missionen, som vi står tilbage med nu efter vores undersøgelse, er markant forskelligt fra det vi havde, da vi påbegyndte specialet. Vi havde regnet med, at missionernes ekstreme natur ville stille nogle ekstreme krav til matematikken, og at den matematiske udvikling fra mission til mission derfor ville være langt større, end det synes at være tilfældet. Jon Hamkins udtalelse på side 96 antyder faktisk det modsatte. De ekstreme krav, der stilles, synes for matematikkens vedkommende at dreje sig om *ekstrem pålidelighed*.

Teknologien i missionen tilhører ikke direkte dette speciales problemfelt. Alligevel vil vi bringe et citat af Jørn Justesen, som viser, at konservatismen ikke udelukkende gælder matematikken i missionen, men at teknologien også i en vis grad er omfattet:

> Der en skæg historie om den dér skildpadde på Mars, som gik fuldstændigt i baglås og måtte rebootes. Så viser det sig, at det er en fejl i DOS, som man aldrig har fået rettet, og det synes jeg da var helt sødt. Jeg troede sgu da ikke, at der fandtes nogen DOS-systemer mere, men det viser lidt om, at teknologien ikke går så hurtigt. Der er ikke Windows eller Linux eller noget

> andet, der er DOS. Og en 8086-processor, som man troede var på museum, men som er deroppe i en eller anden særlig hård udgave. (Justesen & Høholdt; 2004b, side 5 i INTERVIEWS)

8 Konklusion

De overordnede videnskabelige mål for MER-missionen er stærkt afhængige af de to rovere Spirit og Opportunitys evne til at bevæge sig rundt på Mars' overflade. Denne evne afhænger i høj grad af kilde- og kanalkodning, og de to teorier indtager derfor en væsentlig plads i missionen.

Rovernes mobilitet er i stort omfang afhængige af deres evne til at transmittere billeder til at navigere efter på Jorden. MERs komprimeringsalgoritme ICER sørger for, at billederne kan komprimeres med mere end en faktor ti inden de transmitteres. Komprimeringen, som er særligt hensigtsmæssig for navigationsbilleder, betyder en besparelse af både tid og energi. Den energi, som roverne skal bruge til billedtransmission er en del af et større energiregnskab. Alle arbejdsopgaver skal nemlig dækkes af den mængde energi, roverne får fra deres solpaneler. Denne mængde er yderst begrænset. Begrænsningen af tidsforbruget til transmission betyder mere tid til andre formål og dermed mere mission for pengene.

Kanalkodning er vigtigt for alle missioner med energibegrænsninger og dermed i særdeleshed for MER. Kanalkodning muliggør pålidelig transmission af information, selvom denne information findes i signaler med en meget lav effekt. Pålidelig kommunikation er vigtig både af hensyn til de videnskabelige data, der transmitteres fra Mars, men også af hensyn til beskeder, der sendes fra Jorden til Mars. En besked om at køre til en bestemt lokation kan få katastrofale følger for missionen, hvis den modtages forkert af roveren. Kanalkodningens betydning forstærkes yderligere af brugen af kildekodning, da afkodningen af komprimerede data er mere følsom overfor transmissionsfejl end afkodningen af ikke-komprimerede data.

Samlet udgør kilde- og kanalkodning altså grundstenene i MERs kommunikationssystem og er derfor særdeles vigtige for missionen. Anvendelsen af kilde- og kanalkodning betyder desuden, at man kan reducere størrelsen af de antenner, der er om bord på roverne. Denne reduktion betyder en massebesparelse, som mere eller mindre direkte kan omregnes til en økonomisk gevinst. Dette gælder MER og alle andre rover-missioner i øvrigt.

De aktuelle Mars-missioner udgør en vigtig anvendelse for både kilde- og kanalkodning, men nogen teoretisk påvirkning af teorierne synes der ikke at komme fra missionerne. For kanalkodningens vedkommende er de to vigtigste udviklinger i nyere tid opdagelsen af turbokoderne og genopdagelsen af LDPC-koderne i begyndelsen henholdsvis midten af 1990'erne. Arbejdet med MER-missionen og de tidligere Mars-missioner har ikke påvirket udviklingen af disse koder. Missionerne kan derfor ikke siges at have påvirket den seneste udvikling inden for kanalkodning. På JPL arbejdes der dog på at standardisere begge kodetyper til fremtidige anvendelser i rummet.

For billedkomprimering, som er den del af kildekodning, vi interesserer os for, er den vigtigste udvikling i nyere tid wavelets. MERs algoritme ICER er baseret på wavelets, men algoritmen er stykket sammen af bestanddele, som er udviklet uden for JPL, og som i øvrigt i mange tilfælde er gamle teknikker. Dette gør det usandsynligt, at algoritmen skulle have haft – eller få – nogen indflydelse på udviklingen inden for kildekodning. Wavelets har ikke været om bord på andre Mars-missioner.

Der er ingen grund til at tro, at udviklingen inden for kilde- og kanalkodning skulle været blevet påvirket af de aktuelle Mars-missioner. Grundene til, at JPL ikke har været involveret i denne udvikling, er mange. Krav om pålidelighed og lave omkostninger til kommunikationssystemerne er uden tvivl betydelige årsager.

Kilde- og kanalkodning synes at være eksemplariske for matematikken i missionerne. De eksempler, som vi har præsenteret, tyder på, at matematikken som regel 'hentes ned fra hylden', når den skal i brug, og at der på JPL generelt ikke udvikles matematik til lejligheden. De aktuelle missioner er alle såkaldte 'low-cost' missioner, og kravet om lave omkostninger er derfor en afgørende faktor for mange af de valg, der skal træffes. Nyudvikling af matematik til en mission introducerer for mange usikkerhedsmomenter, som i høj grad er relateret til missionens økonomi. Derfor foretrækkes, at missionernes problemstillinger imødekommes ved genbrug af tidligere afprøvede løsninger.

Generelt er den matematik, der anvendes i missionerne skjult. Skjultheden forekommer på flere niveauer. Først og fremmest er matematikken skjult for den brede offentlighed, som udelukkende ser teknologien. Denne skjulthed var en af vores motivationer for at arbejde med specialets problemstilling. Matematikken er imidlertid også skjult på en række andre områder, som er mere overraskende. JPLs brug af kommercielle softwarepakker, i hvilke store mængder matematik er indlejret, betyder, at de matematiske detaljer i et vist omfang er skjult for forskerne ved JPL. Institutionens størrelse og missionernes videnskabelige omfang bevirker derudover, at matematikken er skjult afdelingerne og de ansatte imellem. Forskerne på JPL har kun et overordnet billede af den matematik, der arbejdes med i de afdelinger, de ikke selv er en del af. Tilsidst skal det nævnes, at der muligvis også er tiltag på JPL med det specifikke formål at skjule matematikken. Klassificering af dele af MER-missionens videnskab har i hvert fald været med til at skjule matematikken for os.

Den samlede karakteristik, vi har givet af matematikken i missionen, antyder kraftigt, at JPL i meget høj grad er en *forbruger* af allerede udviklet og etableret matematik og kun i ringe grad selv bedriver matematisk grundforskning. De arbejdsforhold, der er på JPL, peger i samme retning. En fuldstændig afvisning af bidrag til matematisk grundforskning kan vi dog ikke give. JPLs stærke bånd til CalTech betyder sandsynligvis, at der er en vis påvirkning institutionerne imellem og dermed, at JPL muligvis indirekte bidrager til matematisk grundforskning.

9 Efterskrift

> Science is not stagnating. It's not. (Folkner; 2005a, side 68 i Interviews)

I dette kapitel vil vi afrunde specialet samt nævne et par problemstillinger, som vi meget sent i projektforløbet valgte fra. Dette efterskrift kan således tjene som inspiration for studerende, der savner inspiration til projekter. Vores interviews indeholder en del information, som er relevante for de problemstillinger, der i løbet af specialeperioden blev fravalgt.

9.1 Stagnationshypotesen

På det tidspunkt, vi foretog vores interviews på JPL, handlede en del af specialet om at undersøge en stagnationshypotese, som vi her kort skal redegøre for.

Man kan vælge at kategorisere udvikling inden for en teori eller et fag som *ekspanderende* eller *konsoliderende*. Udvikling, som tilfører nye begreber og problemstillinger, kaldes ekspanderende, mens udvikling som giver en ny eller bedre forståelse af allerede eksisterende begreber kaldes konsoliderende. Stagnationshypotesen, som berøres i Booß-Bavnbek (2005) og i Galar (2005), handler om forholdet mellem ekspansion og konsolidering i naturvidenskaberne i det tyvende århundrede.

Ifølge hypotesen har fysikken og biologien i den første halvdel af det tyvende århundrede oplevet en periode med store konsolideringer i form af relativitetsteorien, kvantemekanikken og for biologiens vedkommende opdagelsen af, at DNA-molekyler har form som en dobbelt-helix. Siden midten af århundredet har de to fag imidlertid ikke oplevet konsolidering, men udelukkende ekspanderende udvikling. For kemiens vedkommende tegner sig et lignende billede. De store konsolideringer fandt sted under den industrielle revolution i 1800-tallet, og siden da har faget 'kun' oplevet ekspanderende udvikling. *Stagnationen* handler om den mangel på konsolidering, der har været i naturvidenskaberne i nyere tid.

Vores undersøgelse af stagnationshypotesen skulle være et forsøg på at afklare, om faget matematik passer ind i hypotesen, eller om det forholder sig anderledes med matematikken. Vi blev hurtigt klar over, at der er meget, der taler for, at det rent faktisk forholder sig anderledes med matematikken, og vores speciale kom derfor i et vist omfang til at handle om at understøtte hypotesen med udgangspunkt i kilde- og kanalkodning.

Oprindeligt havde vi en forventning om, at der grundet missionens ekstreme natur ville være en masse nyudviklet matematik i spil i MER, og vi syntes derfor dengang, at missionen udgjorde et godt udgangspunkt for en undersøgelse af hypotesen. Som specialet viser, er dette imidlertid ikke tilfældet, og vi kan derfor ikke komme med nogen konklusioner i den

retning, men vi vil alligevel nævne et forhold, som vi opdagede under vores interviews, og som forekommer os at være interessant.

Vi talte om »stagnation of science« med de forskellige forskere ved MIT og JPL og diskuterede hypotesen lidt mere bredt med dem. På MIT tilsluttede Gilbert Strang og Ralph Kötter sig synspunktet om, at udviklingen inden for kilde- og kanalkodning i de sidste årtier er begyndt at sagtne farten, mens alle på JPL var direkte afvisende overfor tanken om en stagnation i de naturvidenskabelige discipliner. Dette er illustreret i kapitlets indledende citat.

Vores bedste bud på en forklaring af de stærkt divergerende synspunkter for de to institutioner er, at MIT-folkene til dagligt arbejder mest med rene matematiske problemstillinger, mens folkene ved JPL i højere grad beskæftiger sig med praktisk anvendt matematik. For en ren matematiker betyder tilkomsten af endnu en kode af den ene eller den anden art ikke det store, med mindre den giver et helt nyt billede af teorien. For en anvendt matematiker er det imidlertid muligt, at den nye kode, selvom den ikke er fundamentalt anderledes end andre koder, alligevel løser et *praktisk problem*, og på den måde ændrer hverdagen for de folk, som arbejder med netop dette problem. Vores forklaring er naturligvis ikke så nuanceret, og vi så derfor gerne en videre undersøgelse af årsagerne til de divergerende synspunkter.

9.2 Den videnskabshistoriske dimension

Udover at have fravalgt stagnationshypotesen fra specialets problemstilling har vi også på et sent tidspunkt fravalgt endnu et aspekt, nemlig en videnskabshistorisk tilgang til kilde- og kanalkodningen. Vores undersøgelse involverer teorierne her og nu, men flere ting peger i retning af, at en historisk undersøgelse af samspillet mellem rummissionerne og kilde- og kanalkodningen kunne være interessant. Vi har i denne rapport beskrevet rummissionerne udelukkende som forbrugere af kilde- og kanalkodning, som allerede har været udviklet, når det skulle bruges. Vi har endvidere afvist, at de aktuelle rummissioner på nogen måde skulle have været styrende for udviklingen af kilde- og kanalkodningen. Sådan ser det imidlertid ikke ud til altid at have været.

Shannons artikel fra 1948 banede i en vis forstand vejen for opdagelsen af kanalkoder med de egenskaber, som blev lovet i artiklen. I det årti, der fulgte, blev der derfor brugt mange kræfter på at opdage og udvikle koder. Da rumindustrien i tresserne begyndte at efterspørge kanalkoder, fik man i kraft af deep-space kanalens pæne matematiske egenskaber således en fantastisk afprøvningsmulighed for de udviklede koder.

Afprøvning af teorien var af stor vigtighed for kanalkodningen, men der var et andet forhold, som muligvis havde en større betydning. I kanalkodningens tidlige år blev koderne udviklet som rene matematiske spekulationer, fordi der ingen anvendelser var til at aftage koderne. Kanalkodning blev derfor af mange betragtet som en teoretisk finurlighed, der ikke havde nogen praktiske konsekvenser (Forney, Jr.; 1970a, side 47). Anvendelserne i rummissionerne, som på dette tidspunkt var omgivet af endnu større prestige end i dag, udgjorde derfor en vigtig retfærdiggørelse af forskningen inden for kanalkodning. Således hedder det i Massey (1992)

om kanalkodningen og deep-space kanalen i Pioneer 9 missionen (1968) til Solen:

> It is no exaggeration to say that the Pioneer 9 mission provided communications engineers with the first incontrovertible demonstration of the practical utility of channel coding techniques and thereby paved the way for the successful application of coding to many other channels. (Massey; 1992, side 1)

For kanalkodningens vedkommende ser det altså ud som om, at båndene til rumindustrien *har* været stærkere, end det er tilfældet i dag. Andrew Viterbi var eksempelvis tilknyttet JPL som konsulent, da han udviklede sin afkodningsalgoritme til foldningskoder. I et interview[1] fra 1999 erklærede han i øvrigt, at han *havde* deep-space kanalen i tankerne, da han udviklede algoritmen. Viterbi troede dengang, at algoritmen var for kompleks til implementering, men rent faktisk blev der på JPL foretaget arbejde, som muliggjorde anvendelser af algoritmen.

Også kildekodning har tidligere modtaget interessante bidrag fra rumindustrien. Til Voyager-missionerne i 1977 udvikledes på JPL af Robert Rice en ny algoritme til komprimering af billeder. Kodningsproceduren, der i dag kendes som Rice-kodning, må siges at have været et vigtigt bidrag til kildekodningen. JPL støttede i øvrigt også Golombs arbejde, da han udviklede Golomb-koderne. (Sayood; 2000, side 64-66), (Golomb; 1966, fodnote side 399)

Introduktionen af Rice-kodningen betød grundet dekomprimeringens følsomhed overfor bitfejl at en BER på 10^{-6} i stedet for den gængse på $5 \cdot 10^{-3}$ var påkrævet. Voyagers konkatenerede kodningssystem, der som ydre kode havde $(24, 12)$ Golay-koden, rakte ikke til en så lille BER, hvorfor Reed-Solomon koderne i stedet blev taget i brug. Det konkatenerede system bestående af den ydre $(255, 223)$-Reed-Solomon kode og den indre $(2, 1, 7)$-foldningskode, som anvendes i MER, fandt således sin vej ind i rumindustrien i 1977 på grund af Rices algoritme. Dette illustrerer på eksemplarisk vis samspillet mellem kilde- og kanalkodning i rummissionerne. (McEliece & Swanson; 1994, side 30), (Wicker; 1998, side 2138-2141)

9.3 Afsluttende kommentarer

I dette speciale har vi blandt andet diskuteret i hvilket omfang institutionen JPL bidrager til matematisk grundforskning. Vi vil gerne her gøre opmærksom på, at vores diskussion på ingen måde er *moralsk* ladet. Når vi konkluderer, at meget peger på, at JPL i højere grad er en forbruger af matematik end en bidragyder, så ligger der ikke heri en anklage mod JPL, men udelukkende en konstatering af hvilket arbejde, der bedrives på institutionen.

Vi har i rapporten i høj grad afvist, at JPL skulle bidrage til den nutidige udvikling inden for kilde- og kanalkodning. For kanalkodningens vedkommende er det spørgsmålet om rummissionerne i fremtiden overhovedet *kan* bidrage til udviklingen. Ralph Kötter fortalte om kanalkodningens fremtid:

1 Se www.ieee.org/organizations/history_center/oral_histories/transcripts/viterbi.html.

> Well, there was the old problem about achieving capacity on channels, the AWGN channel and the binary selective channel – the two old big ones. But all this is done. It is just simply done. The best codes provably achieve up to 0.00-something. For all practical purposes it's done. So there's not even room left for any breakthroughs, that is with respect to this old problem. [...] The old coding theory focusing on achieving capacity on these two channels has served it's purpose and achieved its goal. (Kötter; 2005, side 28 i INTERVIEWS)

JPL opfylder en for videnskab meget vigtig rolle, nemlig rollen som *aftager*. Samtidig er Mars-missionerne meget synlige og med til at popularisere videnskab, hvilket er et bidrag, der ikke må undervurderes. Matematikken i sig selv er jo i en vis forstand *usynlig*.

Det arbejde, der er foretaget i forbindelse med dette speciale, mener vi ikke er trivielt. Matematikkens skjulte natur gør undersøgelser som denne interessante. Som specialets afsluttende kommentar bringer vi et citat af »the famous« Gilbert Strang. Citatet viser, at matematikken ikke kun er skjult for den almindelige befolkning, men i høj grad også for matematikere verden over.

> ... because I'm one of the common people when it comes to the Mars Mission, I couldn't say what was used that was essential. But I bet there was essential math. Probably solving ordinary differential equations with great accuracy, prediction, estimation. Kalman Filters would probably enter. So there is a whole world there. That would be a good question to ask: 'What was the math?' (Strang; 2005, side 32 i INTERVIEWS)

Litteratur

Adams, M. D. & Kossentini, F. (2000). Reversible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis, *IEEE Transactions on Image Processing* **9**: 1010–1024.

Berger, T. (1971). *Rate Distortion Theory – A Mathematical Basis for Data Compression*, Information and System Science Series, Prentice-Hall, Inc., Englewood, New Jersey.

Berlekamp, E. R. (1974). Key Papers in The Development of Coding Theory, IEEE Press, New York, pp. 1–6, 67–69, 107–108, 157–158, 233–237. Introductory Comments.

Biggs, N. L. (1989). *Discrete Mathematics*, Revised edn, Oxford Science Publications, Oxford.

Blahut, R. E. (2003). *Algebraic Codes for Data Transmission*, Cambridge University Press, Cambridge.

Booß-Bavnbek, B. (2005). Was bedeutet die Mathematisierung von Krieg?, *FIfF Kommunikation – Forum InformatikerInnen für Frieden und gesellschaftliche Verantwortung e.V.* **2**: 27–34.

Calderbank, A. R. (1998). The Art of Signaling: Fifty Years of Coding Theory, *IEEE Transactions on Information Theory* **44**(6): 2561–2595.

CCSDS (1997). Recommendation for Space Data System Standards – Lossless Data Compression.
URL: *http://public.ccsds.org/publications/archive/121x0b1.pdf*

CCSDS (2002). Recommendation for Space Data System Standards – Telemetry Channel Coding.
URL: *http://www.ccsds.org/documents/101x0b6.pdf*

Costello, Jr., D. J., Hagenauer, J., Imai, H. & Wicker, S. B. (1998). Applications of Error-Control Codes, *IEEE Transactions on Information Theory* **44**(6): 2531–2560.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*, Wiely Series in Telecommunications, John Wiley & Sons, Inc., New York.

Daubechies, I. (1992). *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series In Applied Mathematics, Capital City Press, Montpelier, Vermont.

Davis, P. J. (2004). A Letter to Christina of Denmark, *EMS* pp. 21–24.

Davis, P. J. (2005). Interview med professor emeritus Philip J. Davis den 6. marts 2005. Foretaget på Brown University, Providence.

Elias, P. (1954). Error-Free Coding, *IRE Transactions on Information Theory* **4**: 29–37.

Fernández-Cara, E. & Zuazua, E. (2003). Control Theory: History, Mathematical Achievements and Perspectives, *Boletin SEMA* **26**: 79–140.
URL: *http://me.wpi.edu/IEEETC_DPS/ARTICLES/DivSEMA.pdf*

Folkner, W. (2005a). Interview med doktor William Folkner den 16. marts 2005. Foretaget på JPL, Pasadena.

Folkner, W. (2005b). Interview med doktor William Folkner den 17. marts 2005. Foretaget på JPL, Pasadena.

Forchhammer, S. (2004). Interview med lektor Søren Forchhammer den 31. januar 2005. Foretaget på DTU, Lyngby.

Forney, Jr., G. D. (1966). Concatenated Codes, *MIT Press* .

Forney, Jr., G. D. (1970a). Coding and its application in space communication, *IEEE Spectrum* **7**: 47–58.

Forney, Jr., G. D. (1970b). Convolutional Codes I: Algebraic Structure, *IEEE Transactions on Information Theory* **IT-16**(6): 720–738.

Forney, Jr., G. D. (1974a). Convolutional Codes II: Maximum-Likelihood Decoding, *Information and Control* **25**: 222–266.

Forney, Jr., G. D. (1974b). Convolutional Codes III: Sequential Decoding, *Information and Control* **25**: 267–297.

Fraleigh, J. B. (1999). *A First Course in Abstract Algebra*, Sixth edn, Addison-Wesley, Reading, Massachusetts.

Frazier, M. W. (1999). *An Introduction to Wavelets through Linear Algebra*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, Berlin, Heidelberg.

Galar, R. (2005). *Adaptive vs Managerial Approach to S&T Policy*, Nowy Sacz.

Gallager, R. G. (1968). *Information Theory and Reliable Communication*, John Wiley & Sons, Inc., New York.

Gallager, R. G. (1978). Variations on a Theme by Huffman, *IEEE Transactions on Information Theory* **IT-24**(6): 668–674.

Gallager, R. G. & van Voorhis, D. C. (1978). Optimal Source Codes for Geometrically Distributed Integer Alphabets, *IEEE Transactions on Information Theory* **IT-21**(6): 228–230.

Golay, M. J. E. (1949). Notes on Digital Coding, *Proceedings of the IRE* **37**: 657.

Golomb, S. W. (1966). Run-Length Encodings, *IEEE Transactions on Information Theory* **IT-12**: 399–401.

Goodrich, M. T. & Tamassia, R. (2002). *Algorithm Design – Foundations, Analysis and Internet Examples*, John Wiley & Sons, Inc., New York.

Hamkins, J. (2005). Interview med doktor Jon Hamkins den 14. marts 2005. Foretaget på JPL, Pasadena.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes, *Bell System Technical Journal* **29**: 147–160.

Høholdt, T. (2004). Interview med docent Tom Høholdt den 26. august 2004. Foretaget på DTU, Lyngby.

Howard, P. G. (1998). Interleaving Entropy Codes, *Proceedings Compression and Complexity of Sequences 1997* pp. 45–55.

Howard, P. G. & Vitter, J. S. (1993). Fast and Efficient Lossless Image Compression, *Proceedings of the IEEE Computer Society/NASA/CESDIS Data Compression Conference* pp. 351–360.

Huffman, D. A. (1952). A Method for the Construction of Minimum-Redundancy Codes, *Proceedings of the IRE* **40**: 1098–1101.

Jacobs, I. M. (1974). Practical Applications of Coding, *IEEE Transactions on Information Theory* **IT-20**: 305–310.

Jensen, A. & la Cour-Harbo, A. (2001). *Ripples in Mathematics*, Springer-Verlag, New York, Berlin, Heidelberg.

JPL (1982). Reed-Solomon Encoders – Conventional vs Berlekamp Architecture.

JPL (2002). Mars Exploration Rover (MER) Project Mission Plan.

Justesen, J. & Høholdt, T. (2004a). *A Course In Error-Correcting Codes*, EMS Textbooks in Mathematics, European Mathematical Society, Zürich.

Justesen, J. & Høholdt, T. (2004b). Interview med professor Jørn Justesen og docent Tom Høholdt den 27. september 2004. Foretaget på DTU, Lyngby.

Kiely, A. (2005). Interview med doktor Aaron Kiely den 14. marts 2005. Foretaget på JPL, Pasadena.

Kiely, A. & Klimesh, M. (2001a). A New Entropy Coding Technique for Data Compression, *IPN Progress Report* (42-146): 1–48.
URL: *http://ipnpr.jpl.nasa.gov/progress_report/42-146/146G.pdf*

Kiely, A. & Klimesh, M. (2001b). Memory-Efficient Recursive Interleaved Entropy Coding, *IPN Progress Report* (42-146): 1–14.
URL: *http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-146/146J.pdf*

Kiely, A. & Klimesh, M. (2003). The ICER Progressive Wavelet Image Compressor, *IPN Progress Report* (42-155): 1–46.
URL: *http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-155/155J.pdf*

Kiely, A. & Klimesh, M. (2004). Preliminary Image Compression Results from the Mars Exploration Rovers, *IPN Progress Report* (42-156): 1–8.
URL: *http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-156/156I.pdf*

Kjeldsen, T. H., Pedersen, S. A. & Sonne-Hansen, L. M. (2004). *New Trends in the History and Philosophy of Mathematics*, Vol. 19 of *Studies in Philosophy*, University Press of Southern Denmark, Odense, pp. 11–27. Introduction.

Klimesh, M. (2005). Interview med doktor Matthew Klimesh den 14. marts 2005. Foretaget på JPL, Pasadena.

Kötter, R. (2005). Interview med doktor Ralph Kötter den 7. marts 2005. Foretaget på MIT, Cambridge.

Latour, B. (1987). *Science in Action – How to follow scientists and engineers through society*, Harvard University Press, Cambridge, Massachusetts.

Lin, S. & Costello, Jr., D. J. (1983). *Error Control Codes – Fundamentals and Applications*, Prentice-Hall, Inc., New Jersey.

MacKay, D. J. C. & Neal, R. M. (1996). Near Shannon Limit Performance of Low Density Parity Check Codes, *Electronics Letters* **32**: 1645–1646.
URL: *http://www.inference.phy.cam.ac.uk/mackay/mncEL.pdf*

Maimone, M. W. (2005). Interview med doktor Mark W. Maimone den 17. marts 2005. Foretaget på JPL, Pasadena.

Massey, J. L. (1975). Error Bounds for Tree Codes, Trellis Codes and Convolutional Codes with Encoding and Decoding Procedures, number 216 in *CISM Courses and Lectures*, Springer Verlag, Wien and New York, pp. 1–58.

Massey, J. L. (1992). Deep-Space Communication and Coding: A Marriage Made in Heaven, number 182 in *Lecture Notes in Control and Information Sciences*, Springer, Heidelberg and New York, pp. 1–17.

Matijevic, J. (2005). Interview med doktor Jacob Matijevic den 14. marts 2005. Foretaget på JPL, Pasadena.

McEliece, R. J. (1987). *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic Press, Boston, Dordrecht, Lancaster.

McEliece, R. J. (1998). *The Algebraic Theory of Convolutional Codes*, Elsevier Science B.V., Amsterdam, pp. 1065–1138. Kapitel 12 i Handbook of Coding Theory Volume I.

McEliece, R. J. (2002). *The Theory of Information and Coding*, Vol. 86 of *Encyclopedia of Mathematics and its Applications*, Second edn, Cambridge University Press, Cambridge, United Kingdom.

McEliece, R. J. (2004a). 2004 Shannon Lecture, *IEEE Information Theory Society Newsletter* **54**(4): 1–20.

McEliece, R. J. (2004b). Shannon Lecture XXVI – Are there Turbo Codes on Mars? Slides til foredrag.
URL: *http://www.ee.caltech.edu/EE/Faculty/rjm/papers/ShannonLecture.pdf*

McEliece, R. J. & Swanson, L. (1994). *Reed-Solomon Codes and the Exploration of the Solar System*, IEEE Press, New York, pp. 25–40. Kapitel 3 i Reed-Solomon Codes and Their Applications.

NASA (2003). Mars Exploration Rover Launches, *Press Kit* pp. 1–44.
URL: *http://www.jpl.nasa.gov/news/press_kits/merlaunch.pdf*

Niss, M. (1992). Noget om naturvidenskab, teknologi og samfund i den naturvidenskabelige revolutions periode (ca.: 1550-1750), number 228, IMFUFA, Roskilde, pp. 13–47.

Ono, F., Kino, S., Yoshida, M. & Kimura, T. (1989). Bi-Level Image Coding with MELCODE – Comparison of Block Type Code and Arithmetic Type Code, *Proceedings IEEE Global Telecommunications Conference* pp. 0255–0260.

Rabbani, M. & Jones, P. W. (1991). *Digital Image Compression Techniques*, Vol. 7 of *Tutorial Texts in Optical Engineering*, Spie Optical Engineering Press, Bellingham, Washington.

Reed, I. S. & Solomon, G. (1960). Polynomial Codes over Certain Finite Fields, *Journal of the Society for Industrial and Applied Mathematics* **8**: 300–304.

Reed, I. S. & Solomon, G. (1994). *Reed-Solomon Codes: A Historical Overview*, IEEE Press, New York, pp. 17–24. Kapitel 2 i Reed-Solomon Codes and Their Applications.

Roman, S. (1992). *Coding and Information Theory*, number 134 in *Gradute Texts in Mathematics*, Springer-Verlag, New York, Berlin, Heidelberg.

Roman, S. (1997). *Introduction to Information and Coding Theory*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, Berlin, Heidelberg.

Said, A. & Pearlman, W. A. (1993). Reversible Image Compression via Multiresolution Representation and Predictive Coding, *Proceedings of the SPIE Conference on Visual Communications and Image Processing 1993* **2094**: 664–674.

Said, A. & Pearlman, W. A. (1996). An Image Multiresolution Representation for Lossless and Lossy Compression, *Proceedings of the SPIE Conference on Visual Communications and Image Processing 1993* **9**(5): 1303–1310.

san Martin, M. & Folkner, W. (2005). Interview med doktor Miguel san Martin & doktor William Folkner den 14. marts 2005. Foretaget på JPL, Pasadena.

Sayood, K. (2000). *Introduction to Data Compression*, Second edn, Morgan Kaufmann Publishers, San Francisco, California.

Shannon, C. E. (1948). A Mathematical Theory of Communication I, II, Vol. 27, pp. 379–423; 623–656. Side 5-18 og 19-29 i Key Papers in The Development of Information Theory.

Shannon, C. E. (1949). Communication in the Presence of Noise, Vol. 37, pp. 10–21. Side 30-41 i Key Papers in The Development of Information Theory.

Shannon, C. E. (1959). Coding for a Discrete Source with a Fidelity Criterion, number 4, pp. 142–163. Side 245-266 i Key Papers in The Development of Information Theory.

Slepian, D. (1973). Key Papers in The Development of Information Theory, IEEE Press, New York, pp. 1–4. Introduction.

Solomon, D. (1998). *Data Compression – The Complete Reference*, Springer Verlag, New York.

Sontag, E. D. (1990). *Mathematical Control Theory – Deterministic Finite Dimensional Systems*, number 6 in *Texts in Applied Mathematics*, Springer Verlag, New York.

Strang, G. (2005). Interview med professor Gilbert Strang den 9. marts 2005. Foretaget på MIT, Cambridge.

Strang, G. & Nguyen, T. (1996). *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, Massachusetts.

Thompson, T. M. (1983). *From Error-Correcting Codes through Sphere Packings to Simple Groups*, number 21 in *The Carus Mathematical Monographs*, The Mathematical Association of America.

Vardy, A. (1998). *Trellis Structure of Codes*, Elsevier Science B.V., Amsterdam, pp. 1989–2117. Kapitel 24 i Handbook of Coding Theory Volume II.

Verdú, S. (1998). Fifty Years of Shannon Theory, *IEEE Transactions on Information Theory* **44**(6): 2057–2078.

Vitter, J. S. (1987). Design and Analysis of Dynamic Huffman Codes, *Journal of the Association for Computing Machinery* **34**(4): 825–845.

Walker, J. S. (1999). *A Primer on Wavelets and their Scientific Applications*, Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, Florida.

Weinberger, M. J., Seroussi, G. & Sapiro, G. (1996). LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm, *Proceedings of the IEEE Data Compression Conference* pp. 1–10.
URL: *http://www.hpl.hp.com/loco/dcc96copy.pdf*

Welch, G. & Bishop, G. (2004). An Introduction to the Kalman Filter, (TR-95-041): 1–16.
URL: *http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf*

Wicker, S. B. (1998). *Deep Space Applications*, Elsevier Science B.V., Amsterdam, pp. 2121–2169. Kapitel 25 i Handbook of Coding Theory Volume II.

Wicker, S. B. & Bhargava, V. K. (1994). *An Introduction to Reed-Solomon Codes*, IEEE Press, New York, pp. 1–16. Kapitel 1 i Reed-Solomon Codes and Their Applications.

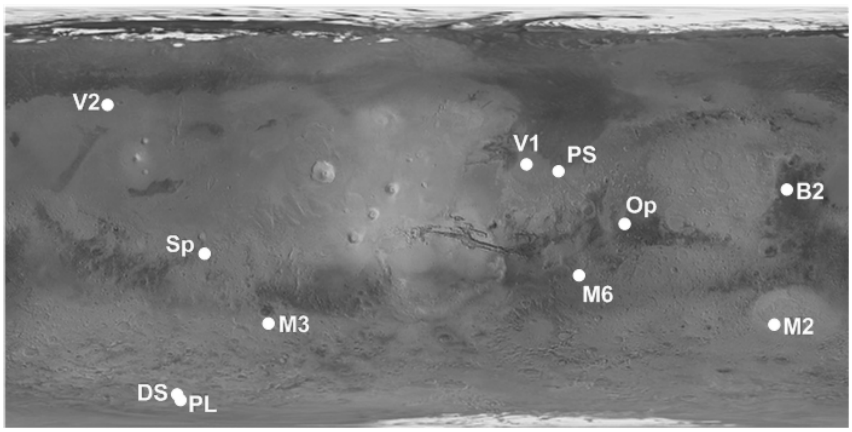
A NASAs gennemførte Mars-missioner

> Well the Viking Lander 1 had a design life of about thirty days, but it worked for six years and it finally stopped working when they incorrectly sent the command to point the antenna down. After that they could not send any more commands. Otherwise it would still be working today. (san Martin & Folkner; 2005, side 58 i Interviews)

Siden 1960 for Sovjets vedkommende og 1964 for USAs har der været adskillige missioner til Mars og ofte med svingende udfald. Missioner til Mars er ikke nogen nem opgave. Cirka $^2/_3$ af samtlige missioner er mislykkedes på den ene eller den anden vis. Tabel A.1 på side 119 giver en oversigt over samtlige Mars-missioner.

Figur A.1 viser landings- og nedstyrtningsstederne på Mars for såvel amerikanske som sovjetiske og europæiske missioner. For en forklaring af forkortelserne på figuren se de efterfølgende kasser på side 118.

Vi vil i dette appendiks give en overordnet præsentation af NASAs tidligere gennemførte Mars-missioner, de igangværende samt de missioner, der er planlagt frem til 2009. Appendikset kan læses som supplerende oplysninger til de missioner, der er nævnt i rapporten. Præsentationen er baseret på informationer fra NASAs hjemmeside samt fra Costello, Jr. et al. (1998), Wicker (1998) og McEliece & Swanson (1994).



Figur A.1 Landings- og nedstyrtningssteder på Mars. http://www.johnstonsarchive.net/astro/marsprobemap.html

Succesfulde landinger på Mars:

M3	Mars 3, USSR, 2. december 1971, operationsdygtig i 90 sekunder.
V1	Viking 1, USA, 20. juli 1976, operation frem til 13. november 1982.
V2	Viking 2, USA, 3. september 1976, operation frem til 12. april 1980.
PS	Mars Pathfinder og Sojourner, USA, 4. juli 1996, rover-operation frem til 20. august 1997 og lander frem til 7. oktober 1997.
Sp	Spirit, USA, 4. januar 2004, fortsat operation.
Op	Opportunity, USA, 25. januar 2004, fortsat operation.

Nedstyrtninger på Mars:

M2	Mars 2, USSR, nedstyrtning 27. november 1971.
M6	Mars 6, USSR, nedstyrtning 12. marts 1974.
PL	Mars Polar Lander, USA, nedstyrtning 3. december 1999.
DS	Deep Space 2, Microprobes 1 og 2, USA, nedstyrtning 3. december 1999.
B2	Beagle 2, ESA, nedstyrtning 25 december 2003.

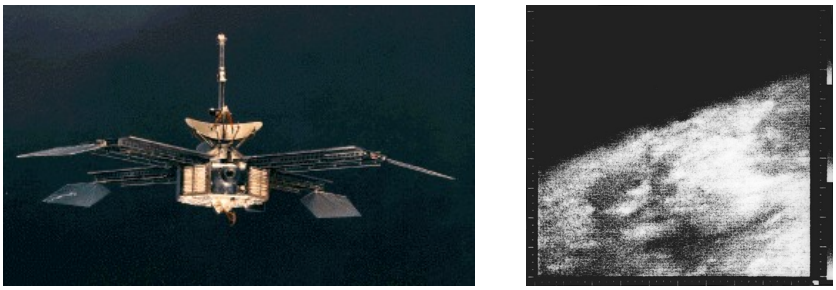
A.1 Tidligere missioner til Mars

NASAs tidligere og gennemførte Mars-missioner omfatter Mariner 4, Mariner 6 og 7, Mariner 9, Viking 1 og 2 og Mars Pathfinder.

A.1.1 Mariner 4

Mariner-missionerne[1] til Mars fandt sted i 1960erne og i begyndelsen af 1970erne. Den 28. november 1964 blev Mariner 4 opsendt og 228 dage senere blev den det første rumfartøj til at tage billeder af Mars og transmittere disse til Jorden.

Den 14. juni 1965 passerede Mariner 4 Mars i en afstand af 9.855 km. Sonden producerede et overfladebillede af Mars sammensat af 200 linier hver bestående af 200 pixels. Overfladebilledet udgjordes af 22 TV-billeder, som blev gemt på fartøjets firespors båndoptager. Tre af billederne gik dog tabt grundet pladsmangel.



Figur A.2 Til venstre: Illustration af Mariner 4. http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1964-077A Til højre: Det første billede af Mars taget af Mariner 4. http://nssdc.gsfc.nasa.gov/imgcat/html/object_page/m04_01d.html

1 Følgende afsnit om Mariner-missionerne 4, 6, 7 og 9 er baseret på http://nssdc.gsfc.nasa.gov/planetary/mars/mariner.html

Tidligere og igangværende missioner til Mars

Navn	Dato og år	Formål og status	Nation
Marsnik 1	10/10 1960	Attempted Mars Flyby (Launch Failure)	USSR
Marsnik 2	14/10 1960	Attempted Mars Flyby (Launch Failure)	USSR
Sputnik 22	24/10 1962	Attempted Mars Flyby	USSR
Mars 1	01/11 1962	Mars Flyby (Contact Lost)	USSR
Sputnik 24	04/11 1962	Attempted Mars Lander	USSR
Mariner 3	05/11 1964	Attempted Mars Flyby	USA
Mariner 4	28/11 1964	Mars Flyby	USA
Zond 2	30/11 1964	Mars Flyby (Contact Lost)	USSR
Zond 3	18/07 1965	Lunar Flyby, Mars Test Vehicle	USSR
Mariner 6	25/02 1969	Mars Flyby	USA
Mariner 7	27/03 1969	Mars Flyby	USA
Mars 1969A	27/03 1969	Attempted Mars Orbiter (Launch Failure)	USSR
Mars 1969B	02/04 1969	Attempted Mars Orbiter (Launch Failure)	USSR
Mariner 8	08/05 1971	Attempted Mars Flyby (Launch Failure)	USA
Cosmos 419	10/05 1971	Attempted Mars Orbiter/Lander	USSR
Mars 2	19/05 1971	Mars Orbiter/ Attempted Lander	USSR
Mars 3	28/05 1971	Mars Orbiter/ Lander	USSR
Mariner 9	30/05 1971	Mars Orbiter	USA
Mars 4	21/07 1973	Mars Flyby (Attempted Mars Orbiter)	USSR
Mars 5	25/07 1973	Mars Orbiter	USSR
Mars 6	05/08 1973	Mars Lander (Contact Lost)	USSR
Mars 7	09/08 1973	Mars Flyby (Attempted Mars Lander)	USSR
Viking 1	20/08 1975	Mars Orbiter and Lander	USA
Viking 2	09/09 1975	Mars Orbiter and Lander	USA
Phobos 1	07/07 1988	Attempted Mars Orbiter/Phobos Landers	USSR
Phobos 2	12/07 1988	Mars Orbiter/Attempted Phobos Landers	USSR
Mars Observer	25/09 1992	Attempted Mars Orbiter (Contact Lost)	USA
Mars Global Surveyor	07/11 1996	Mars Orbiter	USA
Mars 96	16/11 1996	Attempted Mars Orbiter/Landers	Rusland
Mars Pathfinder	04/12 1996	Mars Lander and Rover	USA
Nozomi (Planet-B)	03/07 1998	Mars Orbiter	Japan
Mars Climate Orbiter	11/12 1998	Attempted Mars Orbiter	USA
Mars Polar Lander	03/01 1999	Attempted Mars Lander	USA
Deep Space 2 (DS2)	03/01 1999	Attempted Mars Penetrators	USA
2001 Mars Odyssey	07/04 2001	Mars Orbiter	USA
Mars Express	02/06 2003	Mars Orbiter and Lander	Europa
Spirit (MER-A)	10/06 2003	Mars Rover	USA
Opportunity (MER-B)	07/07 2003	Mars Rover	USA
Mars Reconnaissance Orbiter	10/08 2005	Mars Orbiter (in flight)	USA

Fremtidige planlagte missioner til Mars

Navn	Dato og år	Formål og status	Nation
Phoenix	– 2007	Small Mars Scout Lander	USA
Mars 2009	– 2009	Mars Science Laboratory Rover	USA
Mars 2011	– 2011	Scout Mission	USA

Tabel A.1 Oversigt over tidligere, igangværende og fremtidige planlagte missioner til Mars. http://nssdc.gsfc.nasa.gov/planetary/chronology_mars.html

Transmissionen fra Mariner 4 foregik uden brug af kanalkodning og uden brug af datakomprimering, som på dette tidspunkt heller ikke havde fundet sin vej ind i rummissionerne endnu. Den manglende komprimering af data kombineret med datidens teknologi betød, at transmissionen varede fire dage. Transmissionsraten var på dette tidspunkt $8,33$ bit per sekund.

Mariner 4 havde flere formål, hvoraf fotograferingen af Mars' overflade kun var ét. Der skulle også udføres målinger af Mars' magnetfelt, kosmisk støv, kosmisk stråling og solvind. Desuden skulle missionen udgøre grundlaget for undersøgelser af operationelle og ingeniørmæssige teknikker til anvendelse i senere interplanetariske missioner.

Billederne fra Mariner 4 dækkede cirka 1 procent af overfladen i et diskontinuert bånd. Billederne viste en måneagtig overflade fyldt med kratere, hvilket var i strid med selv de mest konservative estimater af Mars'

topografi. Det viste sig senere at billederne forestillede en ældre geologisk region af planeten.

A.1.2 Mariner 6 & 7

Mariner 6 og 7 var to identiske fartøjer, der blev opsendt henholdsvis den 24. februar 1969 og den 27. marts 1969. Mariner 6 og 7 var fysisk tungere rumfartøjer end nogen andre rumfartøjer NASA tidligere havde opsendt. Sonderne passerede tættest ved Mars henholdsvis den 30. juli og den 4. august samme år. Mariner 7 fulgte en mere direkte bane til Mars end Mariner 6, hvorfor den indhentede Mariner 6 og således kun ankom fem dage senere. Fartøjerne tilvejebragte serier af såvel billeder taget på lang afstand som serier af billeder taget på kortere afstande. Hver af disse billeder bestod af 704 linier med hver 945 pixels. Billederne taget fra lang afstand havde en opløsning rangerende fra 4 til 43 km per pixel. Billederne på kortere afstand havde opløsninger helt ned til 300 m per pixel. I alt blev 143 billeder fra længere afstande og 58 'nærbilleder' transmitteret.

Med Mariner-missionen i 1969 blev kanalkodning for alvor en del af Mars-missionerne. Mariner 6 og 7 gjorde brug af en ikke-lineær udgave af en Reed-Muller kode[2]. Koden var en biortogonal blokkode med 64 (2^6) kodeord og en længde på 32 – altså en kode med rate $^6/_{32}$. Afkodningsproceduren for koden var soft-decision maximum-likelihood afkodning[3], baseret på den såkaldte 'Green Machine' udviklet af R. R. Green ved JPL. I modsætning til Mariner 4 havde Mariner 6 og 7 grundet den teknologiske udvikling computere om bord, som kunne omprogrammeres fra Jorden under missionen. Sonderne var i stand til at transmittere data med en rate på 16.200 bit per sekund – næsten to tusinde gange hurtigere end den rate, hvormed Mariner 4 havde transmitteret.

På vej til Mars blev forbindelsen til Mariner 7 midlertidigt mistet sandsynligvis på grund af brud på batteriet. Efter 7 timers stilhed blev den

Figur A.3 Illustration af Mariner 6 og 7. http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1969-014A

2 For en beskrivelse af Reed-Muller koder se for eksempel Roman (1992).
3 Maximum likelihood afkodning er forklaret i kapitel 6

dog genoprettet, men det varede ikke længe, førend det stod klart, at de instrumenter, der rapporterede orienteringen af TV-kameraerne, var blevet beskadiget og ikke længere fungerede korrekt. Uden denne information kunne kameraerne om bord på Mariner 7 ikke peges i retning af Mars, og eftersom fartøjet hastigt nærmede sig planeten var en hurtig løsning påkrævet. Problemet blev løst ved en manuel kalibrering af kameraerne og den 2. august begyndte Mariner 7 at transmittere billeder af Mars. Denne begivenhed viste vigtigheden af at have en omprogrammerbar computer om bord på rumfartøjerne.

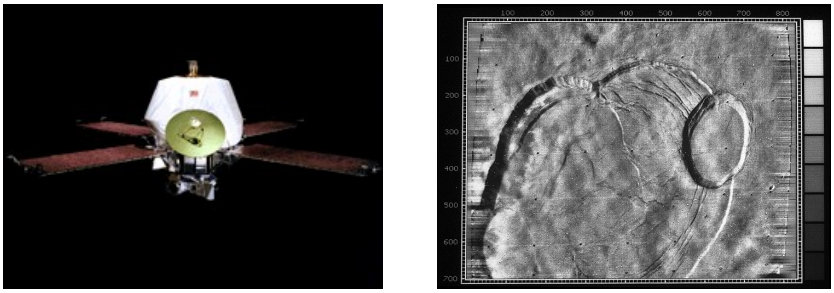
Resultatmæssigt var missionerne en succes. Der blev i alt taget 201 billeder, hvoraf nærbillederne dækkede omkring 10 procent af Mars' overflade. Selv om billederne ikke viste det stærkt varierende terræn, man havde håbet på, så gjorde de alligevel op med en del teorier baseret på observationer fra Jorden.

A.1.3 Mariner 9

Oprindeligt skulle Mariner 9 have haft en identisk rejsefælle, Mariner 8, men den 8. maj 1971 365 sekunder efter opsendelsen holdt raketten, som bar sonden, op med at fungere, og Mariner 8 styrtede i Atlanterhavet. Mariner 9 klarede imidlertid opsendelsen, som for dens vedkommende fandt sted den 30. maj 1971. Efter at have tilbagelagt mere end 600 millioner kilometer ankom Mariner 9 og gik i kredsløb om Mars den 14. november, 1971. Mariner 9 var NASA's første Mars-orbiter og udgjorde NASA's mest ambitiøse mission til Mars. Fartøjet vejede mere end Mariner 6 og 7 tilsammen og medbragte en række sofistikerede instrumenter.

Resultaterne fra Mariner 9 missionen kom til at overgå alle forventninger. Fartøjet klarede sig i månederne efter sin ankomst i november 1971 gennem en større støvstorm i Mars' atmosfære – ikke mindst på grund af den omprogrammerbare computer. I januar 1972 begyndte missionens automatiske affotografering af Mars' overflade. Affotograferingen kom med sine 7.329 billeder til at dække 80 procent af planetens overflade. Desuden blev Mars' to måner Phobos og Deimos fotograferet.

Sammenlagt blev 54 millioner bit data transmitteret. Dette var 27 gange så meget som data modtaget fra samtlige tidligere Mariner-missioner.



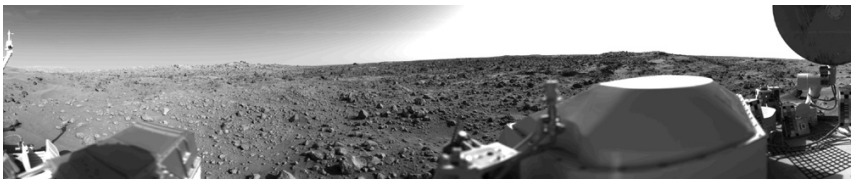
Figur A.4 Til venstre: Illustration af Mariner 9. http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1971-051A Til højre: Billede af Olympus Mons taget af Mariner 9. http://nssdc.gsfc.nasa.gov/imgcat/html/object_page/m09_mtvs4265_52.html

Omfanget af informationer fra Mariner 9 var revolutionerende, og missionen skulle da også lægge fundamentet for det fremtidige Viking-program. Mariner 9 benyttede samme system til kanalkodning, som blev introduceret med Mariner 6 og 7. Ligesom de tidligere Mariner-missioner gjorde Mariner 9 ikke brug af datakomprimering.

Affotograferingen viste den variation af såvel atmosfære som overflade, som forskere havde håbet på. Der blev observeret gamle flodsenge, kratere, store uddøde vulkaner, vandfronte, isskyer, lokale støvstorme, morgentåge og meget mere. Med aftrykkene af flydende stoffer på Mars' overflade og dermed muligheden for at der på et tidligere tidspunkt havde været flydende vand, blev spørgsmålet om eksistensen af liv på Mars mere relevant. Vejen for Viking-programmet var banet.

A.1.4 Viking-missionerne

Viking-missionerne[4] til Mars bestod af to rumfartøjer, Viking 1 og Viking 2, som hver bestod af en orbiter og en lander. De primære formål med missionerne var at tage billeder med høj opløsning af Mars' overflade, karakterisere strukturen og sammensætningen af såvel atmosfæren som overfladen, men i modsætning til Mariner-missionerne også denne gang at lede efter tegn på liv.

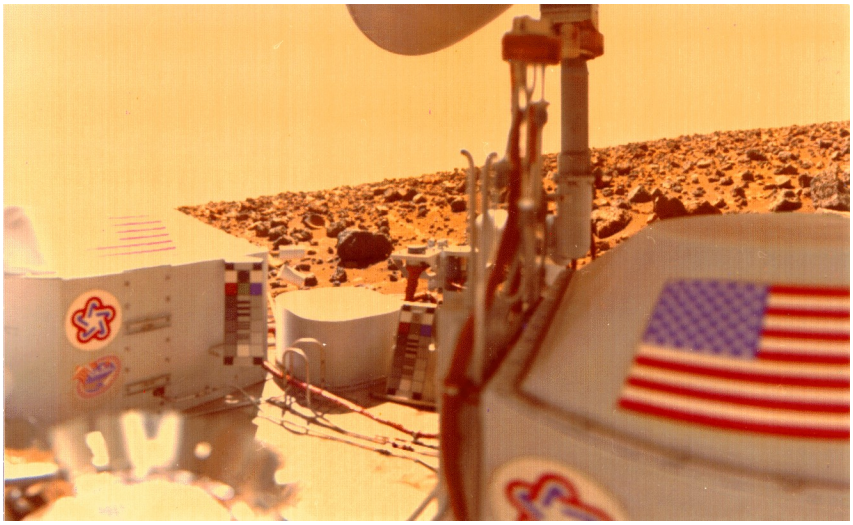


Figur A.5 Første panoramabillede af Mars' overflade set fra Viking 1 landeren. http://nssdc.gsfc.nasa.gov/imgcat/html/object_page/vl1_p17045.html

Viking 1 blev opsendt den 20. august 1975 og ankom til Mars den 19. juni 1976. Den første måned i kredsløb var opgaven at fotografere Mars' overflade primært for at finde egnede landingssteder til de to landere om bord på henholdsvis Viking 1 og Viking 2. Den 20. juli 1976 blev landeren om bord på Viking 1 koblet fra orbiteren og landede ved Chryse Planitia (se figur A.1). Viking 2 blev opsendt den 9. september 1975 og gik i kredsløb om Mars den 7. august 1976. Den 3. september 1976 landede Viking 2 landeren ved Utopia Planitia (se figur A.1).

Viking 2 orbiteren holdt op med at fungere den 25. juli 1978 efter at have foretaget 706 kredsløb om Mars, og Viking 1 orbiteren den 17. august 1980 efter at have kredset over 1.400 gange rundt om planeten. Viking-orbiterne nåede at fotografere hele Mars' overflade i en opløsning af 150 til 300 m per pixel og udvalgte områder i en opløsning på helt ned til 8 m per pixel. Viking-landerne transmitterede billeder af overfladen, tog overfladeprøver og analyserede disses sammensætning, ledte efter spor af liv samt studerede Mars' atmosfæres sammensætning og meteorologi.

4 Nærværende afsnit om Viking-missionerne 1 og 2 er baseret på http://nssdc.gsfc.nasa.gov/planetary/viking.html



Figur A.6 Billede taget fra Viking 2 landeren i Utopia Planitia. http://nssdc.gsfc.nasa.gov/imgcat/html/object_page/vl2_21c056.html

Viking-missionerne anvendte også den med Mariner 6 og 7 introducerede kanalkode og i stil med tidligere missioner heller ingen datakomprimering. Dataraten for Viking 1 og 2 var på 3.000 bit per sekund. Viking 2s lander sluttede sin kommunikation med Jorden den 11. april 1980. Viking 1 landeren afsluttede først sin den 13. november 1982 under omstændigheder antydet i det indledende citat. Tilsammen nåede de to landere at transmittere over 1.400 billeder af de to lokationer.

Resultaterne fra Viking-missionerne giver stadig et af de mest komplette billeder af Mars, omend de biologiske eksperimenter ikke viste tegn på liv. Der skulle nu gå tyve år førend NASA igen vendte tilbage til Mars.

A.1.5 Mars Pathfinder

Mars Pathfinder[5] var den anden[6] af NASA's såkaldte 'low-cost planetary discovery missions'. Pathfinder-missionen bestod af en stationær lander som ved Viking-missionerne og en mindre overflade-rover ved navn Sojourner. Det primære formål med missionen var at vise gennemførligheden af en landing og udforskning af Mars' overflade med få økonomiske midler. De videnskabelige mål omfattede undersøgelser angående det at bryde gennem atmosfæren, overfladefotografering på såvel kort som lang afstand, klippe- og jordbundssammensætning og i det hele taget at karakterisere miljøet på Mars med henblik på fremtidige missioner.

Mars Pathfinder blev opsendt den 4. december 1996. Fartøjet landede i Ares Vallis regionen (se figur A.1) den 4. juli 1997. Landingsproceduren for Pathfinder var lig den for MER, som er beskrevet i kapitel 1. Den 6.

5 Nærværende afsnit om Mars Pathfinder er baseret på http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1996-068A

6 MGS var den første 'low-cost' mission, men da MGS stadig er flyvende findes beskrivelsen af denne i næste afsnit.

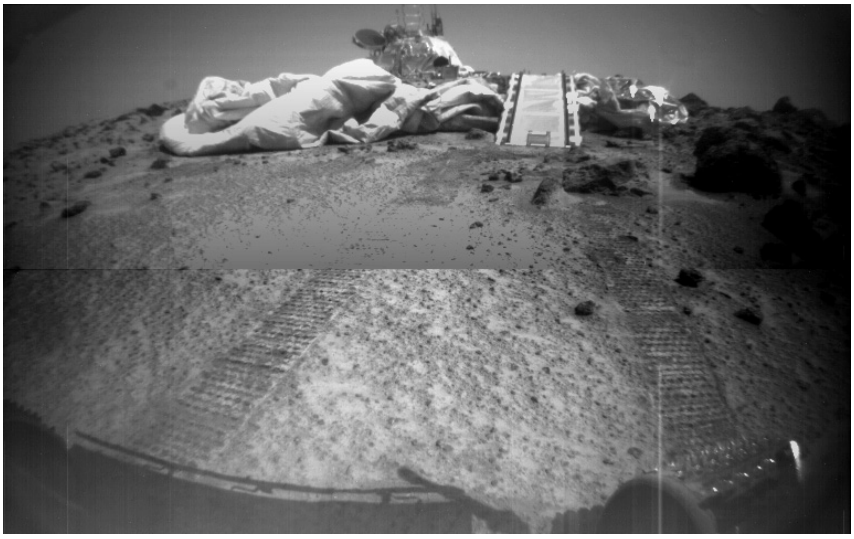
juli rullede Sojourner ned af rampen og tog sin første køretur på Mars' overflade.

Selve Sojourners vægt var 11,5 kg og derudover var i alt 6 kg på landeren allokeret til telekommunikation og lignende. På roverens egen hjemmeside[7] står der følgende om dens formål:

> The rover is primarily a technology experiment itself, designed to determine micro-rover performance in the poorly understood Martian terrain so that future rovers may be designed to be effective in navigating and moving about the surface of Mars.

Landeren skulle støtte roveren i dens gøremål blandt andet med transmission til Jorden. Derudover var landeren udstyret med en meteorologisk station, med genopladelige batterier og solceller. Landeren var endvidere udstyret med tre antenner samt et kamera påmonteret en 80 centimeter høj mast. Der blev taget billeder og udført eksperimenter af såvel lander som rover frem til den 27. september 1997, hvor forbindelsen forsvandt.

Grundet den udvikling, der var sket i de tyve år siden sidste overflademission, var der kommet kildekodning om bord på Pathfinder, og kanalkodningen var også en del mere avanceret. Pathfinder gjorde til kanalkodningen brug af et af CCSDS foreskrevet konkateneret system bestående af en indre $(6, 1, 15)$-foldningskode og en ydre Reed-Solomon kode med parametre $(255, 223)$. Til billedkomprimering med tab anvendtes den kommercielle algoritme JPEG. Til tabsfri komprimering Rice-kodning. Pathfinder havde en transmissionsrate på 8.000 bit per sekund.



Figur A.7 Første tilgængelige billede i fuld størrelse af Mars Pathfinder landeren set fra roveren Sojourner. http://nssdc.gsfc.nasa.gov/planetary/marspath_images_h.html

7 http://mpfwww.jpl.nasa.gov/rover/mission.html

Som en sidebemærkning til Mars Pathfinder missionen bør nævnes, at Danmark, med Jens Martin Knudsen (1930-2005) som ankermand, for første gang var involveret i en mission til Mars. De danske instrumenter om bord på Pathfinder skulle undersøge støvet på planetens overflade.

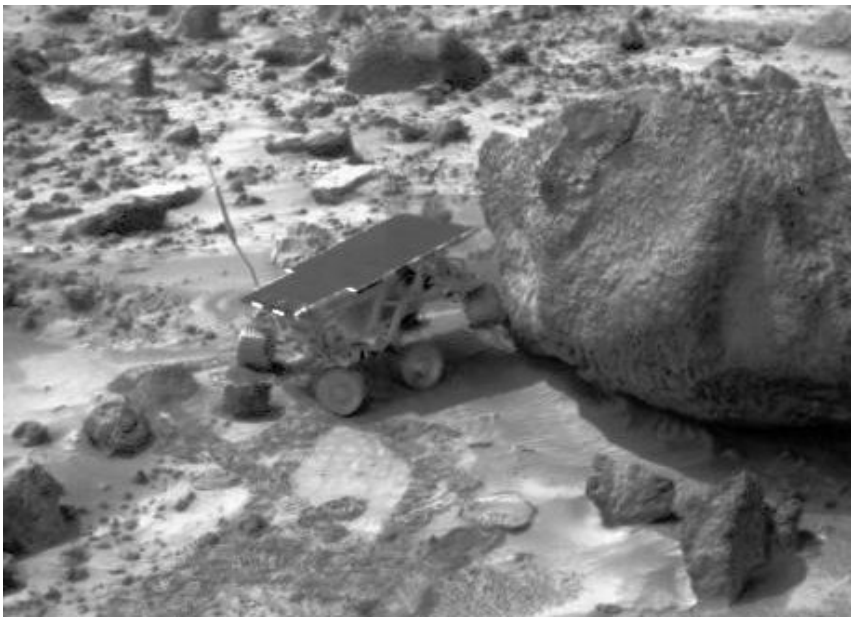
A.2 Igangværende missioner til Mars

Af igangværende missioner til Mars er der udover MER-missionen også to orbitere i kredsløb om planeten foruden en ny, som er på vej.

A.2.1 Mars Global Surveyor

Mars Global Surveyor (MGS)[8] var den første mission til Mars efter Viking-missionerne. MGS var ligeledes den første af NASA's såkaldte 'low-cost' missioner.

MGS er en orbiter med flere formål. For det første skal den fotografere Mars' overflade i høj opløsning, foretage studier af såvel topografi som tyngdefelt, afdække betydningen af vand og støv på og i Mars' overflade og atmosfære, undersøge vejr, klima og sammensætning af overflade og atmosfære såvel som Mars' magnetfelt. For det andet skulle MGS virke som en kommunikationsforstærker for fremtidige Mars-missioner. Denne funktion har MGS den dag i dag stadig.



Figur A.8 Billede af roveren Sojourner taget fra Mars Pathfinders lander. Roveren var 28 cm høj, 63 cm lang og 48 cm bred. http://nssdc.gsfc.nasa.gov/planetary/marspath_images_h.html

8 Nærværende afsnit om Mars Global Surveyor er baseret på http://nssdc.gsfc.nasa.gov/planetary/marsurv.html

MGS blev opsendt den 7. november 1996 og gik efter 10 måneder i transit i kredsløb om Mars den 12. september 1997. Transmissionsraten for MGS er på 128.000 bit per sekund, og der anvendes til kanalkodning et CCSDS-standardiseret konkateneret system bestående af en ydre $(255, 223)$-Reed-Solomon kode og en indre $(2, 1, 7)$-foldningskode. Billedkomprimeringen er tabsfri og foregår med Rice-kodning.

A.2.2 Mars Odyssey

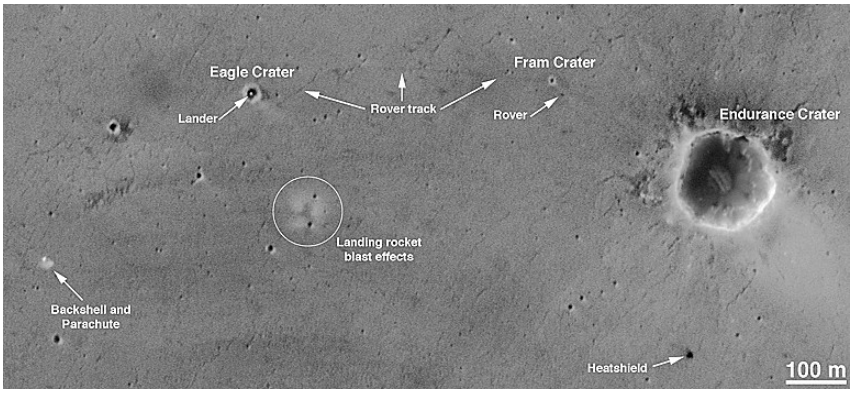
2001 Mars Odyssey[9] er den tilbageblivende del af Mars Surveyor 2001-projektet, der oprindeligt omfattede en orbiter og en lander. Landeren blev imidlertid aflyst, og orbiteren blev i den forbindelse omdøbt til 2001 Mars Odyssey. Odyssey skulle oprindeligt kun løbe over tre år, men er foreløbig blevet forlænget frem til september 2006. Ligesom MGS har Odyssey fungeret og fungerer stadig som kommunikationsforstærker for MERs rovere. Odysseys kilde- og kanalkodning er den samme som for MGS.

Odyssey blev opsendt den 7. april 2001. Efter syv måneder i transit nåede fartøjet Mars den 24. oktober 2001. Fartøjet gik derefter ind i et 18,6 timers langt elliptisk kredsløb. Kredsløbet blev i nogle måneder løbende justeret indtil det blev et 2 timers langt omtrent cirkulært kredsløb.

De oprindelige mål for 2001 Mars Odyssey var at indsamle data til belysning af forudsætningerne for livets opståen på Mars, at karakterisere klima og geologi på Mars, at studere potentielle strålingsfarer for fremtidige astronaut-missioner samt at fungere som kommunikationsforstærker for fremtidige Mars-missioner.

Figur A.9 Til venstre: Illustration af Mars Global Surveyor. http://mpfwww.jpl.nasa.gov/mgs/index.html Til højre: Billede af Coprates Catena, en mindre dal på Mars, taget af MGS. http://nssdc.gsfc.nasa.gov/imgcat/html/object_page/mgs_88217.html

9 Nærværende afsnit om 2001 Mars Odyssey er baseret på http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=2001-014A



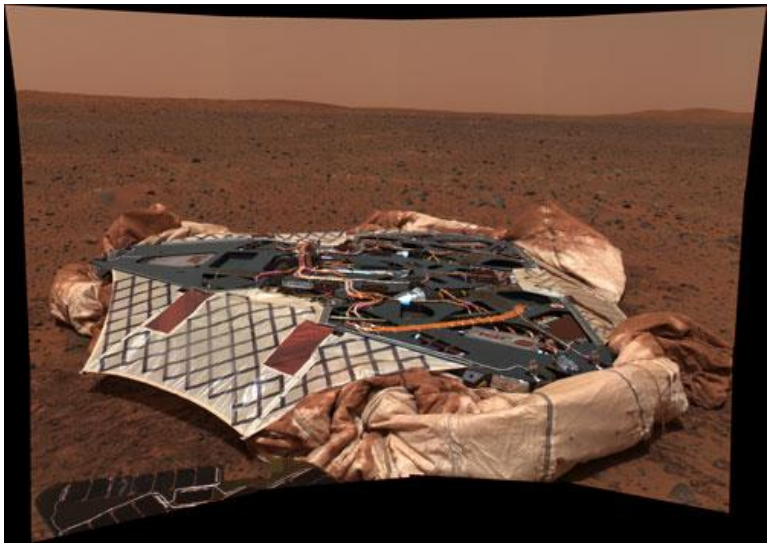
Figur A.10 MER-rover Opportunity og dens færden set fra Mars Global Surveyor den 24. januar 2005. http://www.msss.com/mars_images/moc/2005/01/24/index.html

A.2.3 MER-missionerne

Mars Exploration Rover (MER)[10] missionerne omfatter de to rovere Spirit og Opportunity. MER-missionerne gik igang med opsendelsen af roverne i sommeren 2003, og roverne ankom til Mars i januar måned året efter. Det oprindelige mål var, at missionerne skulle operere i tre måneder, men denne tidsramme er indtil videre blevet forlænget med mere end et år, og roverne kører stadig. Roverne er i stand til at køre over hundrede meter per dag – væsentligt længere afstande end Sojourner var i stand til at tilbagelægge i 1997 – og nye rekorder er siden missionens begyndelse blevet

Figur A.11 Til venstre: Opsendelse af Spirit (MER A) den 10. juni 2003 fra Cape Canaveral Air Force Station. http://www.nasa.gov/lb/missions/solarsystem/merb_covg.html Til højre: Opsendelse af Opportunity (MER B) den 8. juli 2003 fra samme sted. http://en.wikipedia.org/wiki/Opportunity_rover_timeline

10 Afsnittet om Spirit og Opportunity er baseret på henholdsvis http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=2003-027A og http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=2003-032A



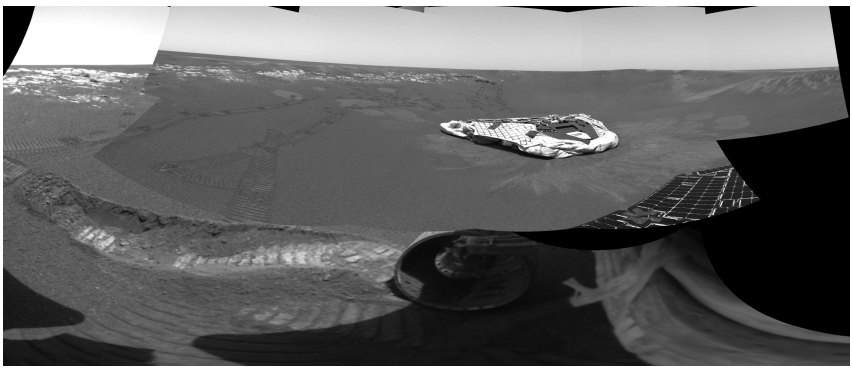
Figur A.12 Spirits landingsmodul set fra roveren. http://www.seds.org/~spider/spider/Mars/Hi-res/meralander.jpg

sat løbende. De to rovere har hver især et helt batteri af videnskabelige instrumenter om bord (heriblandt dansk udviklede).

Som ved et flertal af de andre Mars-missioner er et af formålene med MER at indsamle data, som skal hjælpe med at afgøre, om liv nogensinde opstod på Mars, klima og geologi skal karakteriseres, og der skal gøres forberedelser til fremtidige menneskelige udforskninger af Mars. Igen er der tale om at analysere landskabet og lede efter tegn på om flydende vand har været til stede på den røde planet.

Ligesom Sojourner er Spirit og Opportunity udstyret med hver seks hjul. Hjulene er forbundet til et såkaldt rocker-bogie system, der bevirker at roveren er mere fleksibel, når den kører i ujævnt terræn. Hvert hjul har sin egen motor og de to forhjul henholdsvis baghjul er uafhængigt styrbare. Tophastigheden er på 5 cm/s, men gennemsnitshastigheden vil i praksis være mindre da roveren skal undgå forhindringer. Roveren kan modstå en hældning på op til 45 grader uden at vælte, men er programmeret til at undgå hældninger på over 30 grader. Computeren om bord på roveren er sammen med batterier og andre elektroniske komponenter beskyttet mod store temperatursvingninger i en speciel kasse monteret på roveren. Panoramakamera og navigationskameraer er monteret på en mast oven på roveren 1,4 m over hjulenes base. Påmonteret roveren er også et periskop, diverse spektrometre, et kamera til mikroskopibilleder foruden en magnet. Der er monteret to kameraer i hver ende af roveren, hvis formål er at opdage eventuelle forhindringer.

Spirit blev opsendt den 10. juni 2003. Den gik ind i Mars' atmosfære og landede på planeten den 4. januar 2004. Opportunity, der blev opsendt den 8. juli 2003, landede på planeten den 25. januar 2004. Gusev-krateret (se figur A.1) blev valgt som landingssted for Spirit på grund af dets udseende som tidligere søbund. Hvis krateret på et tidligere tidspunkt var



Figur A.13 Billede af Opportunitys landingssted ved Meridiani Planum. Billedet er en mosaik af data fra blandt andet navigationskameraerne. http://nssdc.gsfc.nasa.gov/planetary/mars/mars_exploration_rovers/merb_images.html

fyldt med vand vil denne bund måske indeholde interessante sedimenter. Den 15. januar kørte Spirit ud på Mars overflade.

Opportunity landede i Terra Meridiani (se figur A.1) også kendt som 'Hematite Site', da det er rigt på forekomster af hematit, som er et jernholdigt mineral, der ofte dannes i vand. Ydermere er netop dette sted et fladt og jævnt landskab, hvorfor det blev betragtet som et sikkert steder at lande. I løbet af de første fire dage blev roveren 'foldet ud', hvorefter den kørte ud på Mars' overflade.

MERs transmissionsrate er på 168.000 bit per sekund. Systemet til kanalkodning er magen til det, der blev anvendt af Mars Pathfinder, bortset fra at MER også anvender $(2, 1, 7)$-foldningskoden. Kildekodningen er væsentlig anderledes, idet der til billedkomprimering med tab anvendes ICER og til komprimering uden tab LOCO-I.

Med hensyn til de foreløbige resultater fra MER-missionen fortalte Folkner (2005b), at man har været i stand til at identificere sten, der ligner meteoritter fundet på Antarktis – meteoritter, der indeholder spor af liv. Derudover har roverne fundet betydelige spor på tidligere eksistens af vand på Mars' overflade, blandt andet i form af lagdelte sten. William Folkner tilskriver i høj grad disse resultater det faktum, at roverne er i stand til at bevæge sig rundt på overfladen. (Folkner; 2005b, side 75 i Interviews)

A.2.4 Mars Reconnaissance Orbiter

Mars Reconnaissance Orbiter[11] (MRO) er designet til at kredse om Mars i ét Mars-år, som svarer til 687 dage på Jorden. MRO er udstyret med i alt seks videnskabelige instrumenter inklusiv et højopløsningskamera. De videnskabelige formål med missionen er at karakterisere det nuværende klima på Mars samt de fysiske mekanismer med hensyn til klima-

11 Nærværende afsnit om Mars Reconnaissance Orbiter er baseret på http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=MARSRECON

Figur A.14 Til venstre: Opsendelsen af MRO den 12. august, 2005. Til højre: Illustration af Mars Reconnaissance Orbiter. http://nssdc.gsfc.nasa.gov/planetary/mars_future.html

ændringer, sæsonskift og så videre, at identificere vand-relaterede landskaber foruden at finde egnede landingssteder for fremtidige missioner, specielt tur-retur missioner. MRO vil transmittere højopløsningsbilleder, studere Mars' overflade, lede efter vand i undergrunden, analysere støv og vand i atmosfæren samt overvåge vejret. Ligesom MGS og Odyssey skal MRO også fungere som kommunikationsforstærker mellem Jorden og fremtidige Mars-missioner.

Mars Reconnaissance Orbiter blev opsendt den 12. august i år (2005). Rejsen til Mars tager cirka syv måneder, hvorefter planen er, at sonden skal gå i kredsløb om planeten. Går alt planmæssigt vil NASA da have i alt tre satellitter i kredsløb om planeten: Mars Global Surveyor, 2001 Mars Odyssey samt Mars Reconnaissance Orbiter.

MRO har en datarate på 12 Mbit per sekund. Den anvender tabsfri komprimering med FELICS, og som den første Mars-mission gør den brug af CCSDS-standardiserede turbokoder[12] til kanalkodningen.

A.3 Fremtidige missioner til Mars

NASAs næste to Mars-missioner er allerede mere eller mindre planlagt, men også missioner længere ude i fremtiden er på tegnebordet.

A.3.1 Phoenix Lander

Phoenix Lander er en mindre såkaldt scout mission[13] der er planlagt til at finde sted i 2007. Phoenix er designet til at studere Mars' overflade i nærheden af landingsstedet, som skal befinde sig på de nordlige breddegrader. I løbet af sommeren 2008 er der planlagt tre måneders arbejde for Phoenix på Mars.

12 Turbokoder er kort beskrevet i kapitel 7.

13 Nærværende informationer stammer dels fra http://nssdc.gsfc.nasa.gov/planetary/mars_future.html og dels fra http://phoenix.lpl.arizona.edu/

Målene for Phoenix Lander er at bestemme, om der nogensinde opstod liv på Mars, at karakterisere klimaet på Mars, at karakterisere Mars' geologi og at forberede menneskelig udforskning af planeten. Specielt skal disse mål opnås ved at studere isen, som befinder sig lige under Mars' overflade.

A.3.2 Mars Science Laboratory

Et videnskabeligt mobilt laboratorium[14] i form af en langtrækkende rover udstyret til at udføre diverse videnskabelige studier af Mars er planlagt til sidste kvartal af 2009. De primære videnskabelige formål med missionen er følgende: At opregne det biologiske potentiale for minimum ét i forvejen udpeget område, at karakterisere lokal geologi og geokemi, at udforske muligheden for senere bosætning herunder tilstedeværelsen af vand i området samt at karakterisere et bredt spektrum af overfladestråling.

Missionen er planlagt til at løbe over minimum et Mars-år. Landingszonen for denne mission er endnu ikke udvalgt, men kriteriet for en eventuelt lokation er dennes kapacitet til at kunne opretholde liv.

A.4 Afrunding

Siden 1964 har NASA opsendt i alt 18 missioner til Mars. Ud af disse har seks været forfejlede og elleve må betegnes som succesfulde. Ud af sidstnævnte er fire (to orbitere og to rovere) stadig igang, og derudover er MRO på vej til Mars.

Med hensyn til kildekoderne har flere forskellige strategier været i spil til både komprimering med og uden tab. Senest fandt den kommercielle JPEG vej ind i deep-space anvendelserne med Pathfinder-missionen. Også listen af kanalkoder i Mars-missionerne bærer præg af afprøvning af en række forskellige tilgange. I tabel A.2 er vist en oversigt over kanal- og kildekodningen i Mars-missionerne.

Mission	År	Datarate	Kanalkodning	Kildekodning
Mariner 4	1964	8,33 bps	Ingen	Ingen
Mariner 6 & 7	1969	16200 bps	$(32, 6)$-RM	Ingen
Mariner 9	1971	–	$(32, 6)$-RM	Ingen
Viking 1 & 2	1975	3000 bps	$(32, 6)$-RM	Ingen
MGS	1996	128000 bps	$(2, 1, 7)$-F + RS	Rice
Odyssey	1996	–	$(2, 1, 7)$-F + RS	Rice
Pathfinder	1996	8000 bps	$(6, 1, 15)$-F + RS	JPEG/Rice
MER A & B	2003	168000 bps	$(2, 1, 7)$-F + RS/ $(6, 1, 15)$-F + RS	ICER/ LOCO-I
MRO	2005	12 Mbps	CCSDS-turbo	FELICS

Tabel A.2 Tabellen angiver datarater samt kanal- og kildekodning for tidligere Mars-missioner. RS er Reed-Solomon koden med parametre $(255, 223)$. Tabellen er baseret på (McEliece; 2004a, side 5).

14 Informationerne stammer fra http://nssdc.gsfc.nasa.gov/planetary/mars_future.html

Inden for de næste 10 år[15] planlægger NASA yderligere en række orbitere og landere foruden den første mission, der skal bringe prøver fra Mars' overflade tilbage til Jorden. Nuværende planer tyder på en såkaldt Mars Scout mission med opsendelse i år 2011, den første mission med tilbagesendelse af prøver i år 2014 og en yderligere sådan med opsendelse i år 2016.

På endnu længere sigt er der planer om at sende de første mennesker til Mars. Materialerne opsamlet på Mars' overflade har blandt andet betydning for design af rumdragter til kommende Mars-astronauter. Det indtil videre mest optimistiske bud på en bemandet mission til Mars er år 2025, ikke mindst da der foruden tur-retur missionen til Mars også først skal foretages bemandede træningsmissioner til månen.

For nyligt har NASA fået bevilget 100 milliarder dollars til en sådan mission. Målet er inden 2020 at udvikle et nyt fartøj til bemandede rumrejser og at vende tilbage til månen i dette fartøj. Fartøjet skal anvendes til en senere bemandet mission til Mars. [16]

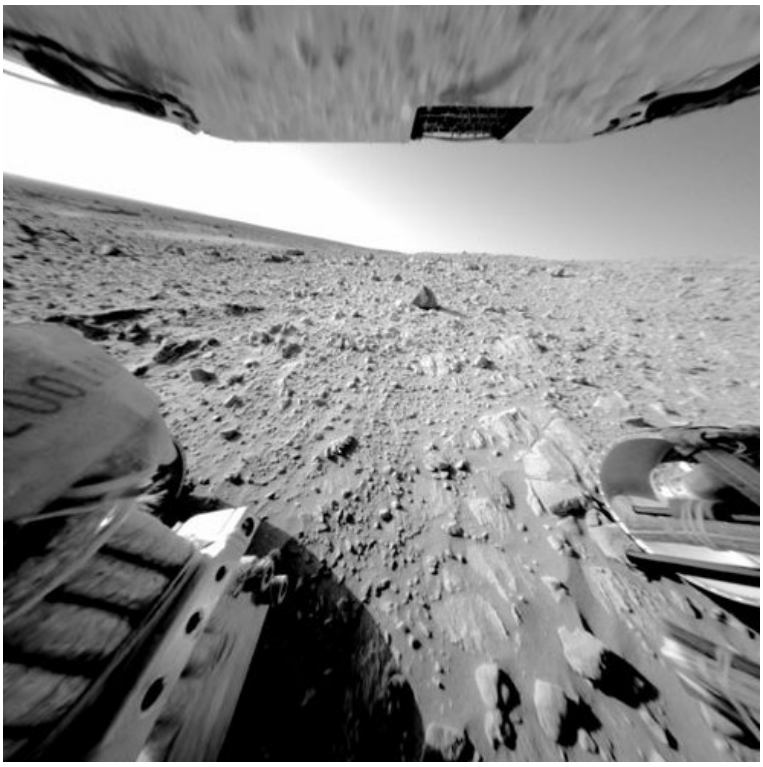
George W. Bush vilje til sådanne missioner kom til udtryk i forbindelse med hans fremlæggelse[17] af USA's nye rumprogram den 14. januar 2004:

> With the experience and knowledge gained on the moon, we will then be ready to take the next steps of space exploration: Human missions to Mars and to worlds beyond. Robotic missions will serve as trail blazers. The advanced guard to the unknown. Probes, landers and other vehicles of this kind continue to prove their worth, sending spectacular images and vast amounts of data back to Earth. Yet the human thirst for knowledge ultimately can not be satisfied by even the most vivid pictures or the most detailed measurements. We need to see and examine and touch for ourselves. And only human beings are capable of adapting to the inevitable uncertainties posed by space travel. As our knowledge improves we will develop new power generations, propulsion, life support and other systems that can support more distant travels. We do not know where this journey will end, yet we know this: *Human beings are headed into the cosmos.*

15 Oplysningerne stammer fra http://nssdc.gsfc.nasa.gov/planetary/mars_future.html

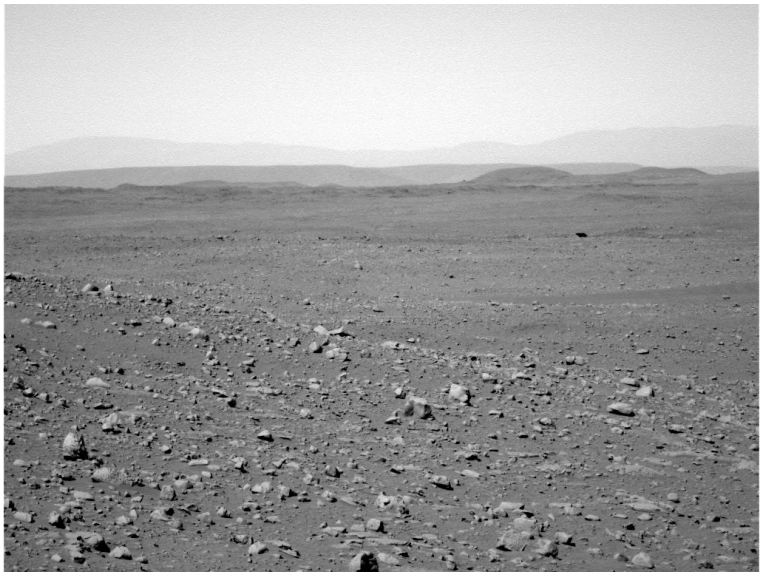
16 Se http://www.nasa.gov/missions/solarsystem/cev.html

17 Citatet er transskriberet fra videoen *A Renewed Spirit of Discovery* på http://www.nasa.gov/missions/solarsystem/explore_main.html



Matematikken bag Mars-missionen

Indførelse i den grundlæggende teori for
kildekodning og kanalkodning i MER



Matematikspeciale af

Uffe Thomas Jankvist & Bjørn Toldbod

Oktober 2005 ∘ IMFUFA ∘ Roskilde Universitetscenter

What are the mysteries of the bits?
Aaron Kiely, JPL, 2005

Forord

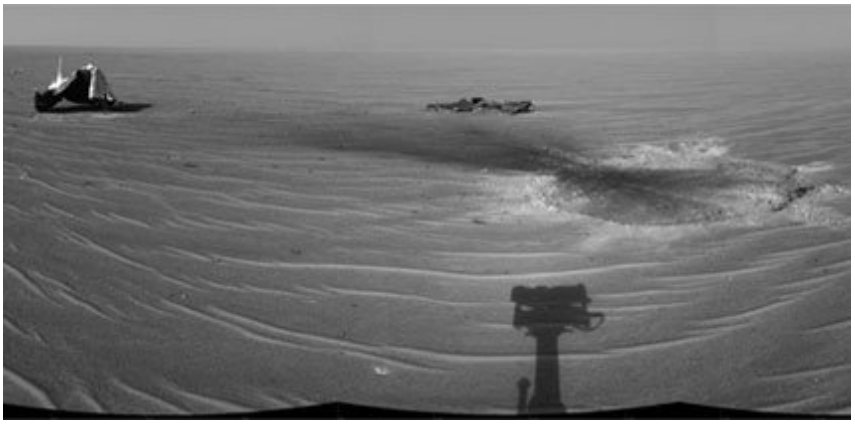
Denne del af specialet giver en indførelse i den grundlæggende teori for billedkomprimering og kanalkodning i MER.†

Indholdet af INDFØRELSEN forudsættes i diskussionen af MERs konkrete teknikker til billedkomprimering og kanalkodning i HOVEDRAPPORTEN. INDFØRELSEN kan ses som et udtryk for den mængde teori, vi har sat os ind i og gjort os fortrolige med for at kunne give en beskrivelse af kilde- og kanalkodningen i MER. INDFØRELSEN er baseret på diverse lærebogsfremstillinger, men trækker også på et udvalg af originalartikler.

INDFØRELSEN indeholder først en overordnet indledning. Dernæst indeholder den to dele: Del I omhandler billedkomprimering og del II omhandler kanalkodning.

Uffe Thomas Jankvist & Bjørn Toldbod
Institut for Matematik og Fysik
Roskilde Universitetscenter
Februar 2006

† Billederne på rapportens omslag er taget af Spirit og kan findes på http://marsrovers.jpl.nasa.gov/gallery/all/



Figur 0.15 Opportunitys varmeskjold og skjoldets landingssted på Mars' overflade. http://marsrovers.jpl.nasa.gov/gallery/press/opportunity/20041227a/1NN325EFF40CYLA3P0685L000M1-crop-B330R1_br.jpg

Indhold

1 Indledning **1**

I Kildekodning i MER-missionen **5**

2 Introduktion til data- og billedkomprimering **5**
2.1 Komprimering med og uden tab af data 6
2.2 Elementære begreber i datakomprimering 7
2.3 Eksempler på simple komprimeringsteknikker 13
2.4 Afrunding 15

3 Wavelets **17**
3.1 Førstetrins waveletbasis 17
3.2 Filterbanks 23
3.3 p'te trins waveletbasis 26
3.4 Billedkomprimering med wavelets 32
3.5 Lifting 34
3.6 Afrunding 38

4 Modellering **39**
4.1 Diskret hukommelsesløs kilde 39
4.2 Markov-kilde 42
4.3 Sammensatte kilder 45
4.4 Afrunding 46

5 Entropikodning **47**
5.1 Entropi og kodning 47
5.2 Huffman-kodning 48
5.3 Golomb-kodning 55
5.4 Sammenflettet entropikodning 57
5.5 Adaptiv entropikodning overordnet 59
5.6 Afrunding 62

II Kanalkodning i MER-missionen 65

6 Introduktion til kanalkodning 65
6.1 Elementære kodningsteoretiske begreber 66
6.2 Fejldetektion versus fejlkorrektion 68
6.3 En mere intuitiv tilgang 69
6.3 Perfekte koder 69
6.5 Eksempler på simple koder 73
6.6 Afrunding 74

7 Lineære koder 75
7.1 Grundlæggende for lineære koder 75
7.2 Cykliske koder 78
7.3 Afrunding 84

8 Reed-Solomon koder 85
8.1 Reed-Solomon koder som cykliske koder 85
8.2 En moderne tilgang til Reed-Solomon koder 86
8.3 Reed-Solomon koder og bit 90
8.4 Afrunding 92

9 Foldningskoder 95
9.1 Fra blokkoder til foldningskoder 95
9.2 Foldningindkoderen 96
9.3 Polynomiel beskrivelse af foldningskoder 99
9.4 Egenskaber for foldningskoder 103
9.5 Matrixbeskrivelse af foldningskoder 105
9.6 Gitterbeskrivelse af foldningskoder 107
9.7 Afrunding 109

Litteratur 111

A Fouriertransformationen på $\mathbb{Z}$ 117

B Wavelets 127
B.1 Waveletbeviser 127
B.2 Resultater for Haar-basen 144

1 Indledning

I denne delrapport gives en indførelse i de matematiske teorier kildekodning og kanalkodning. INDFØRELSEN indeholder således alle de detaljer vedrørende teorierne, som er udeladt i HOVEDRAPPORTEN.

INDFØRELSEN består af to dele. Del I handler om billedkomprimering og del II om kanalkodning. Begge dele underforstår til dels de begreber og problemstillinger, der er beskrevet i kapitel 3 i HOVEDRAPPORTEN.

Del I består af fire kapitler, som på hver sin måde bidrager til forståelsen af MERs komprimeringsteknikker, ICER og LOCO-I. Kapitel 2 er en generel introduktion til kildekodning, herunder hvad der forstås ved en kilde, hvilke egenskaber, der er attraktive for kildekoder og ikke mindst, hvad der forstås ved komprimering med og uden tab. Begge typer komprimering indgår i MER. Den billedkomprimering, som vi vil beskrive, kan opfattes som en tredelt proces; forbehandling, modellering og entropikodning. Formålet med forbehandling og modellering er i den sidste ende at give en mere effektiv entropi-indkodning og dermed komprimering.

Forbehandling beskrives med udgangspunkt i teorien for wavelettransformationer i kapitel 3. Forbehandlingen kan anskues fra flere forskellige vinkler. Vi præsenterer to tilgange, en filter-tilgang og en lifting-tilgang. Den første benyttes hovedsageligt til at give et teoretisk overblik, mens den sidste er simpel i praksis og derfor interessant for MER. Waveletteorien er generelt en teori for funktioner defineret på $\mathbb{R}$ eller $\mathbb{C}$, men fordi vi kun skal bruge teorien til billedkomprimering, kan vi klare os med at se på wavelets defineret på $\mathbb{Z}$.

Modellering af en kilde, som i det tilfælde, vi interesserer os for, er et billede, beskrives i kapitel 4. Beskrivelsen er opdelt for kilder med og uden hukommelse. Sidstnævnte kaldes Markov-kilder, men i forbindelse med billedkomprimering kendes de typisk under navnet kontekst-modeller. Begge MERs algoritmer anvender kontekst-modellering.

Beskrivelsen af entropikodning i kapitel 5 indledes med en beskrivelse af Huffman-kodning, som på eksemplarisk vis forklarer, hvad entropikodning går ud på, og som desuden anvendes af LOCO-I. Dernæst forklares Golomb-kodning, som er en anden teknik til entropikodning. Teknikken anvendes både af LOCO-I og ICER. ICER gør brug af en lidt datalogisk orienteret ramme for entropikodning kaldet sammenflettet entropikodning. Denne er ligeledes forklaret. Sidst eksemplificeres adaptiv entropikodning ved adaptiv Huffman-kodning.

Del II består ligeledes af fire kapitler, som hver især bidrager til forståelsen af MERs system til kanalkodning. Dette system kombinerer to forskellige koder, en Reed-Solomon kode og en foldningskode. Kapitel 6 udgør en generel introduktion til kanalkodning, det vil sige en indførelse i de mest elementære kodningsteoretiske begreber, samt en diskussion af fejldetektion versus fejlkorrektion.

Kapitel 7 giver en generel, men kortfattet indførelse i lineære blokkoder samt cykliske koder, der er et særtilfælde af de lineære koder. De cykliske koder anvendes til at indføre Reed-Solomon koderne, hvilket gøres i kapitel 8.

Foldningskoder, der beskrives i kapitel 9, er ikke blokkoder i generel forstand, og kræver lidt mere forarbejde end Reed-Solomon koderne. I kapitlet gives afslutningsvis en beskrivelse af foldningskoder ved hjælp af trellis-diagrammer. Sådanne diagrammer gør det nemmere at forstå afkodningen af koderne.

Del II beskæftiger sig fortrinsvist med at beskrive kodernes matematiske struktur, samt hvordan data indkodes. Afkodning er henlagt til HOVEDRAPPORTEN, hvor de indgår i en mere MER-orienteret sammenhæng.

INDFØRELSEN har også to appendikser. Appendiks A indeholder en gennemgang af forskellige egenskaber for den diskrete Fouriertransformation. Resultaterne fra dette appendiks benyttes både i beskrivelsen af kilde- og kanalkodning, men videreføres også i appendiks B. Dette appendiks indeholder samtlige beviser for de sætninger, der anvendes i kapitel 3.

I den anvendelse, som vi ser på, handler kilde- og kanalkodning om forskellige metoder til at komprimere og på sikker vis transmittere bit. INDFØRELSEN beskriver den matematik, som ligger til grund for bit-operationerne og kan derfor med rette siges at afdække »the mysteries of the bits« for MERs kilde- og kanalkodning.

Del I

Kildekodning i MER-missionen

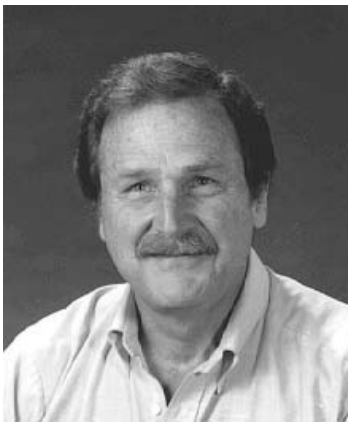
2 Introduktion til data- og billedkomprimering

> Previously people were limited to very simple ideas in data compression just because they didn't have much processing time and the computers weren't very powerful, so they just tried something very simple to see if they could get an improvement and if so they used it. (Klimesh; 2005, side 45 i INTERVIEWS)

I nærværende del af rapporten skal vi studere den del af kommunikationssystemet, som omhandler kildekodningen. Kildekodningen beskæftiger sig med komprimering af data.

For information repræsenteret ved en mængde bit er *datakomprimering* kunsten at repræsentere den samme information, eller i nogle tilfælde *næsten* den samme information, ved et færre antal bit. Typen af information, hvad enten det er tekst, naturlige billeder, computergrafik, video eller audio, har stor betydning for måden at komprimere på. En komprimeringsteknik, som er god for én type, er ikke nødvendigvis god for en anden. Strukturen af informationerne i hver af de nævnte eksempler er forskellig, hvilket de mest effektive komprimeringsteknikker således også er. Vi skal særligt se på teknikker til komprimering af naturlige billeder, da det er sådanne billeder, der tages på Mars.

Som omtalt i HOVEDRAPPORTEN stammer væsentlige elementer af datakomprimeringen fra Shannon (1948). Ideen om komprimering kan dog spores længere tilbage end dette. Et tidligt eksempel er Morse-koden, ud-



David A. Huffman
(1925-1999)

Huffman modtog allerede som 18-årig sin bachelorgrad i elektroteknik fra Ohio State University. Herefter tjente han i en periode som officer i US Navy, hvorefter han genoptog studierne og fik en kandidatgrad fra Ohio State University og en ph.d.-grad fra MIT, begge ligeledes i elektroteknik. Huffman er mest kendt for Huffman-kodning, en teknik som han udviklede som ph.d.-studerende ved MIT i 1952. Udover bidrag til informations- og kodningsteori bidrog Huffman væsentligt til andre områder, for eksempel matematiske egenskaber for papiroverflader samt teknikker til foldning af usædvanlige papirskulpturer. Huffman var ved sin død professor emeritus i datalogi ved University of California, Santa Cruz. Huffman modtog blandt andet en Louis E. Levy Medal fra Franklin Institute og en R. W. Hamming Medal fra IEEE for sit pionerende arbejde, herunder Huffman-kodning samt sine studier af asynkrone sekventielle kredsløb. http://www.soe.ucsc.edu/people/faculty/huffman.html

viklet af Samuel Morse i midten af 1800-tallet. Med Morse-kode indkodes bogstaver som kombinationer af prikker og streger. Hyppigt forekommende bogstaver som a og e angives med (·) og (–), mens sjældne bogstaver som q og j angives ved de længere følger (– – · –) og (· – – –). Ideen med at tildele symboler med stor sandsynlighed for at optræde kortere kodeord er den grundlæggende tanke i entropikodning, som hyppigt anvendes i dag (se kapitel 5). Den mest kendte teknik til entropikodning er Huffman-kodning, som blev udviklet af David A. Huffman i 1952.

De i dag anvendte teknikker til data- og billedkomprimering er langt mere omfattende end eksemplet med Morse-kodning og består af flere forskellige trin. De følgende kapitler af del I beskriver disse trin samt de teknikker, som anvendes i MERs billedkomprimering. Vi vil her overordnet skitsere de forskellige trin: Når man skal komprimere et billede, opfatter man det som en matrix bestående af pixelværdier. I moderne komprimeringsteknikker foretages gerne en forbehandling af matricen efterfulgt af en modellering af denne. Disse trin øger effektiviteten af entropikodningen, som foretages tilsidst. Entropikoderen kan opfattes som 'maskinen' i en komprimeringsalgoritme.

Formålet med dette kapitel er at få slået visse begreber fast. Disse begreber af stor betydning for den videre gennemgang. Fremstillingen er hovedsageligt baseret på (Sayood; 2000, side 1-35). Hvor anden litteratur inddrages, vil dette blive angivet.

2.1 Komprimering med og uden tab af data

Når man taler om en komprimeringsteknik eller -algoritme, taler man i virkeligheden om to algoritmer:

1. Den algoritme som fra inddata x genererer en komprimeret version x_c repræsenteret ved et færre antal bit.
2. Rekonstruktionsalgoritmen, som opererer på x_c for at generere en rekonstruktion y.

Konventionen på området er, at en komprimeringsalgoritme dækker over såvel komprimering som rekonstruktion.

Med udgangspunkt i kravene til rekonstruktionen kan komprimeringsalgoritmer inddeles i to klasser: Komprimering *uden tab* af data, hvor det kræves at $x = y$ og komprimering *med tab* af data[1]. Komprimering med tab resulterer generelt i en langt højere grad af komprimering, fordi der her tillades, at $y \neq x$.

Tabsfri komprimeringsteknikker sørger for, at informationer ikke går tabt. Denne type komprimering anvendes fortrinsvist til applikationer, hvor det er vitalt, at rekonstruerede data ikke afviger fra de oprindelige. For komprimeringsteknikker, der involverer et tab af data, forholder det sig typisk således, at komprimeret data ikke kan genvindes eller rekonstrueres nøjagtigt. For komprimering af eksempelvis billeder udnytter man ofte, at det menneskelige sanseapparat ikke altid er i stand til at skelne mellem to billeder. Man kan altså med andre ord smide noget information væk i et billede, uden at det menneskelige øje vil opfatte dette.

1 Disse to klasser af komprimeringsteknikker kaldes på engelsk henholdsvis *lossless* og *lossy*.

Komprimering af billeder involverer derfor oftest et tab, idet man her er villig til at acceptere en vis forvrængning (distortion) i bytte for en højere grad af komprimering.

2.2 Elementære begreber i datakomprimering

Enhver informations- eller datagenererende proces, som vi vil kalde en *symbolgenerator*, kan betragtes som en *informationskilde*, eller blot *kilde*, der er en matematisk model for processen. Formelt defineres en kilde ved følgende:

Definition 2.1 (Kilde)
En kilde $\mathcal{K}$ *er et ordnet par* $(\mathcal{S}, \mathcal{P})$*, hvor* $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ *er en mængde af symboler kaldet* kildealfabetet *og* $\mathcal{P} = \{p(s_1), p(s_2), \ldots, p(s_n)\}$ *er en sandsynlighedsfordeling på* $\mathcal{S}$*.* (Roman; 1992, side 11)

De fleste tekster, denne rapport inklusiv, kan beskrives ved en kilde med et alfabet, som indeholder alle ASCII-symbolerne. Den computer, som rapporten er skrevet på, anvender binære data, som kan opfattes som en følge af symboler fra alfabetet $\{0, 1\}$. Et n-bit digitalt billede i gråtoner kan beskrives ved en kilde med et alfabet bestående af 2^n symboler, som hver især repræsenterer forskellige pixelværdier for forskellige gråtoner i billedet. (En mere indgående diskussion af kilder findes i kapitel 4.)

En parameter, der gør sig gældende, når man skal vælge en kode for et kildealfabet, er den gennemsnitlige længde af koden. En kode $\mathcal{C}$ for en kilde $\mathcal{K}$ med sandsynlighedsfordeling $\mathcal{P}$ har en gennemsnitlig længde l givet ved

$$l = \sum_{i=1}^{n} p(s_i) l_i \text{ bit per symbol,}$$

hvor l_i er antallet af bit i c_i, kodeordet for s_i. Der ønskes generelt en kode med så lille l som muligt, hvilket som i tilfældet med Morse-koden kan opnås ved at give de symboler, der optræder hyppigst, de korteste kodeord. Des færre bit vi bruger til at repræsentere symboler som optræder ofte, des færre bit vil vi *i gennemsnit bruge per symbol.*

Definition 2.2 (Rate)
Det gennemsnitlige antal af bit per symbol kaldes for kodens rate*.*

En lille gennemsnitlig længde for en kode er ikke altid nok. En kode skal også kunne overføre information på en manér, som ikke giver anledning til flertydighed i afkodningen.

Definition 2.3
En kode, hvorom gælder, at en given følge af kodeord kan afkodes på én og kun én måde, siges at være entydigt dechifrerbar[2]*.*

Lad os se et eksempel.

Eksempel 2.4
Lad os se på to koder $\mathcal{C}_1$ og $\mathcal{C}_2$ af variabel længde med samme kildealfabet,

2 *Fra engelsk: Uniquely decodable.*

$\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, og samme sandsynlighedsfordeling givet ved

$$\mathcal{P} = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}.$$

Koderne er givet ved:

$$\mathcal{C}_1 = \{0, 1, 00, 11\} \quad \text{og} \quad \mathcal{C}_2 = \{0, 10, 110, 111\}.$$

Den gennemsnitlige længde af $\mathcal{C}_1$ kan beregnes til $1,25$ og den gennemsnitlige længde af $\mathcal{C}_2$ til $1,75$. På denne baggrund ser $\mathcal{C}_1$ ud til at være den mest attraktive kode for kilden. Dette ændrer sig dog, når vi betragter afkodning af koderne.

I hver af koderne er kildesymbolerne blevet tildelt entydige kodeord, hvilket må være første krav til entydig dechifrerbarhed. Betragter vi imidlertid følgen af kildesymboler givet ved; $s_2\, s_1\, s_1$, vil denne følge ved indkodning med $\mathcal{C}_1$ resultere i følgen; 100. Denne følge kan afkodes som $s_2\ s_1\ s_1$, men også som $s_2\ s_3$, hvorfor $\mathcal{C}_1$ ikke er entydigt dechifrerbar.

Lad os nu se på $\mathcal{C}_2$. Tilstedeværelsen af et 0 i en følge angiver termineringen af et kodeord. Kodeordet for s_4 er tre bit langt og slutter ikke på et 0. Da alle andre kodeord har færre end tre 1-taller og slutter på et 0 er koden entydigt dechifrerbar. Afkodningsproceduren for $\mathcal{C}_2$ er simpel: Ét kodeord slutter, når man enten modtager et 0, eller når man har modtaget tre 1-taller i træk. (Gallager; 1968, side 44-46) ◇

For $\mathcal{C}_2$ i ovenstående eksempel 'ved' afkoderen øjeblikkeligt, når et kodeord slutter. En kode med denne egenskab kaldes en *øjeblikkelig*[3] kode.

Der kan opstilles en nødvendig betingelse for længderne af kodeordene i en entydigt dechifrerbar kode. Denne betingelse er kendt som *Kraft-McMillans ulighed* og er første del af et større resultat, som vi skal tilvejebringe senere i dette afsnit.

Sætning 2.5 (McMillan)
Lad $\mathcal{C}$ være en kode med N kodeord med længderne $l_1, l_2, \ldots, l_N$. Hvis $\mathcal{C}$ er entydigt dechifrerbar, gælder uligheden:

$$\mathscr{K}(\mathcal{C}) = \sum_{i=1}^{N} 2^{-l_i} \leq 1,$$

også kaldet Kraft-McMillans ulighed.

Bevis
Betragt den n'te potens af $\mathscr{K}(\mathcal{C})$. Hvis $\mathscr{K}(\mathcal{C})$ er større end 1, skal $\mathscr{K}(\mathcal{C})^n$ vokse eksponentielt med n. Hvis dette ikke er tilfældet, har vi bevis for at $\mathscr{K}(\mathcal{C}) \leq 1$.

Lad n være et vilkårligt heltal. Da vil

$$\begin{aligned}\left(\sum_{i=1}^{N} 2^{-l_i}\right)^n &= \left(\sum_{i_1=1}^{N} 2^{-l_{i_1}}\right)\left(\sum_{i_2=1}^{N} 2^{-l_{i_2}}\right)\cdots\left(\sum_{i_n=1}^{N} 2^{-l_{i_n}}\right) \quad (2.1)\\ &= \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_n=1}^{N} 2^{-(l_{i_1}+l_{i_2}+\ldots+l_{i_n})}.\end{aligned}$$

3 Fra engelsk: Instantaneous.

Eksponenten $l_{i_1} + l_{i_2} + \ldots + l_{i_n}$ er længden af n kodeord fra $\mathcal{C}$. Da ethvert kodeord er mindst én bit lang gælder, at den mindste værdi som denne eksponent kan antage er større end eller lig n. Hvis derimod $l = \max\{l_1, l_2, \ldots, l_N\}$ vil den største værdi, eksponenten kan antage være mindre end eller lig nl. Derfor kan ligning (2.1) skrives som

$$\mathscr{K}(\mathcal{C})^n = \sum_{k=n}^{nl} A_k 2^{-k},$$

hvor A_k er antallet af kombinationer af n kodeord, som har en kombineret længde på k.

Lad os se nærmere på størrelsen af A_k. Antallet af mulige forskellige binære følger af længde k er 2^k. Hvis $\mathcal{C}$ er entydigt dechifrerbar, kan hver følge repræsentere én og kun én følge af kodeord. Det betyder, at antallet af mulige kombinationer af kodeord, hvis kombinerede længde er k ikke kan overstige 2^k. Altså $A_k \leq 2^k$, hvilket igen betyder

$$\mathscr{K}(\mathcal{C})^n = \sum_{k=n}^{nl} A_k 2^{-k} \leq \sum_{k=n}^{nl} 2^k 2^{-k} = n(l-1) + 1. \tag{2.2}$$

Ovenstående viser, at $\mathscr{K}(\mathcal{C})$ er mindre end eller lig en størrelse, som vokser lineært. Altså kan $\mathscr{K}(\mathcal{C})$ ikke vokse eksponentielt med n, og der må for en entydigt dechifrerbar kode gælde, at $\mathscr{K}(\mathcal{C}) \leq 1$. □

Kraft-McMillans ulighed fortæller, at man ikke kan have en kode med mange korte kodeord, hvis man ønsker entydig dechifrerbarhed.

Eksempel 2.6
For koden $\mathcal{C}_1$ i eksempel 2.4 bliver $\mathscr{K}(\mathcal{C}) = 1/2 + 1/2 + 1/4 + 1/4 = 3/2$, hvilket viser, at koden ikke er entydigt dechifrerbar. Muligheden for at tilføje kodeord til denne kode er faktisk ikke-eksisterende, når man har medtaget de første to kodeord. For $\mathcal{C}_2$ bliver $\mathscr{K}(\mathcal{C}) = 1/2 + 1/4 + 1/8 + 1/8 = 1$. Koden udelukkes altså ikke af sætning 2.5, men der kan ikke tilføjes yderligere kodeord til koden, hvis dechifrerbarheden skal være entydig. ◇

Entydig dechifrerbarhed er en attraktiv egenskab ved en kode. Vi mangler imidlertid en bedre metode til at afgøre, hvorvidt en given kode er entydigt dechifrerbar eller ej. Vi skal om lidt bringe en sådan metode, men først skal vi se på et par eksempler, der indikerer, hvad man i en sådan test skal kigge efter.

Eksempel 2.7
Betragt de to koder

$$\mathcal{C}_3 = \{0, 01, 11\} \quad \text{og} \quad \mathcal{C}_4 = \{0, 01, 10\},$$

som hver repræsenterer kildealfabetet $\mathcal{S} = \{s_1, s_2, s_3\}$ med en ikke nærmere specificeret sandsynlighedsfordeling.

Betragt koden $\mathcal{C}_3$ og den indkodede følge

$$011111111111111111.$$

Her er det første kodeord enten 0, svarende til s_1, eller 01, svarende til s_2, men hvilken, der er den rigtige, kan ikke afgøres, førend vi har afkodet

hele følgen. Antages det, at det første kodeord svarer til s_1, afkodes de følgende seksten bit til otte s_3'er. Imidlertid ender vi således med et 1-tal, som ikke kan afkodes. Vælger vi derimod det første kodeord svarende til s_2 afkodes de resterende seksten bit som otte s_3'er, og der er ingen bit tilovers. Følgen kan altså afkodes entydigt, men $\mathcal{C}_3$ er ikke øjeblikkelig.

Betragt i stedet koden $\mathcal{C}_4$ og den indkodede følge

$$01010101010101010.$$

Det første kodeord i denne følge kan enten være 0, svarende til s_1, eller 01, svarende til s_2. Begge valg vil føre til en meningsfuld afkodning: I første tilfælde fås s_1 efterfulgt af otte s_3'er. I andet tilfælde fås s_2 efterfulgt af syv s_3'er og til sidst et s_1. Man har som modtager af ovenstående følge ingen mulighed for at vide, hvilken af de to afkodninger der er den korrekte, hvorfor $\mathcal{C}_4$ ikke er en entydigt dechifrerbar kode. Bemærk i øvrigt, at begge koder opfylder Kraft-McMillans ulighed. (Sayood; 2000, side 28-29) ◇

Inden vi kan give metoden til bestemmelse af entydig dechifrerbarhed, må vi have endnu et par begreber på banen. Betragt en følge

$$\Phi = \phi_1\, \phi_2\, \ldots\, \phi_m$$

bestående af symboler fra et kildealfabet. Et *udsnit* af følgen vil da være på formen

$$\phi_i\, \phi_{i+1}\, \phi_{i+2}\, \ldots\, \phi_j,$$

hvor $1 \leq i \leq j \leq m$. Med et *prefix* for følgen Φ forstås en delfølge

$$\phi_1\, \phi_2\, \phi_3\, \ldots\, \phi_i,$$

hvor $1 \leq i \leq m$. De sidste $m - i$ elementer i følgen

$$\phi_{i+1}\, \phi_{i+2}\, \phi_{i+3}\, \ldots\, \phi_m,$$

kaldes et *hængende suffix* for Φ.

Eksempel 2.8
Hvis for eksempel $a_1 = 1100$ og $a_2 = 1100101$ er a_1 et prefix af a_2, og det resterende hængende suffix er 101. ◇

Eksempel 2.7 indikerer, at en test for entydig dechifrerbarhed kan tage sit udgangspunkt i, hvorvidt et hængende suffix er et kodeord eller ej. Testen for entydig dechifrerbarhed, som er givet i følgende algoritme stammer fra Gallager (1968, side 512-513).

Algoritme 2.9
Inddata*: En liste indeholdende samtlige kodeord for en given kode $\mathcal{C}$.*

1. *Undersøg alle kodeord parvis for at se om et kodeord er et prefix af et andet kodeord.*
2. *Når et sådant par opdages tilføjes det hængende suffix til listen af kodeord, med mindre det allerede er tilføjet i en tidligere iteration.*
3. *Gentag skridt 1 og 2 med denne større liste og forsæt således indtil ét af følgende to stopkriterier indtræffer:*

A. *Der optræder et hængende suffix, som er et kodeord.*

B. Der er ikke flere entydige hængende suffix'er at tilføje til listen.

Uddata*: I tilfælde A er koden* ikke *entydigt dechifrerbar. I tilfælde B* er *koden entydigt dechifrerbar.*

Lad os igen se på koderne $\mathcal{C}_3$ og $\mathcal{C}_4$ fra eksempel 2.7.

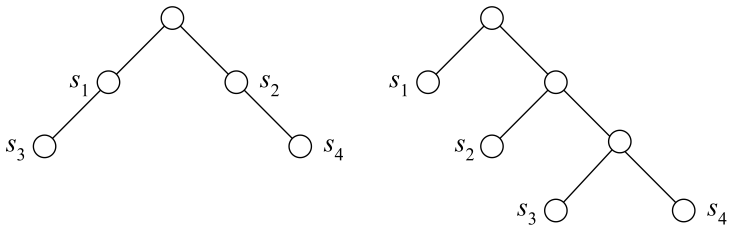
Eksempel 2.10
Betragt koden $\mathcal{C}_3$. Listen indeholdende dennes kodeord er $\{0, 01, 11\}$. Kodeordet 0 er et prefix af kodeordet 01. Det hængende suffix er 1. Der er ikke flere prefix'er at finde, så vi tilføjer det hængende suffix til listen, som da bliver $\{0, 01, 11, 1\}$. Ved igen at kigge efter prefix'er i denne nye liste finder vi, at 1 er et prefix af 11 med hængende suffix 1. Men da 1 allerede er tilføjet til listen befinder vi os altså i tilfælde B i algoritme 2.9, hvorfor $\mathcal{C}_3$ er entydigt dechifrerbar.

Betragt dernæst koden $\mathcal{C}_4$. Dennes liste er $\{0, 01, 10\}$. Kodeordet 0 er et prefix af kodeordet 01. Det hængende suffix er 1. Da der i første omgang her heller ikke er flere prefix'er at finde, tilføjer vi det hængende suffix til listen, som da bliver $\{0, 01, 10, 1\}$. Nu er 1 et prefix af 10, med hængende suffix 0. Da 0 i $\mathcal{C}_4$ er et kodeord for s_1, befinder vi os i tilfælde A i algoritme 2.9, hvorfor $\mathcal{C}_4$ ikke er en entydigt dechifrerbar kode. (Sayood; 2000, side 30) ◇

Problemet, om man så må sige, med hensyn til entydig dechifrerbarhed, er altså disse hængende suffixer, som i sig selv udgør kodeord. En kode, for hvilken denne situation aldrig indtræffer, besidder *prefix-egenskaben.*

Definition 2.11 (Prefix-kode)
En kode siges at være en prefix-kode*, eller at besidde* prefix-egenskaben*, hvis intet kodeord er et prefix af et andet kodeord, det vil sige, at når* $c = \xi_1\xi_2 \ldots \xi_n$*, så er* $\xi_1\xi_2 \ldots \xi_k$ *for* $1 \leq k \leq n$ *ikke et kodeord.*

En måde at afgøre hvorvidt en given kode er et prefix er ved at tegne et rodfæstet binært træ (hver knude har højst to forgreninger) for koden. På figur 2.1 er tegnet de binære træer for koderne $\mathcal{C}_1$ og $\mathcal{C}_2$ fra eksempel 2.4 – hver forgrening mod venstre repræsenterer et 0 og hver forgrening mod højre repræsenterer et 1-tal. For en prefix-kode er kodeordene kun associeret med *eksterne* knuder, det vil sige knuder, hvorfra der ikke udgår nogen forgreninger, i modsætning til *interne* knuder, hvorfra der gør.



Figur 2.1 Til venstre: Binært træ for ikke-prefix-koden $\mathcal{C}_1 = \{0, 1, 00, 11\}$. Til højre: Binært træ for prefix-koden $\mathcal{C}_2 = \{0, 10, 110, 111\}$. (Sayood; 2000, side 31)

Kodeordet for et kildesymbol kan opnås ved at traversere træet fra roden og ned til den eksterne knude, der repræsenterer symbolet. Som det fremgår af figuren er $\mathcal{C}_1$ ikke en prefix-kode, men det er $\mathcal{C}_2$ derimod. En særlig, og for MERs komprimeringsalgoritme vigtig klasse af prefix-koder, er de såkaldte *udtømmende* prefix-koder. En prefix-kode $\mathcal{C}$ siges at være *udtømmende*, hvis der ikke kan tilføjes kodeord til koden uden at krænke prefix-egenskaben for denne. Koden $\mathcal{C}_2$ ses at være et eksempel på en sådan udtømmende prefix-kode.

Vi har altså i prefix-koderne en klasse af koder der er entydigt dechifrerbare. Spørgsmålet er imidlertid, om vi mister noget ved udelukkende at begrænse vores opmærksomhed til disse koder. Kan vi finde koder med kortere kodeord, hvis vi også medtager ikke-prefix-koder? Svaret er nej. For en hvilken som helst entydigt dechifrerbar ikke-prefix-kode kan vi altid finde en prefix-kode med samme kodeordslængder. Resultatet hviler på sætning 2.5 af McMillan samt følgende sætning af Kraft. Sætningen siger, at hvis vi har en mængde af kodeordslængder, der opfylder Kraft-McMillans ulighed, så kan vi altid finde en prefix-kode med disse kodeordslængder.

Sætning 2.12 (Kraft)
Givet en mængde af heltal $\{l_1, l_2, \ldots, l_N\}$, som opfylder Kraft-McMillans ulighed, kan vi altid finde en prefix-kode med kodeordslængder $l_1, l_2, \ldots, l_N$.

Bevis
Sætningen bevises ved at udvikle en procedure til konstruktion af en prefix-kode med kodeordslængder $l_1, l_2, \ldots, l_N$ opfyldende Kraft-McMillans ulighed. Det kan uden tab af generalitet antages, at $l_1 \leq l_2 \leq \ldots \leq l_N$. Definér en følge af tal $w_1, w_2, \ldots, w_N$ ved:

$$w_1 = 0, \quad w_j = \sum_{i=1}^{j-1} 2^{l_j - l_i},$$

for $j > 1$. Den binære repræsentation af w_j for $j > 1$ kræver $\lceil \log_2 w_j \rceil$ bit. Denne binære repræsentation bruges til at konstruere en prefix-kode.

Først bemærkes det, at antallet af bit i den binære repræsentation af w_j er $\leq l_j$. Dette følger trivielt for w_1. For $j > 1$,

$$\begin{aligned} \log_2 w_j &= \log_2 \left(\sum_{i=1}^{j-1} 2^{l_j - l_i} \right) = \log_2 \left(2^{l_j} \sum_{i=1}^{j-1} 2^{-l_i} \right) \\ &= l_j + \log_2 \left(\sum_{i=1}^{j-1} 2^{-l_i} \right) \leq l_j. \end{aligned} \tag{2.3}$$

Den sidste ulighed kommer fra forudsætningen om opfyldelse af Kraft-McMillans ulighed, som giver, at

$$\sum_{i=1}^{j-1} 2^{-l_i} \leq 1.$$

Da summen er mindre end eller lig 1 er logaritmen negativ eller nul, og dermed er ligning 2.3 opfyldt.

Anvendende den binære repræsentation af w_j, kan vi udtænke en binær kode på følgende vis: Hvis $\lceil \log_2 w_j \rceil = l_j$, vil det j'te kodeord c_j være den binære repræsentation af w_j. Hvis $\lceil \log_2 w_j \rceil < l_j$, vil c_j være den binære repræsentation af w_j med $l_j - \lceil \log_2 w_j \rceil$ nuller tilføjet på højre side. Nok er dette en kode, men er det en prefix-kode? Kan vi vise, at $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$ er en prefix-kode, har vi vist sætningen ved konstruktion.

Antag det modsatte; at $\mathcal{C}$ er en ikke-prefix-kode. Så vil, for et eller andet $j < k$, c_j være et prefix af c_k. Det betyder, at de l_j mest betydende bit af w_k danner den binære repræsentation af w_j. Derfor, hvis vi flytter den binære repræsentation af w_k $l_k - l_j$ bit mod højre, bør vi få den binære repræsentation for w_j. Vi kan skrive dette som $w_j = \lfloor w_k / 2^{l_k - l_j} \rfloor$. På den anden side har vi, at

$$w_k = \sum_{i=1}^{k-1} 2^{l_k - l_j},$$

hvorfor

$$\frac{w_k}{2^{l_k - l_j}} = \sum_{i=0}^{k-1} 2^{l_j - l_i} = w_j + \sum_{i=j}^{k-1} 2^{l_j - l_i} = w_j + 2^0 + \sum_{i=j+1}^{k-1} 2^{l_j - l_i} \geq w_j + 1.$$

Det vil sige, at den mindste værdi for $w_k / 2^{l_k - l_j}$ er $w_j + 1$. Dette er i modstrid med kravet om, at c_j skal være prefix'et af c_k. Derfor kan c_j ikke være et prefix af c_k, og da j og k var arbitrære, betyder dette, at intet kodeord kan være et prefix af et andet kodeord. Altså er $\mathcal{C}$ en prefix-kode. (McEliece; 2002, side 332-334) □

Sætningerne af McMillan og Kraft fortæller os altså, at vi ved at begrænse vores opmærksomhed til prefix-koder ikke overser nogen ikke-prefix-koder, som er entydigt dechifrerbare, og som har en kortere gennemsnitlig længde. Vi vil derfor begrænse vores videre studie af kildekoder i kapitel 5 til prefix-koder.

2.3 Eksempler på simple komprimeringsteknikker

Vi vil her præsentere to eksempler på simple teknikker til komprimering. Begge teknikker tager udgangspunkt i strukturen af de givne data.

Eksempel 2.13 (Differenser mellem model og datasæt)
Antag, at vi har en følge af tal givet ved:

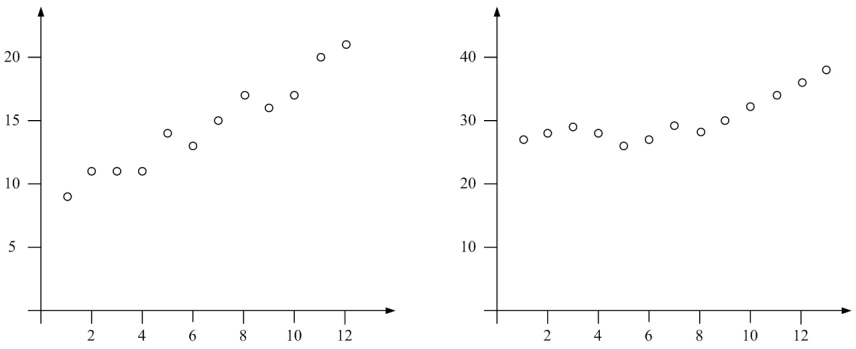
$$\{9, 11, 11, 11, 14, 13, 15, 17, 16, 17, 20, 21\}.$$

Hvis man blot gemmer den binære repræsentation af disse tal, skal man bruge fem bit per tal, fordi tallene ligger mellem 0 og 2^5. Studerer man strukturen af tallene, vil man imidlertid lægge mærke til at de omtrent falder på en ret linie – se figur 2.2. En model for dataene kunne derfor være den rette linie givet ved forskriften

$$\tilde{x}_n = n + 8,$$

for $n = 1, 2, \ldots, 12$. Differensen e_n mellem de oprindelige data x_n og modellen $\tilde{x}_n$,

$$e_n = x_n - \tilde{x}_n,$$



Figur 2.2 Til venstre: Graf for datasættet i eksempel 2.13. Til højre: Graf for datasættet i eksempel 2.14.

er givet ved følgen

$$\{0, 1, 0, -1, 1, -1, 0, 1, -1, -1, 1, 1\}.$$

Da denne følge kun består af tre forskellige tal (−1, 0, 1) kan vi repræsentere hvert element i den ved hjælp af to bit, for eksempel:

$$\begin{aligned} -1 &\mapsto 00 \\ 0 &\mapsto 01 \\ 1 &\mapsto 10. \end{aligned}$$

Komprimering kan derfor opnås ved kun at beholde parametrene for modellen samt følgen af differenser.

Pointen her er, at når man først har gennemskuet strukturen i datasættet, kan man bruge denne til at 'forudsige' værdien af hvert element i følgen og derefter indkode differensen. (Sayood; 2000, side 7-8) ◇

Eksempel 2.14 (Fortløbende differenser i datasæt)
Antag, at vi har nedenstående følge af tal:

$$\{27, 28, 29, 28, 26, 27, 29, 28, 30, 32, 34, 36, 38\}.$$

Som det fremgår af figur 2.2 følger afbildningen af dette datasæt ikke en ret linie som det var tilfældet i foregående eksempel. Imidlertid kan observeres at forskellen på to på hinanden følgende tal i datasættet ikke er større end to. Mere præcist er de på hinanden følgende differenser begyndende med datasættets første tal, 27, givet ved:

$$\{1, 1, -1, -2, 1, 2, -1, 2, 2, 2, 2, 2\}.$$

Da der kun optræder fire forskellige differenser, kan vi igen nøjes med to bit til at repræsentere disse. Tallet 27 sendes efterfulgt af følgen af differenser. Afkoderen lægger hver modtaget værdi til den forrige afkodede værdi og opnår dermed det oprindelige datasæt. (Sayood; 2000, side 8) ◇

Ovenstående eksempel virker umiddelbart som en fornuftig tilgang til komprimering af billeder, hvor pixelværdier for to nabopixels typisk ikke vil afvige meget fra hinanden.

2.4 Afrunding

Vi har i dette kapitel givet en introduktion til kildekodning af diverse former for data samt præsenteret et udvalg af begreber herunder komprimering med og uden tab, informationskilder, entydigt dechifrerbare koder samt visse resultater for disse koder. Derudover har vi givet nogle eksempler på simple komprimeringsteknikker.

I de næste kapitler af del I skal vi have uddybet forbehandlingen med wavelets, modelleringen af de forbehandlede data samt se forskellige teknikker til entropi-indkodning af disse.

3 Wavelets

> Data compression got a big boost of excitement when wavelets brought in a new sort of possibility that's not perfect at all, but it opened a new door. (Strang; 2005, side 31 i INTERVIEWS)

I dette kapitel skal vi beskæftige os med matematikken bag en klassisk disciplin i den ingeniør-orienterede matematik kaldet *signalanalyse*. De signaler, som vi vil interessere os for, er *diskrete* signaler. Kapitlet bygger på (Frazier; 1999, side 165-263). Derudover trækkes der på en række definitioner og resultater fra appendiks A.

På et overordnet plan indeholder signalanalyse og -behandling typisk en eller anden transformation af det givne signal. Et klassisk eksempel er den velkendte Fouriertransformation, som opløser signalet i harmoniske svingninger. Vi vil i kapitlet beskrive en anden type af transformationer kaldet diskrete wavelettransformationer. Wavelettransformationer fås i mange forskellige udgaver med dertil hørende forskellige egenskaber og udgør derfor et meget mere alsidigt analyseapparat end Fouriertransformationen.

Komprimeringsalgoritmen ICER består som tidligere beskrevet af en wavelettransformation som efterfølges af en modellering og en entropikodning. Formålet med transformationen er at forbehandle billederne, så de bedre kan modelleres og entropikoderes. Selve wavelettransformationen er, som vi senere skal se, beregningsmæssig simpel, men sværere at forstå matematisk uden det rette fundament. Dette fundament vil vi forsøge at støbe i kapitlet. De resultater, som vi ønsker at nå frem til, kræver en del bevisarbejde, som vi så vidt muligt har placeret i appendiks B.

Kapitlets resultater vil ved første øjekast synes noget teoretiske, og vi vil derfor illustrere dem ved brug af et gennemgående eksempel, samt afslutte kapitlet med en tilgang, som er mere praktisk orienteret.

3.1 Førstetrins waveletbasis

Karakteristisk for elementerne i den i appendiks A beskrevne Fourierbasis for $\mathbb{C}^N$ er, at disse svarer til harmoniske svingninger af varierende frekvens. Kapitlets hovedformål er at opstille en alternativ basis for $\mathbb{C}^N$ kaldet en *waveletbasis*. Det første indledende skridt på vejen tages i dette afsnit.

Basiselementerne i en typisk waveletbasis har ofte følgende egenskab:

Definition 3.1
En vektor $z \in \mathbb{C}^N$ siges at være lokaliseret i rum omkring n_0*, hvis z's komponenter $z(n)$ er nul eller relativt små, undtagen komponenter tæt på n_0.*

Hvis $B = \{v_i\}_{i=0}^{N-1}$ er en basis for $\mathbb{C}^N$, kan ethvert element $z \in \mathbb{C}^N$ skrives som en linearkombination af elementerne fra B, nemlig

$$z = \sum_{i=0}^{N-1} b_i v_i,$$

hvor $b_i \in \mathbb{C}$. Hvis hvert element v_i fra B er lokaliseret i rum omkring et til elementet hørende n_i, giver basen os mulighed for at give en simpel beskrivelse af z, når vi kun interesserer os for z's værdi i nærheden af et bestemt n. Lad k være et heltal tæt på dette n. Så gælder, da v_i'erne er lokaliseret i rum, at z kan tilnærmes ved

$$z(k) \sim \sum_{j=0}^{l} b_{i_j} v_{i_j}(k),$$

hvor v_{i_j}'erne er de l vektorer, som er lokaliseret i rum omkring k.

Eksempel 3.2
Elementerne i den euklidiske basis $B = \{e_i\}_{i=0}^{N-1}$ er fuldstændigt lokaliseret i rum, da $e_i(j) = 0$ undtagen for $j = i$. Den lokale beskrivelse af z er derfor meget simpel. Lad $z \in \mathbb{C}^N$ og lad

$$z = \sum_{i=0}^{N-1} b_i e_i.$$

For hvert $n = 0, \ldots, N-1$ gælder så, at

$$z(n) = \sum_{i=0}^{N-1} b_i e_i(n) = b_n.$$

Den euklidiske basis siges derfor at være *fuldstændigt* lokaliseret i rum. ◇

Eksempel 3.3
Betragt Fourierbasen for $\mathbb{C}^N$. For et element F_m i basen gælder, at

$$|F_m(n)| = \left| \frac{1}{N} e^{2\pi i m n/N} \right| = \frac{1}{N}$$

for alle $n = 0, \ldots, N-1$, altså er Fourierbasen ikke lokaliseret i rum, men tværtimod fuldstændigt spredt ud i rum. De *Fouriertransformerede* til basens elementer er imidlertid fuldstændigt lokaliserede i rum, da $\hat{F}_m(n) = 1$ for $m = n$ og 0 ellers. Fourierbasens elementer kan derfor opfattes som værende fuldstændigt lokaliserede i *frekvens*. ◇

Baser, som er lokaliseret i rum, er særligt anvendelige inden for billedkomprimering. Hvis man forestiller sig to billeder som er ens undtagen i et lille område, vil de to billeders repræsentationer kun adskille sig fra hinanden i den del, som beskriver det pågældende område. Skal man transmittere to sådanne billeder efter hinanden, kan man nøjes med at transmittere det ene og derefter transmittere differensen mellem de to billeders repræsentationer. Denne differens vil være 0 alle de steder, billederne er ens. Hvis man tænker på en sekvens af fjernsynsbilleder, er det ikke svært at forestille sig, at man kan spare meget transmission på denne måde.

Den ortonormalbasis, som vi arbejder hen imod, er til en hvis grad lokaliseret i *både* rum og frekvens, omend ikke fuldstændigt.

Hvis man betragter den euklidiske basis, ses det, at basen er på formen $\{R_k u\}_{k=0}^{N-1}$, hvor $u = e_1$ og R_k betegner en translation med k pladser. Man kan desværre let vise, at en sådan ortonormalbasis aldrig kan være lokaliseret i frekvens (Frazier; 1999, side 170). Det viser sig imidlertid (hvorfor ligger uden for dette speciales problemstilling), at man med relativt simple midler alligevel kan få en basis med de ønskede egenskaber.

Vi vil herfra udelukkende betragte $\mathbb{C}^N$ med $N = 2M$, altså rum med en lige dimension. Når vi i det følgende skriver $\mathbb{C}^N$, er det altså underforstået, at $N = 2M$, $M \in \mathbb{N}$.

Definition 3.4
Lad $u, v \in \mathbb{C}^N$. En ortonormalbasis for $\mathbb{C}^N$ på formen

$$\{R_{2k}u\}_{k=0}^{M-1} \cup \{R_{2j}v\}_{j=0}^{M-1}$$

kaldes en førstetrins waveletbasis *for $\mathbb{C}^N$.*

En førstetrins waveletbasis består altså af to vektorer u, v samt alle vektorer, som fremkommer ved en lige translation af u og v. Definitionen giver ingen kriterier for, hvornår de translaterede til u og v udgør en basis. Sådanne kriterier gives i det følgende.

Definition 3.5
Lad $u, v \in \mathbb{C}^N$. Definér for hvert $n \in \mathbb{Z}$ matricen $A(n)$ kaldet systemmatricen *for u og v ved*

$$A(n) = \frac{1}{\sqrt{2}} \begin{pmatrix} \hat{u}(n) & \hat{v}(n) \\ \hat{u}(n+M) & \hat{v}(n+M) \end{pmatrix}.$$

Systemmatricen gør det muligt at undersøge, hvornår de translaterede til u og v udgør en ortonormalbasis.

Sætning 3.6
Lad $u, v \in \mathbb{C}^N$.

$$B = \{R_{2k}v\}_{k=0}^{M-1} \cup \{R_{2k}u\}_{k=0}^{M-1}$$

udgør en ortonormalbasis for $\mathbb{C}^N$, hvis og kun hvis systemmatricen $A(n)$ for u og v for hvert $n = 0, \ldots, M-1$ er unitær[1].

For en 2×2-matrix over $\mathbb{C}$ er kravet om, at matricen skal være unitær ensbetydende med, at matricens søjler skal udgøre en ortonormalbasis for $\mathbb{C}^2$ (Frazier; 1999, side 90). Fra definitionen af systemmatricen giver ortogonalitetskravet for søjlerne, at

$$\hat{u}(n)\overline{\hat{v}(n)} + \hat{u}(n+M)\overline{\hat{v}(n+M)} = 0.$$

At søjlerne er normeret til 1 betyder grundet faktoren foran matricen i definition 3.5, at

$$\begin{aligned} |\hat{u}(n)|^2 + |\hat{u}(n+M)|^2 &= 2, \\ |\hat{v}(n)|^2 + |\hat{v}(n+M)|^2 &= 2, \end{aligned}$$

hvorfor sætning 3.6 kan formuleres således:

1 *En matrix A er unitær netop hvis $A^{-1} = A^*$, hvor A^* er den adjungerede til A.*

Sætning 3.7
Lad $u, v \in \mathbb{C}^N$.

$$B = \{R_{2k}v\}_{k=0}^{M-1} \cup \{R_{2k}u\}_{k=0}^{M-1}$$

udgør da en førstetrins waveletbasis for $\mathbb{C}^N$*, hvis og kun hvis*

$$\begin{aligned} |\hat{u}(n)|^2 + |\hat{u}(n+M)|^2 &= 2, \\ |\hat{v}(n)|^2 + |\hat{v}(n+M)|^2 &= 2 \end{aligned}$$

og

$$\hat{u}(n)\overline{\hat{v}(n)} + \hat{u}(n+M)\overline{\hat{v}(n+M)} = 0$$

for hvert $n = 0, \ldots, M-1$.

Bevis
Sætning B.10 i appendiks. □

Vektorerne u og v i ovenstående siges at *generere* basen B.

3.1.1 Haar-basen

I dette underafsnit vil vi betragte en særlig førstetrins waveletbasis, kaldet *Haar*-basen. Haar-basens funktioner er meget simple, og basen er derfor ideel til illustration af de begreber, vi betragter. Iøvrigt spiller Haar-basen en rolle for fremstillingen af ICER i kapitel 5 i HOVEDRAPPORTEN.

Lad $u, v \in \mathbb{C}^N$ være defineret ved

$$u = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right)$$

og

$$v = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right).$$

For $N = 2$ er det let at se, at u og v udgør en ortonormalbasis. For $N \geq 4$ bestemmes matricens elementer. For $n = 0$:

$$\begin{aligned} \hat{u}(0) &= \sum_{k=0}^{N-1} u(k) \cdot 1 = u(0) + u(1) = \sqrt{2}. \\ \hat{u}(M) &= \sum_{k=0}^{N-1} u(k)e^{-2\pi iMk/N} = \sum_{k=0}^{N-1} u(k)e^{-\pi ik} = u(0) - u(1) = 0. \\ \hat{v}(0) &= \sum_{k=0}^{N-1} v(k) \cdot 1 = v(0) + v(1) = 0 \\ \hat{v}(M) &= \sum_{k=0}^{N-1} v(k)e^{-2\pi iMk/N} = \sum_{k=0}^{N-1} v(k)e^{-\pi ik} = v(0) - v(1) = \sqrt{2}. \end{aligned}$$

Systemmatricen for u og v med $n = 0$ bliver altså

$$A(0) = \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

som tydeligvis er unitær. Ved en tilsvarende udregning fås, at systemmatricen er unitær for $n = 1, \ldots, M-1$.

Definition 3.8 (Haar-basen)
Lad $u, v \in \mathbb{C}^N$ være defineret ved

$$u = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right)$$

og

$$v = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right).$$

Så udgør

$$B = \{R_{2k}v\}_{k=0}^{M-1} \cup \{R_{2k}u\}_{k=0}^{M-1}$$

en førstetrins waveletbasis for $\mathbb{C}^N$. Denne basis kaldes Haar-*basen på første trin.*

Fra definitionen af u og v er det tydeligt, at de er lokaliseret i rum. Vektorernes Fouriertransformerede er givet ved:

$$\begin{aligned}
\hat{u}(m) &= \sum_{n=0}^{N-1} u(n)e^{-2\pi imn/N} = \frac{1}{\sqrt{2}}\left(1 + e^{-2\pi im/N}\right), \\
\hat{v}(m) &= \sum_{n=0}^{N-1} v(n)e^{-2\pi imn/N} = \frac{1}{\sqrt{2}}\left(1 - e^{-2\pi im/N}\right).
\end{aligned}$$

Realdelen af $\hat{u}$ og $\hat{v}$ bliver da

$$\begin{aligned}
\mathrm{Re}(\hat{u})(m) &= \frac{1}{\sqrt{2}}\left(1 + \cos(2\pi\frac{m}{N})\right), \\
\mathrm{Re}(\hat{v})(m) &= \frac{1}{\sqrt{2}}\left(1 - \cos(2\pi\frac{m}{N})\right).
\end{aligned}$$

Fra ovenstående kan vi se, at når m er tæt på 0 eller N, så er $\mathrm{Re}(\hat{u})(m)$ relativt stor, mens $\mathrm{Re}(\hat{u})(m)$ er relativt lille, når m er tæt på $N/2$. For $\mathrm{Re}(\hat{v})(m)$ forholder det sig omvendt. Når vi kun ser på realdelene af u og v gælder derfor (se eventuelt appendiks A side 120) at u er koncentreret omkring de lave frekvenser, mens v er koncentreret omkring de høje frekvenser. Haar-basens lokalisering i frekvens er altså begrænset til en høj/lav opdeling.

Definition 3.9
Definér P ved

$$P(z) = \sum_{m=0}^{M-1} \langle z, R_{2m}u\rangle R_{2m}u.$$

Vektoren P kaldes da den partielle rekonstruktion af z.

Den partielle rekonstruktion $P(z)$ opfylder

$$\begin{aligned}
P(z)(2j) &= \sum_{m=0}^{M-1} \langle z, R_{2m}u\rangle (R_{2m}u)(2j) \\
&= \sum_{m=0}^{M-1} \left(\sum_{k=0}^{N-1} z(k) R_{2m}u(k) \right) R_{2m}u(2j) \\
&= \sum_{m=0}^{M-1} \left(z(2m) + z(2m+1)\frac{1}{\sqrt{2}} \right) R_{2m}u(2j) \\
&= \left((z(2j) + z(2j+1))\frac{1}{\sqrt{2}} \right) R_{2j}u(2j) \\
&= \left((z(2j) + z(2j+1))\frac{1}{\sqrt{2}} \right) \frac{1}{\sqrt{2}} \\
&= \frac{1}{2}\left(z(2j) + z(2j+1) \right).
\end{aligned}$$

For $P(2j+1)$ fås samme resultat. Vektoren $P(z)$'s koordinater er altså lig de parvise gennemsnit af z's koordinater. Det giver intuitivt god mening, at de vektorer, som er koncentreret omkring de lave frekvenser, det vil sige $R_{2m}u$, indeholder information om de gennemsnitlige værdier af z.

De vektorer, som er koncentreret omkring det højfrekvente bidrager naturligvis også. Definér $Q(z)$ ved

$$Q(z) = \sum_{m=0}^{M-1} \langle z, R_{2m}v\rangle R_{2m}v.$$

Da B er en ortonormalbasis for $\mathbb{C}^N$ gælder naturligvis

$$z = P(z) + Q(z). \tag{3.1}$$

$Q(z)$ indeholder altså differensen mellem det oprindelige z og $P(z)$.

Hvis vi i et kort øjeblik tænker på z som en række af pixelværdier i et billede, kan vektoren P opfattes som z betragtet i en opløsning, der er halvt så god som den oprindelige. Vektoren Q indeholder så den information, som gør det muligt at transformere fra den 'dårlige' opløsning til den oprindelige.

Eksempel 3.10
Lad $z \in \mathbb{C}^8$ være vektoren givet ved

$$z = (48, 48, 40, 16, 56, 40, 8, 24).$$

Lad $[z]_H$ være z's fremstilling i forhold til Haar-basen H som defineret ovenfor. Så er

$$[z]_H = \frac{1}{\sqrt{2}}(0, 24, 16, -16, 96, 56, 96, 32),$$

$P(z)$ bliver

$$P(z) = (48, 48, 28, 28, 48, 48, 16, 16)$$

og $Q(z)$ bliver

$$Q(z) = (0, 0, 12, -12, 8, -8, -8, 8).$$

Bemærk, at ligning (3.1) er opfyldt. Bemærk iøvrigt, at halvdelen af P's og Q's koordinater egentlig er overflødige i forhold til at transformere fra den dårlige opløsning til den oprindelige. Hvis vi kalder den n'te komponent i P og Q for p_n henholdsvis q_n gælder nemlig, at

$$z = (p_0 + q_0, p_0 - q_0, p_2 + q_2, p_2 - q_2, \ldots, p_6 + q_6, p_6 - q_6).$$

Denne oplysning gemmes til senere. ◇

Inden vi fortsætter fremstillingen skal vi bemærke endnu et par egenskaber ved Haar-transformationen. I nedenstående er et signal en vektor over de reelle tal.

Definition 3.11
Lad z være et signal. Energien $\mathcal{E}$ *af z defineres ved*

$$\mathcal{E} = \|z\|.$$

Hvis vi betragter eksemplet ovenfor, så gælder at

$$\|z\| = \|[z]_H\|.$$

For Haar-transformationen er ovenstående opfyldt for alle $z \in \mathbb{R}^N$, hvorfor Haar-transformationen siges at *bevare energien*. For z's vedkommende gælder, at de første 4 koordinater indeholder omtrent 55% af energien, mens de sidste fire koordinater indeholder 45%. For $[z]_H$ indeholder de første fire koordinater omtrent 5% af energien, mens de sidste fire koordinater indeholder 95%. Som vi så tidligere svarer de første fire koordinater til højfrekvente bidrag og de sidste fire til de lavfrekvente bidrag. Ovenstående beregning af energifordelingen illustrerer en meget vigtig egenskab ved Haar-transformationen, nemlig at den koncentrerer energien (i det lavfrekvente område). Dette kaldes *'compaction of energy'*. (Walker; 1999, side 7)

3.2 Filterbanks

I dette afsnit vil vi behandle basisskiftet fra den euklidiske basis ind i en waveletbasis ved at betragte filtre. Filtertilgangen inspirerer til at konstruere waveletbaser med nogle interessante egenskaber.

Definition 3.12
Lad T_b være en foldningsoperator. Så kaldes T_b også for et filter.

Filtre kommer til at spille en stor rolle i vores tilgang til wavelets. Hvis T_b er et filter vil vi kalde påføringen af T_b på z for en filtrering af z med hensyn til filteret T_b.

Lad $z \in \mathbb{C}^N$, og lad B være en førstetrins waveletbasis for $\mathbb{C}^N$. Ifølge lemma B.3 i appendiks gælder, at

$$\langle z, R_{2k} v \rangle = (z * \tilde{v})(2k).$$

$z's$ koordinater i forhold til B kan derfor skrives som

$$[z]_B = \begin{pmatrix} z * \tilde{v}(0) \\ z * \tilde{v}(2) \\ \vdots \\ z * \tilde{v}(N-2) \\ z * \tilde{u}(0) \\ z * \tilde{u}(2) \\ \vdots \\ z * \tilde{u}(N-2) \end{pmatrix}.$$

Basisskiftet kan altså opfattes som filtrering af z med hensyn til filtrene T_u og T_v efterfulgt af en sammenstilling af filtreringerne.

Fra ovenstående er det klart, at kun pladser i u og v med et lige indeks er interessante for basisskiftet. Denne observation er motivationen for det følgende.

Definition 3.13
Definér $D : \mathbb{C}^N \to \mathbb{C}^M$ *ved*

$$D(z)(n) = z(2n), \qquad \text{for } n = 0, \ldots, M-1.$$

D *kaldes da* downsampling-operatoren.

Downsampling af z svarer altså til at smide z's ulige koordinater væk. Fra definitionen af operatoren kan vi se, at z's koordinater i B-basen kan skrives som

$$[z]_B = \begin{pmatrix} D(z * \tilde{v}) \\ D(z * \tilde{u}) \end{pmatrix}.$$

Et basisskift som det, der er foretaget ovenover er kendetegnende for en *filterbank*. Generelt defineres en filterbank[2] som en følge af foldninger samt andre operationer, ovenfor i form af downsampling. Basisskiftet fra den euklidiske basis til waveletbasen kaldes *analysen*.

Efter analysen finder selve datakomprimeringen sted. Efter dekomprimering skal der foretages et basisskift tilbage til den euklidiske basis. Dette basisskift kaldes *syntesen*.

At et basisskift kan beregnes ved hjælp af foldninger betyder, at det kan foretages ved relativt få beregninger sammenlignet med et basisskift foretaget ved multiplikation med en matrix. Dette diskuteres kort i (Frazier; 1999, side 183-184). Som beskrevet i kapitel 2 er den processorkraft, der er til rådighed på MER-roverne stærkt begrænset, og af denne årsag har beregningskompleksitet i praksis meget at sige. Det er derfor ønskværdigt at kunne foretage basisskiftet tilbage på tilsvarende vis.

Downsampling har oplagt en modsat operation:

Definition 3.14
Definér $U : \mathbb{C}^M \to \mathbb{C}^N$ *ved*

$$U(z)(n) = \begin{cases} z(n/2) & \text{for } n \text{ lige} \\ 0 & \text{for } n \text{ ulige} \end{cases}$$

U *kaldes da* Upsampling-operatoren.

2 Se (Frazier; 1999, side 183).

Upsampling foregår altså ved efter hvert af z's koordinater at indsætte et 0.

Fra definitionerne af up- og downsampling er det klart, at $D \circ U = I$, men at den omvendte sammensætning ikke nødvendigvis giver identiteten. Fra definition B.5 i appendiks ses nemlig, at

$$(U \circ D)(z) = \frac{1}{2}(z + z^*).$$

Det er derfor ikke umiddelbart klart, at det er muligt at foretage et basisskift tilbage til den euklidiske basis ved hjælp af disse filtre. I næste afsnit skal vi se, hvornår dette er tilfældet.

3.2.1 Perfekt rekonstruktion

Som nævnt svarer en filterbank til et basisskift, i vores tilfælde ind i en førstetrins waveletbasis B (vi vil her tillade os at kalde z for inddata til filterbanken og $[z]_B$ for uddata fra filterbanken). Vi så også, at

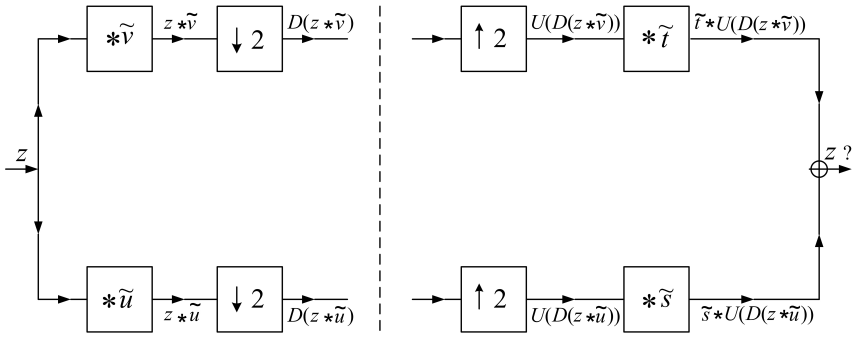
$$[z]_B = \begin{pmatrix} D(z * \tilde{v}) \\ D(z * \tilde{u}) \end{pmatrix}.$$

Den filterbank, som skal foretage skiftet tilbage til den euklidiske basis, vil vi lade arbejde separat på $D(z * \tilde{u})$ og $D(z * \tilde{v})$. Første skridt i den anden filterbank bliver først at upsample henholdsvis $D(z * \tilde{u})$ og $D(z * \tilde{v})$ for derefter at foretage en foldning med vektorer $\tilde{s}$ og $\tilde{t}$. Uddata fra den anden filterbank fastlægges til sidst ved summen af disse foldninger. Dette er fremstillet skematisk på figur 3.1.

Et naturligt spørgsmål er nu, hvornår der gælder

$$z = \tilde{s} * U(D(z * \tilde{u})) + \tilde{t} * U(D(z * \tilde{v})).$$

Hvis dette er opfyldt, siges filterbanken at give *perfekt rekonstruktion*. Følgende sætning besvarer spørgsmålet.



Figur 3.1 Figuren viser to på hinanden følgende filterbanks. Operationerne $\uparrow 2$ og $\downarrow 2$ symboliserer up- og downsampling. Symbolet $\oplus$ yderst til højre angiver, at de to grenes uddata lægges sammen. Den stiplede linie adskiller de to filterbanks. (Frazier; 1999, side 185)

Sætning 3.15 (Perfekt rekonstruktion)
Lad $u, v, s, t \in \mathbb{C}^N$. Lad $A(n)$ for hvert $n = 0, \ldots, N-1$ være systemmatricen for u og v. Så er der perfekt rekonstruktion, det vil sige

$$z = \tilde{s} * U(D(z * \tilde{u})) + \tilde{t} * U(D(z * \tilde{v})) \quad \forall z \in \mathbb{C}^N,$$

hvis og kun hvis

$$A(n) \begin{pmatrix} \hat{s}(n) \\ \hat{t}(n) \end{pmatrix} = \begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix}$$

for alle $n = 0, \ldots, N-1$. Hvis $A(n)$ er unitær simplificerer dette til $t = \tilde{v}$ og $s = \tilde{u}$.

Bevis
Sætning B.15 i appendiks. □

Der skal her knyttes en bemærkning til sætningen. Den betingelse, der angives som tilstrækkelig for at sikre perfekt rekonstruktion, omfatter ikke et krav om at systemmatricen skal være unitær. Sætningen er bredere og lover altså perfekt rekonstruktion, selvom $B = \{R_{2k}v\}_{k=0}^{M-1} \cup \{R_{2k}u\}_{k=0}^{M-1}$ ikke udgør en ortonormalbasis for $\mathbb{C}^N$. Indtil videre vil vi dog kun interessere os for det specielle tilfælde, hvor B *er* en ortonormalbasis. Af samme årsag vil vi herfra lade $\tilde{u}$ og $\tilde{v}$ betegne filtre i analysen og kalde de tilhørende filtre i syntesen for u og v.

Et typisk princip for billedkomprimering ved brug af wavelets er nu antydet; en *analyse*, efterfulgt af *komprimering*[3] og afsluttet af en *syntese*. Analysen udføres af en filterbank bestående af et filter fulgt af en downsampling. Syntesen består af en upsampling efterfulgt af et filter. Sætning 3.15 fortæller, hvornår de to filterbanks giver perfekt rekonstruktion. Dette vil vi forfølge yderligere.

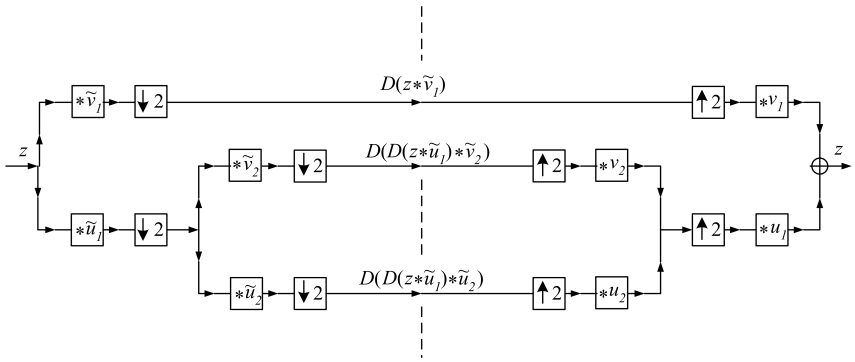
3.3 *p*'te-trins waveletbasis

I dette afsnit vil vi gennem iteration opbygge mere komplekse filterbanks. Den grundlæggende strategi er at vælge en gren i diagrammet over den filterbank, som foretager analysen (se figur 3.1) og lade uddata fra denne gren gennemgå endnu en filtrering og en downsampling. Til at modsvare dette tilføjes et filter og en upsampling til den tilhørende gren i den filterbank, som foretager syntesen. Dette er illustreret på figur 3.2.

De tilføjede filtre vælges, så betingelserne i sætning 3.15 er opfyldt, det vil sige, så den perfekte rekonstruktion bevares. Processen kan itereres. Hver gang tilføjes endnu en forgrening til den *nederste* gren i analysen såvel som syntesen. I løbet af afsnittet vil det blive klart, at den resulterende filterbank på analysesiden kommer til at svare til et basisskift til en basis kaldet en *p'te-trins waveletbasis*, hvor p angiver antallet af iterationer i den ovenfor beskrevne process. Principielt kan en hvilken som helst gren itereres på denne vis, men i den sædvanlige wavelet-analyse vælges kun én gren (Frazier; 1999, side 196).

Definition 3.16
Lad N være delelig med 2^p, $p \in \mathbb{N}$. En p'te-trins waveletfilter-følge *er en*

3 Faktisk deles komprimeringen op i flere trin. Dette skal vi se i de næste kapitler.



Figur 3.2 Figuren viser de udvidede filterbanks. Nederste gren har fået tilføjet endnu et filter både i analysen og i syntesen. (Frazier; 1999, side 198)

følge af vektorer $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ *sådan at der for hvert* $\ell = 1, \ldots, p$ *gælder*

$$u_\ell, v_\ell \in \mathbb{C}^{N/2^{\ell-1}},$$

og at systemmatricen $A_\ell(n)$ *defineret ved*

$$A_\ell(n) = \frac{1}{\sqrt{2}} \begin{pmatrix} \hat{u}_\ell(n) & \hat{v}_\ell(n) \\ \hat{u}_\ell(n + N/(2^{\ell-1})) & \hat{v}_\ell(n + N/(2^{\ell-1})) \end{pmatrix}$$

er unitær for alle $n = 0, \ldots, N/(2^{\ell-1})$. *Definér*

$$\begin{aligned} x_1 &= D(z * \hat{v}_1) \in \mathbb{C}^{N/2}, \\ y_1 &= D(z * \hat{u}_1) \in \mathbb{C}^{N/2}. \end{aligned}$$

Definér endvidere ved induktion vektorerne

$$\begin{aligned} x_\ell &= D(y_{\ell-1} * \hat{v}_\ell) \in \mathbb{C}^{N/2^\ell}, \\ y_\ell &= D(y_{\ell-1} * \hat{u}_\ell) \in \mathbb{C}^{N/2^\ell}. \end{aligned}$$

for $\ell = 2, \ldots, p$.

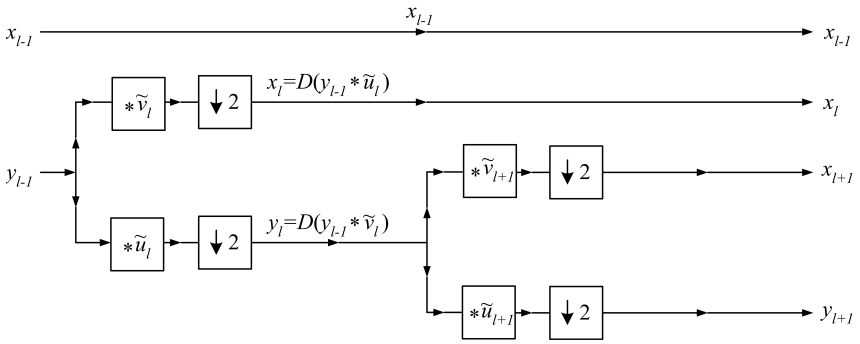
Uddata fra en p'te-trins waveletfilter-følge er mængden af vektorer

$$\{x_1, x_2, \ldots, x_p, y_p\}.$$

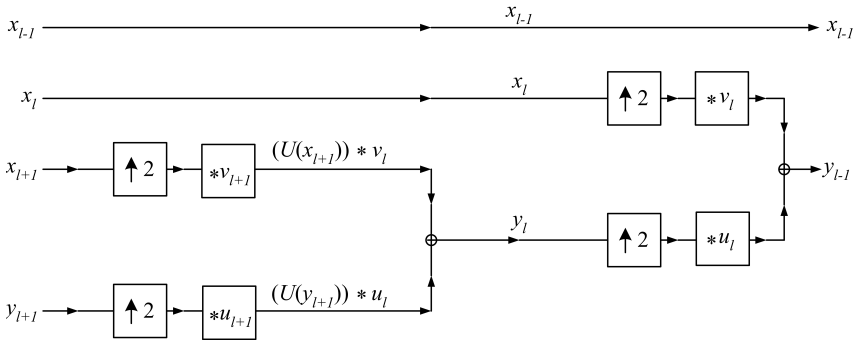
Bemærk, at kun det sidste y er interessant i denne forbindelse. For hvert $\ell = 1, \ldots, p$ gælder, at x_ℓ er resultatet af ℓ på hinanden følgende filtreringer og downsamplinger. På figur 3.3 og 3.4 ses udsnit af analysen henholdsvis syntesen.

Et hovedresultat i dette afsnit og kapitel er at vise, at alle disse filtreringer og samplinger kan foretages i ét snuptag. Vi skal derfor forsøge at fjerne induktionen fra definitionen af uddata fra en p'te-trins waveletfilter-følge.

I det følgende vil vi begrænse os yderligere og kun se på N, som er delelige med 2^p. Desuden vil vi indføre notationen $U^\ell(z)$ og $D^\ell(z)$ for henholdsvis up- og downsampling ℓ gange.



Figur 3.3 Figuren illustrerer, hvordan x_ℓ, y_ℓ optræder i analysefasen af filtreringen. (Frazier; 1999, side 200)



Figur 3.4 Figuren illustrerer, hvordan x_ℓ, y_ℓ optræder i syntesefasen af filtreringen. (Frazier; 1999, side 201)

Definition 3.17
Lad $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ opfylde

$$u_\ell, v_\ell \in \mathbb{C}^{N/(2^{\ell-1})},$$

for $\ell = 1, \ldots, p$. Lad $f_1 = v_1$, $g_1 = u_1$ og definér ved induktion $f_\ell, g_\ell \in \mathbb{C}^N$ ved

$$\begin{aligned} f_\ell &= g_{\ell-1} * U^{\ell-1}(v_\ell), \\ g_\ell &= g_{\ell-1} * U^{\ell-1}(u_\ell). \end{aligned}$$

Formålet med ovenstående definition er som vi om lidt skal se, at flytte induktionen fra selve filterbankens opbygning over til de filtre, som anvendes. Fra definitionen af f_ℓ og g_ℓ ses, at g_ℓ fremkommer efter ℓ foldninger med stadigt mere upsamplede udgaver af u_ℓ. Vektoren f_ℓ fremkommer på samme vis, dog involverer den sidste foldning ikke u_ℓ, men i stedet v_ℓ.

Lemma 3.18
Lad $z \in \mathbb{C}^N$ og lad $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ være som i definition 3.17. De-

finér $x_1, \ldots x_p, y_1, \ldots, y_p$ *samt* $f_1, \ldots f_p, g_1, \ldots, g_p$ *som tidligere. Så gælder for* $\ell = 1, \ldots, p$*, at*

$$\begin{aligned} x_\ell &= D^\ell(z * \tilde{f}_\ell) \\ y_\ell &= D^\ell(z * \tilde{g}_\ell) \end{aligned}$$

Bevis
Lemma B.20 i appendiks. □

Nu er vi klar til første vigtige resultat. Følgende sætning fortæller, hvordan uddata afhænger af inddata i den filterbank, som hører til syntesen, det vil sige rekonstruktionen.

Sætning 3.19
Lad $u_1, v_1, \ldots, u_p, v_p$ *for hvert* $\ell = 1, \ldots, p$ *opfylde, at*

$$u_\ell, v_\ell \in \mathbb{C}^{N/2^{\ell-1}}.$$

Definér f_ℓ *og* g_ℓ *som ovenfor. Hvis inddata til den* ℓ*'te gren i syntesen er* x_ℓ*, og alle andre input er* 0*, så er uddata fra syntesen lig*

$$f_\ell * U^\ell(x_\ell).$$

Hvis inddata til den sidste gren er y_p*, og alle andre inddata er* 0*, så er uddata fra syntesen lig*

$$g_p * U^p(y_p).$$

Bevis
Sætning B.22 i appendiks. □

Vi opsummerer: Lemma 3.18 tillader os at opskrive uddata fra analysen uden brug af induktion. Den ℓ'te grens uddata fås ved filtrering med f_ℓ efterfulgt af downsampling ℓ gange. Den sidste grens uddata fås ved filtrering med g_p efterfulgt af downsampling p gange. Sætning 3.19 giver et tilsvarende resultat for syntesen. Samlet har vi altså, at vores oprindelige filtre kan erstattes med filtre, som ikke indeholder induktion. Disse er vist på figur 3.5.

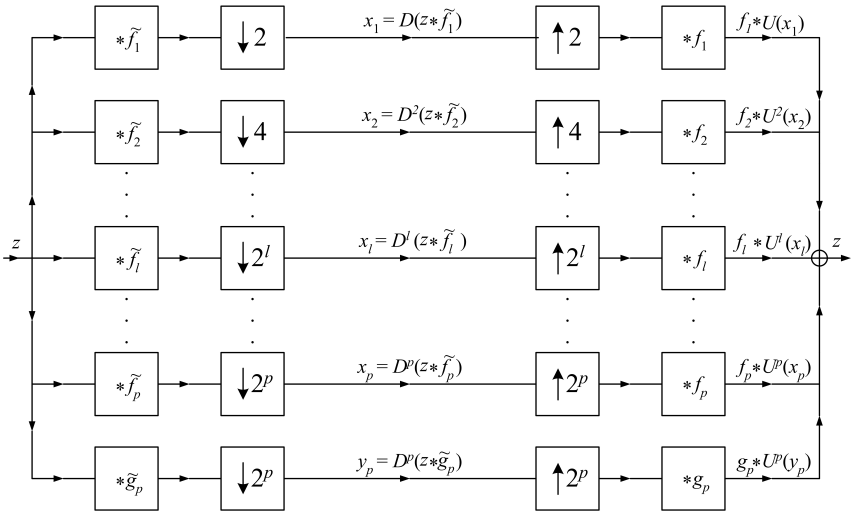
Betragt uddata fra den ℓ'te gren i analysen, x_ℓ. Om x_ℓ gælder ifølge sætning B.21 i appendiks

$$x_\ell(k) = \langle z, R_{2^\ell k} f_\ell \rangle. \tag{3.2}$$

for $k = 0, \ldots, N/(2^\ell - 1)$. For den sidste gren har vi tilsvarende

$$y_p(k) = \langle z, R_{2^p k} g_p \rangle. \tag{3.3}$$

Bemærk iøvrigt, at det samlede antal af komponenter i x_ℓ'erne samt y_p er lig N. Dette kan indses ved at kigge på figur 3.5, men er også vist i sætning B.17 i appendiks. Sammen med ovenstående antyder dette, at komponenterne angiver z's koordinater i forhold til en ortonormalbasis for $\mathbb{C}^N$. Dette motiverer følgende definition:



Figur 3.5 Figuren viser den nye filterbank, som ikke indeholder gentagne filtreringer. Alle grene indeholder nu kun et filter i analysen og et filter i syntesen. (Frazier; 1999, side 208)

Definition 3.20
Lad $f_1, \ldots, f_p, g_p \in \mathbb{C}^N$*, og lad* B *være på formen*

$$\{R_{2k} f_1\}_{k=0}^{N/2-1} \cup \{R_{4k} f_2\}_{k=0}^{N/4-1} \cup \ldots \cup \{R_{2^p k} f_p\}_{k=0}^{N/2^p-1} \cup \{R_{2^p k} g_p\}_{k=0}^{N/2^p-1}.$$

Hvis B *udgør en ortonormalbasis for* $\mathbb{C}^N$*, kaldes* B *en* p'te-trins wavelet-basis. *Vektorerne* $f_1, \ldots, f_p, g_p \in \mathbb{C}^N$ *siges at generere* B*.*

Med denne definition er vi nået frem til kapitlets hovedresultat.

Sætning 3.21
Lad N *være delelig med* 2^p *og lad* $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ *udgøre en* p*'te-trins waveletfilter-følge. Definer* $f_1, \ldots, f_p, g_p$ *som i definition 3.17. Så genererer* $f_1, \ldots, f_p, g_p$ *en* p*'te-trins waveletbasis for* $\mathbb{C}^N$*.*

Bevis
Sætning B.26 i appendiks. □

Ovenstående resultat formuleres til tider således:

Definition 3.22
Lad $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ *udgøre en* p*'te-trins waveletfilter-følge. Definer* $f_1, \ldots, f_p, g_p$ *som i definition 3.17 og definer*

$$\begin{aligned} \psi_{-j,k} &= R_{2^j k} f_j, \\ \phi_{-j,k} &= R_{2^j k} g_j. \end{aligned}$$

Den ovenfor beskrevne ortonormalbasis for $\mathbb{C}^N$ *har da formen*

$$\{\psi_{-1,k}\}_{k=0}^{(N/2)-1} \cup \{\psi_{-2,k}\}_{k=0}^{(N/4)-1} \cup \ldots \cup \{\psi_{-p,k}\}_{k=0}^{(N/2^p)-1} \cup \{\phi_{-p,k}\}_{k=0}^{(N/2^p)-1}.$$

Elementerne i denne basis kaldes wavelets *på* $\mathbb{C}^N$*.*

Vi opsummerer endnu en gang: Den gentagne filtrering, som vi har beskrevet, kan i kraft af sætning 3.21 opfattes som et basisskift fra den euklidiske basis til en ortonormal waveletbasis.

3.3.1 Haar-basen

Vi vil her vende tilbage til Haar-basen fra afsnit 3.1.1 og lade den undergå de forandringer, som er beskrevet indtil nu. Resultaterne, som gengives her, er udledt i afsnit B.2 i appendiks. Lad $u_\ell, v_\ell \in \mathbb{C}^{N/(2^{\ell-1})}$ være defineret ved

$$\begin{aligned} u_\ell &= \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right), \\ v_\ell &= \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right). \end{aligned}$$

Systemmatricen $A_\ell(n)$ for u_ℓ, v_ℓ ses at blive lig systemmatricen $A(n)$ fra afsnit 3.1.1 for $\ell = 1, \ldots, p$. Mængden

$$\{u_1, v_1, u_2, v_2, \ldots, u_p, v_p\}$$

udgør altså en p'te-trins waveletfilter-følge og med udgangspunkt i denne kan vi derfor opstille en p'te-trins waveletbasis.

Definition 3.23 (Haar-basen)
Lad $\psi_{-\ell,k}, \phi_{-\ell,k}$ være defineret ved

$$\psi_{-\ell,k}(n) = \begin{cases} 2^{-\ell/2}, & n = 2^\ell k, 2^\ell k + 1, \ldots, 2^\ell k + 2^{\ell-1} - 1 \\ -2^{-\ell/2}, & n = 2^\ell k + 2^{\ell-1}, \ldots, 2^\ell k + 2^\ell - 1 \\ 0, & n = 0, 1, \ldots, 2^\ell k - 1, 2^\ell k + 2^\ell, \ldots, N-1 \end{cases},$$

$$\phi_{-\ell,k}(n) = \begin{cases} 2^{-\ell/2}, & n = 2^\ell k, 2^\ell k + 1, \ldots, 2^\ell k + 2^\ell - 1 \\ 0, & n = 0, 1, \ldots, 2^\ell k - 1, 2^\ell k + 2^\ell, \ldots, N-1 \end{cases}.$$

Så udgør

$$\{\psi_{-1,k}\}_{k=0}^{(N/2)-1} \cup \{\psi_{-2,k}\}_{k=0}^{(N/4)-1} \cup \ldots \cup \{\psi_{-p,k}\}_{k=0}^{(N/2^p)-1} \cup \{\phi_{-p,k}\}_{k=0}^{(N/2^p)-1}$$

en p'te-trins waveletbasis for $\mathbb{C}^N$. Denne basis kaldes Haar-*basen på p'te trin.*

I eksempel 3.10 betragtede vi den partielle rekonstruktion af z. Den vil vi definere endnu en gang.

Definition 3.24
Definer P_{-j} ved

$$P_{-j}(z) = \sum_{k=0}^{N/(2^j)-1} \langle z, \phi_{-j,k} \rangle \phi_{-j,k}.$$

Vektoren P kaldes da den partielle rekonstruktion af z på niveau j.

Ved udregning fås, at for $2^j k \leq n \leq 2^j k + 2^j$ er

$$(P_{-j}(z))(n) = \frac{1}{2^j}\left(z(2^j k) + z(2^j k) + 1 + \ldots + z(2^j k + 2^j - 1)\right).$$

De første 2^j koordinater i P_{-j} er altså lig gennemsnittet af de første 2^j koordinater i z, de næste 2^j koordinater i P_{-j} er lig gennemsnittet af de næste 2^j koordinater i z og så videre. Vektoren P_{-j} svarer altså til $P(z)$ som tidligere defineret, blot benyttes 2^j koordinater fra z til at udregne gennemsnit. På tilsvarende vis defineres Q_{-j}:

$$Q_{-j}(z) = \sum_{k=0}^{N/(2^j)-1} \langle z, \psi_{-j,k} \rangle \psi_{-j,k},$$

og der gælder

$$P_{-j+1}(z) = P_{-j}(z) + Q_{-j}(z).$$

Hvis vi igen tænker på z som en række pixelværdier i et billede, så svarer P_{-j} til en opløsning, i hvilken 2^j pixel tildeles den samme værdi bestemt ved deres gennemsnit. I Q_{-j} findes så den information, som gør det muligt at transformere fra én opløsning til en opløsning, der er dobbelt så god.

P og Q illustrerer en vigtig egenskab ved Haar-transformationen. Denne egenskab er kendetegnende for de fleste wavelettransformationer, nemlig at de giver mulighed for at analysere et signal ved forskellige opløsninger eller detaljeniveauer.

Med disse ord vil vi forlade den teoretiske diskussion af wavelets og begive os over i en potentiel anvendelse af disse.

3.4 Billedkomprimering med wavelets

Vi har i det foregående vist, at et signal kan repræsenteres ved en waveletbasis, og vi har som eksempel givet Haar-basen. Indtil nu har vi ikke interesseret os for, hvad basen egentlig skal bruges til. Det vil vi i dette afsnit. Vi vil tage udgangspunkt i de problemer, som Fouriertransformationen giver anledning til, når den anvendes til billedkomprimering og diskutere wavelets med udgangspunkt i disse problemer.

Et billede opfattes typisk som en matrix M med pixelværdier, der normalt er heltallige. Ved en diskret cosinus-transformation[4] (DCT) foretaget først på hver række og derefter på hver søjle i matricen bliver matricens oplysninger om pixelværdier transformeret til oplysninger om billedets frekvenser. Denne procedure udgør en af grundpillerne i den velkendte JPEG-standard for billedkomprimering.

Ved den diskrete cosinus-transformation opløses billedet i todimensionale harmoniske svingninger med varierende frekvenser, og den transformerede matrix F indeholder oplysninger om disse svingningers amplitude.

En simpel metode til komprimering af billedet med tab er nu at fjerne alle svingninger, hvis amplitude er mindre end en fastsat grænse. Dette gøres simpelthen ved at sætte alle værdier i F lig 0, hvis de er mindre end grænsen.

Da energien i et billede ofte er koncentreret i det lavfrekvente område, svarer ovenstående typisk til at fjerne en del af de høje frekvenser. Det menneskelige syn er ligesom den menneskelige hørelse ikke så følsom overfor høje frekvenser, og elimineringen af de høj frekvenser påvirker derfor

4 En cosinus-transformation er tilstrækkelig, da pixelværdierne ikke er komplekse.

ikke billedet på en måde, som er synlig for det menneskelige øje. Denne form for komprimering omfatter naturligvis et tab af information, men kan opfattes som *visuel* tabsfri. (Forchhammer; 2004, side 12 i INTERVIEWS).

Matricen F', som er resultatet af komprimeringen, repræsenterer en komprimeret version af billedet. Ved visning af billedet skal F' inverstransformeres. Ved denne transformation opnås en matrix M' som udgør en tilnærmelse til det oprindelige billede M. Matricen F' vil typisk indeholde en stor mængde nuller. Sådanne data kan effektivt komprimeres med en af de teknikker, som præsenteres i kapitel 5.

Denne form for billedkomprimering har tidligere været meget udbredt, da den er forholdsvis simpel og giver gode komprimeringsresultater. Den udgør da også en del af komprimeringsstrategien i standarden JPEG. Ved meget høje grader af komprimering, det vil sige ved meget lave rater for bit per pixel, giver komprimering med en DCT dog en række problemer, som vi vil forsøge at forklare. (Rabbani & Jones; 1991, side 113)

En linie i et billede, hvor pixelværdierne ændrer sig meget hurtigt, kaldes også en *kant*. Hvis man forestiller sig et helt sort billede med en enkelt hvid streg, så vil stregen udgøre en kant. En pixelrække, som krydses af stregen vil have mere eller mindre konstante pixelværdier (svarende til det sorte), undtagen for en enkelt 'peak', dér hvor den hvide streg krydser.

Hvis man forsøger at komprimere et billede med en sådan kant med den ovenfor beskrevne metode, vil man snart sande, at ved høj komprimering har kanten en tendens til at blive meget udtværet eller udvisket. Den såkaldte 'peak' har det altså med at forsvinde, når man fjerner for mange af svingningerne.

Problemet løses bedst ved at anvende en repræsentation, hvor basisvektorerne til en vis grad er lokaliseret i rum. Sådanne repræsentationer oplever ikke samme udtværing af billedet ved høj komprimering, så længe man ikke fjerner de vektorer, der er lokaliserede i rum omkring kanten.

Ved at opdele et billede i mindre segmenter kan man i første omgang opnå en vis lokalisering i rum med DCT'en. Man foretager simpelthen en separat transformation af billedets segmenter. Jo finere opdeling af billedet, desto større lokalisering i rum får man. En ulempe ved denne tilgang er, at det generelt er sværere at opnå gode komprimeringsresultater, når man komprimerer små billeder.

Et andet problem med at opdele billedet er, at en del nabopixels i det oprindelige billede uundgåeligt kommer til at tilhøre forskellige delbilleder, som komprimeres hver for sig og som følge deraf forvrænges forskelligt under komprimeringen. Når det samlede billede genskabes efter komprimering bliver dette forhold synligt i form af såkaldte *blocking artifacts*. Et eksempel herpå er vist på figur 3.6.

Den nævnte problemstilling gør sig gældende for de fleste naturlige billeder, da disse indeholder kanter af den ene eller den anden art. Pointen er altså, at når man ønsker høj komprimering af billeder og samtidig en god billedkvalitet, er cosinus-transformationen utilstrækkelig.

Umiddelbart synes lokalisering i rum at være det, man skal gå efter. Den euklidiske basis illustrerer dog, hvorfor man ikke udelukkende kan gå efter lokalisering i rum. Hvis man sætter koefficienten, som svarer til et element i denne basis, til nul, så svarer dette til at sætte en enkelt pixelværdi lig nul. Dette vil typisk betyde, at man farver en enkelt pixel sort.

Figur 3.6 Til venstre: Originalt billede af Pathfinders rover Sojourner. Til højre: Samme billede komprimeret til 0,178 bit per pixel med JPEG. De såkaldte 'blocking artifacts' er tydelige. (Kiely & Klimesh; 2003, side 3)

Man kan altså ikke klare sig udelukkende med frekvens-lokalisering eller rum-lokalisering. Man er nødt til at finde en mellemvej. Wavelet-transformationen tilbyder det, man søger.

3.5 Lifting

Vi vil forsøge at behandle wavelettransformationerne ved en tilgang, som er mere orienteret mod anvendelse end den hidtidige. Vi vil introducere en metode kaldet *lifting*, som er baseret på en simpel ide. Vi skal i afsnittet se, at metoden gør det muligt at foretage transformationer, som svarer til det ovenfor beskrevne. Afsnittet er baseret på (Jensen & la Cour-Harbo; 2001, side 7-50)

Som tidligere betragter vi vektorer i et rum med en dimension N, som er delelig med 2^p, hvor $p \in \mathbb{N}$.

Definition 3.25
Lad $z = (z_0, \ldots, z_{N-1}) \in \mathbb{C}^N$, *lad* $z_e = (z_0, z_2, \ldots, z_{N-2})$ *og lad* $z_o = (z_1, z_3, \ldots, z_{N-1})$. *Lad endvidere* $P : \mathbb{C}^{N/2} \to \mathbb{C}^{N/2}$ *og* $U : \mathbb{C}^{N/2} \to \mathbb{C}^{N/2}$ *være givne afbildninger. Afbildningen* $L : \mathbb{C}^N \to \mathbb{C}^N$ *defineret ved*

$$L(z) = (z_e + U(z_o - P(z_e)), z_o - P(z_e))$$

kaldes da en førstetrins lifting. *Afbildningen* P *kaldes en* forudsigelse *og* U *kaldes en* opdatering.

Ovenstående ser kompliceret ud, men det er egentlig ret simpelt. For en vektor z, som repræsenterer et *signal*, eksempelvis en pixelrække i et billede, må man forvente en vis struktur. Definitionen af en førstetrins lifting bygger på antagelsen om, at en komponent $z(n)$ i z til en vis grad afhænger af dens nærmeste naboer. Mere præcist antages, at komponenten $z(2n+1)$ afhænger af komponenten $z(2n)$, det vil sige, at de ulige komponenter indeholdt i z_o afhænger af de lige, som er indeholdt i z_e. Afhængighedens karakter modelleres i en vis forstand af forudsigelsen P. De sidste $N/2$

komponenter i $L(z)$

$$z_o - P(z_e),$$

ses at være differensen mellem de ulige komponenter og forudsigelsen af disse baseret på P. De sidste $N/2$ komponenter i $L(z)$ indeholder derfor en slags angivelse af, hvor god forudsigelsen er i forhold til det egentlige signal. På baggrund af denne information foretages en passende opdatering af signalet bestemt af U. Denne opdatering gemmes i de første $N/2$ komponenter i $L(z)$ som

$$z_e + U(z_o - P(z_e)).$$

Formålet med opdateringen er at transformere en del af signalet, så det bliver mere forudsigeligt i forhold til P. Forklaringen virker måske nok stadig lidt diffus, men følgende eksempel skulle gerne virke opklarende.

Eksempel 3.26
Vi forestiller os, at vi modtager et signal z, som vi har en forventning om er mere eller mindre konstant. Vi vil derfor benytte som forudsigelse om de ulige komponenter, at de er lig de lige, altså sættes

$$P(z_e) = z_e.$$

For at gøre signalet mere modtageligt overfor denne forudsigelse vil vil jævne det ud ved at lade de første $N/2$ komponenter i $L(z)$ være lig de parvise gennemsnit af komponenterne i z. Dette gøres ved at sætte

$$U(z_o - P(z_e)) = \frac{1}{2}(z_o - P(z_e)) = \frac{1}{2}(z_o - z_e),$$

sådan, at for $n < N/2$ er

$$\begin{aligned}(Lz)(n) &= z_e(n) + \frac{1}{2}(z_o(n) - z_e(n)) \\ &= z(2n) + \frac{1}{2}(z(2n+1) - z(2n)) \\ &= \frac{1}{2}(z(2n) - z(2n+1)).\end{aligned}$$

Lad $z \in \mathbb{C}^8$ og lad

$$z = (48, 48, 40, 16, 56, 40, 8, 24).$$

Så er

$$\begin{aligned}z_o &= (48, 16, 40, 24), \\ z_e &= (48, 40, 56, 8).\end{aligned}$$

og

$$\begin{aligned}z_o - z_e &= (0, -24, -16, 16), \\ U(z_o - P(z_e)) &= (0, -12, -8, 8),\end{aligned}$$

så

$$L(z) = (48, 28, 48, 16, 0, -24, -16, 16).$$

Det ses, at de første 4 komponenter er lig de parvise gennemsnit af komponenterne i z, og de 4 sidste komponenter er lig de parvise differenser, det vil sige

$$L(z) = (\frac{1}{2}(z_e + z_o), (z_o - z_e)).$$

Ovenstående transformation bevarer informationen i z. Givet $L(z)$ kan vi nemlig gendanne z ved

$$z(n) = \begin{cases} (Lz)(n/2) - {}^1\!/_2(Lz)((n+N)/2) & \text{for } n \text{ lige,} \\ (Lz)((n-1)/2) + {}^1\!/_2(Lz)((n-1+N)/2) & \text{for } n \text{ ulige.} \end{cases}$$

Dette kan let indses ved indsættelse. Transformationen kan altså foretages begge veje. ◇

Vi vil se på samme eksempel igen.

Eksempel 3.27
Til lifting-proceduren kan vi tilføje en normering, sådan at

$$L(z) = \left(\sqrt{2}(z_e + \frac{1}{2}(z_o - z_e)), \frac{1}{\sqrt{2}}(z_o - z_e)\right).$$

Så får vi for $z = (48, 48, 40, 16, 56, 40, 8, 24)$, at

$$L(z) = \frac{1}{\sqrt{2}}(96, 56, 96, 32, 0, -24, -16, 16).$$

Sammenlignes ovenstående med vektoren $[z]_H$ i eksempel 3.10 ses tydeligt, at de to vektorer er ens bortset fra rækkefølgen af komponenterne. Liftingproceduren beskrevet her er faktisk identisk med Haar-transformationen i eksempel 3.10, blot er transformationen i dette afsnit foretaget med en anden ordning af basisvektorerne. (Jensen & la Cour-Harbo; 2001, side 21) ◇

Liftingproceduren giver oplagt mulighed for iteration. Ved igen at foretage en lifting på $L(z)$'s første ${}^N\!/_2$ komponenter opnås en *andettrins* lifting. Dette kan gentages op til $\log_2(N)$ gange, idet antallet af komponenter, som indgår, halveres i hvert trin. Som nedenstående eksempel antyder, svarer en p'tetrins lifting til en transformation ind i en p'tetrins waveletbasis.

Eksempel 3.28
I dette eksempel vil vi foretage både en wavelettransformation og en lifting i to dimensioner. Vi vil benytte Haar-transformationen. Lad z være matricen

$$z = \begin{pmatrix} 0 & 0 & 8 & 8 & 16 & 8 & 8 & 8 \\ 0 & 8 & 16 & 24 & 16 & 16 & 16 & 8 \\ 8 & 24 & 40 & 32 & 24 & 24 & 24 & 16 \\ 16 & 40 & 56 & 56 & 32 & 16 & 24 & 16 \\ 32 & 48 & 56 & 48 & 40 & 32 & 24 & 8 \\ 24 & 40 & 48 & 40 & 32 & 32 & 16 & 0 \\ 16 & 32 & 40 & 40 & 40 & 16 & 8 & 8 \\ 8 & 16 & 24 & 16 & 32 & 16 & 8 & 0 \end{pmatrix}.$$

Hvis vi foretager en førstetrins lifting af den første række, får vi vektoren

$$(0, 8\sqrt{2}, 12\sqrt{2}, 8\sqrt{2}, 0, 0, -4\sqrt{2}, 0).$$

Nu beholdes de 4 sidste komponenter, og der foretages endnu en lifting på de første 4, hvilket giver vektoren

$$(8, 20, 8, -4, 0, 0, -4\sqrt{2}, 0).$$

Den sidste lifting foretages og resulterer i vektoren

$$(14\sqrt{2}, 6\sqrt{2}, 8, -4, 0, 0, -4\sqrt{2}, 0).$$

Dette gøres for hver række i z, hvilket resulterer i matricen

$$\begin{pmatrix}
14\sqrt{2} & 6\sqrt{2} & 8 & -4 & 0 & 0 & -4\sqrt{2} & 0 \\
26\sqrt{2} & 2\sqrt{2} & 16 & -4 & 4\sqrt{2} & 4\sqrt{2} & 0 & -4\sqrt{2} \\
48\sqrt{2} & -4\sqrt{2} & 20 & -4 & 8\sqrt{2} & -4\sqrt{2} & 0 & -4\sqrt{2} \\
64\sqrt{2} & -20\sqrt{2} & 28 & -4 & 12\sqrt{2} & 0 & -8\sqrt{2} & -4\sqrt{2} \\
72\sqrt{2} & -20\sqrt{2} & 12 & -20 & 8\sqrt{2} & -4\sqrt{2} & -4\sqrt{2} & -8\sqrt{2} \\
58\sqrt{2} & -18\sqrt{2} & 12 & -24 & 8\sqrt{2} & -4\sqrt{2} & 0 & -8\sqrt{2} \\
50\sqrt{2} & -14\sqrt{2} & 16 & -20 & 8\sqrt{2} & 0 & -12\sqrt{2} & 0 \\
30\sqrt{2} & -2\sqrt{2} & 8 & -20 & 4\sqrt{2} & -4\sqrt{2} & -8\sqrt{2} & -4\sqrt{2}
\end{pmatrix}$$

Derefter foretages for hver søjle i ovenstående en tredjetrins lifting og vi får matricen z_L givet ved

$$\begin{pmatrix}
181 & -35 & 30\sqrt{2} & -25\sqrt{2} & 26 & -6 & -18 & -16 \\
29 & -19 & -6\sqrt{2} & -17\sqrt{2} & 2 & -6 & -6 & -4 \\
36\sqrt{2} & -16\sqrt{2} & 12 & 0 & 8\sqrt{2} & -4\sqrt{2} & -2\sqrt{2} & -2\sqrt{2} \\
-25\sqrt{2} & 11\sqrt{2} & 0 & 2 & -2\sqrt{2} & 2\sqrt{2} & -8\sqrt{2} & 6\sqrt{2} \\
12 & -4 & 4\sqrt{2} & 0 & 4 & 4 & 4 & -4 \\
16 & -16 & 4\sqrt{2} & 0 & 4 & 4 & -8 & 0 \\
-14 & 2 & 0 & -2\sqrt{2} & 0 & 0 & 4 & 0 \\
-20 & 12 & -4\sqrt{2} & 0 & -4 & -4 & 4 & -4
\end{pmatrix}$$

Hvis vi foretager en Haar-transformation som beskrevet i afsnit 3.3.1 på hver række i z og derefter på hver søjle i den transformerede, får vi matricen z_F givet ved

$$\begin{pmatrix}
4 & 4 & 4 & -4 & 4\sqrt{2} & 0 & -4 & -12 \\
4 & 4 & -8 & 0 & 4\sqrt{2} & 0 & -16 & -16 \\
0 & 0 & 4 & 0 & 0 & -2\sqrt{2} & 2 & 14 \\
-4 & -4 & 4 & -4 & -4\sqrt{2} & 0 & 12 & 20 \\
8\sqrt{2} & -4\sqrt{2} & -2\sqrt{2} & -2\sqrt{2} & 12 & 0 & -16\sqrt{2} & -36\sqrt{2} \\
-2\sqrt{2} & 2\sqrt{2} & -8\sqrt{2} & 6\sqrt{2} & 0 & 2 & 11\sqrt{2} & 25\sqrt{2} \\
2 & -6 & -6 & -4 & -6\sqrt{2} & -17\sqrt{2} & -19 & -29 \\
26 & 6 & 18 & 16 & -30\sqrt{2} & 25\sqrt{2} & 35 & 181
\end{pmatrix}$$

Ved sammenligning af matricerne z_F og z_L ser vi, at for hver søjle i z_F findes en søjle i z_L, som indeholder de samme komponenter dog permuteret og muligvis med et andet fortegn. Dette skyldes de forskellige ordninger af basisvektorerne for de to transformationer. ◇

Pointen i ovenstående er at klargøre, at vi ved lifting kan opnå samme resultat, som ved den filterbaserede transformation. Fordelen ved lifting er, at det ved denne tilgang er nemmere at konstruere en transformation ud fra en forventning om et signals struktur. Med en sådan forventning kan forudsigelsen og opdateringen fastlægges uden videre arbejde.

3.6 Afrunding

Wavelets har været kendt siden begyndelsen af det 20. århundrede, men den matematiske teori for wavelets er langt nyere, nærmere bestemt fra 1992 og tilskrives den belgisk fødte matematiker Ingrid Daubechies.

I dette kapitel har vi introduceret to tilgange til den diskrete wavelettransformation. Tilgangen gennem filterbanks er hentet fra traditionel signalanalyse og anvender et begrebsapparat dels stammende fra lineær algebra. Elementerne i waveletbasen optræder som filtre, der opløser signalet i delsignaler med egenskaber afhængige af basiselementernes egenskaber. Tilgangen giver en kort og præcis beskrivelse af transformationen, nemlig som et basisskift fra den euklidiske basis over i en ortonormalbasis bestående af wavelets.

Lifting-tilgangen er nyere og forudsætter ikke det store matematiske teoriapparat. Tilgangen får umiddelbart filterbanks-tilgangen til at virke unødvendigt kompliceret i forhold til praksis. Til gengæld bidrager lifting ikke med nogen egentlig forståelse af, hvad der foregår.

De to tilgange komplementerer hinanden meget godt. Filterbankstilgangen giver et teoretisk indblik, og lifting-tilgangen giver et værktøj til at foretage wavelettransformationer på simpel vis.

Vi har med vilje fremstillet sætningen om perfekt rekonstruktion uden kravet om ortonormalitet. I HOVEDRAPPORTEN ser vi nemlig på en transformation, som giver perfekt rekonstruktion uden overhovedet at være ortogonal.

Ingrid Daubechies
(1954-)

Daubechies modtog sin bachelor- henholdsvis ph.d.-grad i 1975 og 1980 fra det frie universitet i Brussels, Belgien, hvor hun efterfølgende var ansat i en forskerstilling indtil 1987. Fra 1987 til 1994 var hun ansat ved den tekniske stab hos AT&T Bell Laboratories, fraregnet ophold ved henholdsvis University of Michigan (1990) og Rutgers University (1991-1993). Daubechies er mest kendt for sit arbejde med wavelets, specielt bogen *Ten Lectures on Wavelets* fra 1992, en bog for hvilken American Mathematical Society tildelte hende Leroy P. Steele prisen i 1994. I 1997 modtog hun også Ruth Lyttle Satter prisen fra samme selskab. Daubechies er på nuværende tidspunkt ansat ved Mathematics Department and the Program in Applied and Computational Mathematics ved Princeton University. Hendes forskningsinteresser koncentrerer sig om de matematiske aspekter af tidsfrekvens-analyse, specielt wavelets, i såvel teoretisk som anvendelsesorienteret øjemed. http://www.pacm.princeton.edu/~ingrid/

4 Modellering

> If we have a good model of what that probability distribution looks like then we can do a good job with coding. (Kiely; 2005, side 35 i Interviews)

Vi skal i dette kapitel se på den del af datakomprimeringsalgoritmen, der foregår efter forbehandlingen af dataene, nemlig den efterfølgende *modellering*.

Som tidligere antydet, er det ofte fordelagtigt at anvende en model for de data, der skal transmitteres. Eller mere nøjagtigt en model for den fysiske informationskilde eller proces, der genererer dataene. Denne model kan anvendes til estimering af entropien af kilden. En god model for en informationskilde fører til en mere effektiv komprimeringsalgoritme. Der findes adskillige tilgange til konstruktion af sådanne matematisk og statistisk baserede modeller. Vi skal i det følgende se på nogle af disse.

I litteraturen kaldes modellen for en fysisk informationskilde ofte selv for en kilde[1]. Vi vil i dette kapitel benytte denne konvention, men for at minimere forvirringen, skal vi omtale den fysiske informationskilde som en *symbolgenerator*, og modellen som *kilden*. Vi skal i det følgende tænke på sådanne kilder som repræsenteret ved stokastiske processer og yderligere begrænse os til studiet af *diskrete* kilder. Disse beskriver symbolgeneratorer, som genererer en besked symbol for symbol. Begrænsningen skyldes, at det udelukkende er de diskrete kilder, som er af relevans for MERs komprimeringsteknik ICER. (Gallager; 1968, side 38), (Shannon; 1948, side 6)

Nærværende kapitel er fortrinsvist baseret på (Rabbani & Jones; 1991, side 15-21). Når anden litteratur er inddraget i fremstillingen er dette angivet.

4.1 Diskret hukommelsesløs kilde

Den mest simple diskrete kilde er den såkaldte diskrete hukommelsesløse kilde.

Definition 4.1 (Diskret hukommelsesløs kilde)
For en diskret hukommelsesløs kilde $\mathcal{K}$ *gælder, at kildesymbolerne er statistisk uafhængige. Kilden er således fuldstændigt beskrevet ved sit symbolalfabet* $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ *og den tilhørende sandsynlighedsfordeling* $\mathcal{P} = \{p(s_1), p(s_2), \ldots, p(s_n)\}$.

Det mest simple tilfælde er det, hvor hvert symbol, s_i, optræder med lige stor sandsynlighed, altså $p(s_i) = 1/n$ for $i = 1, \ldots, n$. Generelt gælder, at kilder med skæve sandsynlighedsfordelinger har lavere entropi end kilder

1 Se for eksempel Gallager (1968) og Rabbani & Jones (1991).

med sådanne ligelig fordelte symbolsandsynligheder. Eller med andre ord: Usikkerheden af hvad en symbolgenerator beskrevet ved en skæv fordeling genererer som det næste symbol, er mindre end usikkerheden af, hvad en symbolgenerator beskrevet ved en ligelig fordeling genererer som det næste symbol. Lad os se et eksempel.

Eksempel 4.2
Antag at vi har en diskret hukommelsesløs kilde $\mathcal{K}_1$ med symbolalfabet $\mathcal{S} = \{A, B, C, D, E\}$ og at forekomsten af hver af symbolerne har sandsynlighed $1/5$. Entropien af denne kilde er givet ved

$$H(\mathcal{K}_1) = -\sum_{i=1}^{5} p_i \log_2 p_i = -5\left(\frac{1}{5}\log_2\frac{1}{5}\right) = 2,32 \text{ bit per symbol.}$$

Antag dernæst, at vi har en diskret hukommelsesløs kilde $\mathcal{K}_2$ med samme symbolalfabet som før, men denne gang med sandsynlighedsfordelingen

$$\mathcal{P} = \left\{\frac{2}{5}, \frac{1}{10}, \frac{1}{5}, \frac{1}{5}, \frac{1}{10}\right\}.$$

Entropien af denne kilde er

$$\begin{aligned} H(\mathcal{K}_2) &= -\left(\frac{2}{5}\log_2\frac{2}{5} + 2\left(\frac{1}{10}\log_2\frac{1}{10}\right) + 2\left(\frac{1}{5}\log_2\frac{1}{5}\right)\right) \\ &= 2,12 \text{ bit per symbol,} \end{aligned}$$

hvilket ses at være mindre end entropien af kilden $\mathcal{K}_1$. (Shannon; 1948, side 7) ◇

Ofte kan det være fordelagtigt at håndtere blokke af symboler i stedet for enkelte symboler.

Definition 4.3
Lad $\mathcal{K}$ være en diskret hukommelsesløs kilde med et alfabet af størrelse n, og lad uddata fra kilden være grupperet i blokke af størrelse N. Hver blok kan da betragtes som et enkelt kildesymbol genereret af en kilde $\mathcal{K}^N$ med et alfabet af størrelse n^N. Kilden $\mathcal{K}^N$ kaldes for den N'te udvidelse af $\mathcal{K}$.

Da vi har at gøre med en hukommelsesløs kilde, er sandsynligheden af symbol $\sigma_i = (s_1, s_2, \ldots, s_n) \in \mathcal{K}^N$ givet ved

$$p(\sigma_i) = p(s_{i_1})p(s_{i_2})\ldots p(s_{i_N}).$$

Entropien af $\mathcal{K}^N$ bestemmes nemt udfra entropien af $\mathcal{K}$, idet der gælder følgende sætning:

Sætning 4.4
Entropien af den N'te udvidelse af $\mathcal{K}$ er N gange entropien af $\mathcal{K}$, det vil sige

$$H(\mathcal{K}^N) = N \cdot H(\mathcal{K}).$$

Bevis
Som bekendt har vi følgende:

$$H(\mathcal{K}) = -\sum_{i=1}^{n} p(s_i)\log p(s_i) \quad \text{og} \quad H(\mathcal{K}^N) = -\sum_{i=1}^{n^N} p(\sigma_i)\log p(\sigma_i).$$

Beviset kan udføres ved induktion. Basistilfældet $N = 1$ er trivielt, da $H(\mathcal{K}^1) = 1 \cdot H(\mathcal{K})$. Antag, at der gælder $H(\mathcal{K}^N) = N \cdot H(\mathcal{K})$, så skal vi vise resultatet for $N+1$; $H(\mathcal{K}^{N+1}) = (N+1)H(\mathcal{K})$.

Lad $\mathcal{K}^N = \{\sigma_1, \sigma_2, \ldots, \sigma_{n^N}\}$. Vi ordner alfabetet for $\mathcal{K}^{N+1}$ sådan, at det er mængden bestående af elementerne ζ_i, $1 \le i \le n^{N+1}$, defineret ved:

$$\begin{aligned}
&\zeta_1 = s_1\sigma_1, \zeta_2 = s_2\sigma_1, \ldots, \zeta_n = s_n\sigma_1,\\
&\zeta_{n+1} = s_1\sigma_2, \zeta_{n+2} = s_2\sigma_2, \ldots, \zeta_{2n} = s_n\sigma_2,\\
&\vdots\\
&\zeta_{(n-1)n+1} = s_1\sigma_n, \zeta_{(n-1)n+2} = s_2\sigma_n, \ldots, \zeta_{n^2} = s_n\sigma_n,\\
&\vdots\\
&\zeta_{(n-1)n^N+1} = s_1\sigma_{n^N}, \zeta_{(n-1)n^N+2} = s_2\sigma_{n^N}, \ldots, \zeta_{n^{N+1}} = s_n\sigma_{n^N}.
\end{aligned}$$

Vi har, at

$$\begin{aligned}
H(\mathcal{K}^{N+1}) &= -\sum_{i=1}^{n^{N+1}} p(\zeta_i)\log p(\zeta_i)\\
&= -\Bigg(\sum_{i=1 \bmod n}^{n^{N+1}} p(\zeta_i)\log p(\zeta_i) + \sum_{i=2 \bmod n}^{n^{N+1}} p(\zeta_i)\log p(\zeta_i) + \ldots\\
&\quad + \sum_{i=n-1 \bmod n}^{n^{N+1}} p(\zeta_i)\log p(\zeta_i)\Bigg).
\end{aligned}$$

Betragt den k'te sum:

$$\begin{aligned}
\sum_{i=k \bmod n}^{n^{N+1}} p(\zeta_i)\log p(\zeta_i) &= \sum_{j=1}^{n^N} p(s_k)p(\sigma_j)\log(p(s_k)p(\sigma_j))\\
&= \sum_{j=1}^{n^N} p(s_k)p(\sigma_j)\log p(s_k)\\
&\quad + \sum_{j=1}^{n^N} p(s_k)p(\sigma_j)\log p(\sigma_j)\\
&= p(s_k)\log p(s_k)\sum_{j=1}^{n^N} p(\sigma_j)\\
&\quad + p(s_k)\sum_{j=1}^{n^N} p(\sigma_j)\log p(\sigma_j)\\
&= p(s_k)\log p(s_k)\sum_{j=1}^{n^N} p(\sigma_j) + p(s_k)H(\mathcal{K}^N),
\end{aligned}$$

hvilket, da $\sum_{j=1}^{n^N} p(\sigma_j) = 1$ er lig

$$p(s_k) \log p(s_k) + p(s_k) H(\mathcal{K}^N). \tag{4.1}$$

Den samlede sum fra ligning (4.1) kan omskrives til

$$\sum_{k=1}^{n} p(s_k) \log p(s_k) + H(\mathcal{K}^N) \sum_{k=1}^{n} p(s_k),$$

som, da

$$\sum_{k=1}^{n} p(s_k) = 1,$$

giver $H(\mathcal{K}) + H(\mathcal{K}^N)$. Ifølge induktionsantagelsen er dette det samme som $H(\mathcal{K}) + N \cdot H(\mathcal{K})$, hvilket som ønsket giver os $(N+1)H(\mathcal{K})$. □

4.2 Markov-kilde

Den diskrete hukommelsesløse kilde er ofte en for simpel model til at give en tilfredsstillende beskrivelse. Generelt gælder, at den foregående del af en besked på betydelig vis kan påvirke sandsynlighederne af det næste symbol; kilden siges da at have *hukommelse*. I denne rapport er sandsynligheden for, at bogstaverne *ME* følges af bogstavet *R* særdeles stor; ergo er sandsynligheden for at se bogstavet *R* i høj grad påvirket af de bogstaver, der kommer før. For digitale billeder forekommer den lignende situation, hvor sandsynligheden for at en given pixel antager en bestemt værdi afhænger af de omkringliggende nabopixels værdier. I sådanne situationer kan modellering finde sted ved hjælp af en *Markov-kilde*. Markov-kilder kendes i litteraturen også under navnet *endelige kontekst-modeller*. MERs komprimeringsalgoritme ICER anvender kontekst-modellering og modellerer således ved en brug af en Markov-kilde.

En Markov-kilde er en generalisering af den diskrete hukommelsesløse kilde.

Definition 4.5 (Markov-kilde)
For en m'te-ordens Markov-kilde $\mathcal{K}$ *afhænger sandsynligheden for at et kildesymbol* s_i *optræder af de foregående* m *symboler. Markov-kilden er beskrevet ved en mængde af betingede sandsynligheder:*

$$p(s_i | s_{j_1}, s_{j_2}, \ldots, s_{j_m}),$$

for $i = 1, 2, \ldots, n$ *og* $1 \leq j_p \leq n$ *med* $p = 1, 2, \ldots, m$.

En Markov-kilde kan opfattes som befindende sig i en eller anden tilstand, der afhænger af de foregående m symboler. For en førsteordens Markov-kilde er der n tilstande, én for hver af kildesymbolerne s_i; for en andenordens Markov-kilde er der n^2 tilstande, én for hvert par af symboler fra kildealfabetet; og så videre, indtil vi for en m'te-ordens Markov-kilde har, at der for denne er n^m tilstande. For en *ergodisk*[2] Markov-kilde er

2 Vi vil i overenstemmelse med Shannon (1948) forstå følgende ved ergodisk:

der en entydig sandsynlighedsfordeling over mængden af tilstande, denne fordeling er kendt som den *stationære* sandsynlighedsfordeling.

En Markov-kilde kan illustreres ved et såkaldt tilstandsdiagram. Her er hver tilstand repræsenteret ved en cirkel og overgangene imellem tilstande ved pile imellem cirklerne. Sandsynligheden af en overgang markeres ved nummeret ved siden af pilen. For et eksempel på et tilstandsdiagram se figur 4.1 i eksempel 4.6. (Sayood; 2000, side 24-25)

Til udregning af entropien for en m'te-ordens Markov-kilde bestemmes $H(\mathcal{K}|s_{j_1}, s_{j_2}, \ldots, s_{j_m})$, altså entropien af kilden givet at den befinder sig i en bestemt tilstand $(s_{j_1}, s_{j_2}, \ldots, s_{j_m})$. Entropien bliver da

$$H(\mathcal{K}|s_{j_1}, s_{j_2}, \ldots, s_{j_m}) = -\sum_{1=i}^{n} p(s_i|s_{j_1}, \ldots, s_{j_m}) \log p(s_i|s_{j_1}, \ldots, s_{j_m}).$$

Disse betingede entropier vægtes med den tilhørende sandsynlighed for at være i den bestemte tilstand, og der summeres over samtlige mulige tilstande, altså

$$H(\mathcal{K}) = -\sum^{|\mathcal{S}|^m} p(s_{j_1}, s_{j_2}, \ldots, s_{j_m}) H(\mathcal{K}|s_{j_1}, s_{j_2}, \ldots, s_{j_m}).$$

Det er vist på sin plads med et eksempel.

Eksempel 4.6
Betragt Markov-kilden $\mathcal{K}$ af orden 2 med et binært kildealfabet $\mathcal{S} = \{0, 1\}$. Kildens tilstandsdiagram er vist på figur 4.1.

De betingede symbolsandsynligheder er givet ved

$$\begin{aligned} p(0|0,0) &= p(1|1,1) = \frac{4}{5} \\ p(1|0,0) &= p(0|1,1) = \frac{1}{5} \\ p(0|0,1) = p(0|1,0) &= p(1|1,0) = p(1|0,1) = \frac{1}{2}. \end{aligned}$$

Der er fire mulige tilstande for Markov-kilden. Disse er repræsenteret ved $(0,0)$, $(0,1)$, $(1,0)$ og $(1,1)$. Grundet symmetrien opfylder den stationære sandsynlighedsfordeling af disse tilstande, at

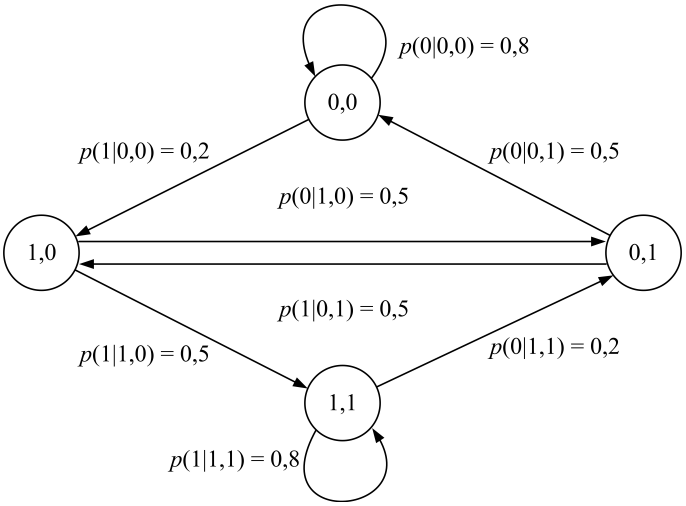
$$p(0,0) = p(1,1) \quad \text{og} \quad p(0,1) = p(1,0). \tag{4.2}$$

Ydermere gælder, da kilden til alle tidspunkter skal befinde sig i en tilstand, at

$$p(0,0) + p(0,1) + p(1,0) + p(1,1) = 1. \tag{4.3}$$

Bemærk at der er to overgange som kan bringe kilden i tilstand $(0,0)$; en er hvis der udsendes et 0, mens den er i tilstand $(0,0)$, hvilket sker

> In an ergodic process every sequence produced by the process is the same in statistical properties. Thus the letter frequencies, diagram frequencies, etc., obtained from particular sequences will, as the lengths of the sequences increase, approach definite limits independent of the particular sequence. Actually this is not true for every sequence but the set for which it is false has probability zero. Roughly the ergodic property means statistical homogeneity. (Shannon; 1948, side 8)



Figur 4.1 Tilstandsdiagram for Markov-kilden af orden 2 med binært kildealfabet. (Rabbani & Jones; 1991, side 18)

med sandsynlighed $^4/_5$; en anden er hvis der udsendes et 0, mens den er i tilstand $(0,1)$, hvilket sker med sandsynlighed $^1/_2$. Et tilsvarende argument gør sig gældende for tilstand $(1,0)$, hvilket fører til følgende ligninger:

$$\begin{aligned} p(0,0) &= \frac{1}{2}\,p(0,1) + \frac{4}{5}\,p(0,0), \\ p(1,0) &= \frac{1}{5}\,p(0,0) + \frac{1}{2}\,p(0,1). \end{aligned}$$

Anvendende ligning (4.2) samt ligning (4.3) følger da, at

$$p(0,0) = p(1,1) = \frac{5}{14} \quad \text{og} \quad p(0,1) = p(1,0) = \frac{1}{7}, \tag{4.4}$$

som er de stationære sandsynligheder for Markov-kilden. Vi kan nu konstruere en tabel 4.1 og udregne entropien for Markov-kilden:

$$H(\mathcal{K}) = -\sum^{2^3} p(s_{i_1}, s_{i_2}, s_i) \log_2 p(s_i | s_{i_1}, s_{i_2}) = 0,80 \text{ bit per symbol.}$$

(Rabbani & Jones; 1991, side 18-19) ◇

Ligesom med den diskrete hukommelsesløse kilde findes der en udvidelse af Markov-kilden.

Definition 4.7
Den N'te udvidelse af en Markov-kilde, kaldet $\mathcal{K}^N$, er en μ'te-ordens Markov-kilde med symboler defineret som blokke, hver bestående af N symboler fra den oprindelige kilde, hvor

$$\mu = \left\lceil \frac{m}{N} \right\rceil .$$

s_{i_1}	s_{i_2}	s_i	$p(s_i \mid s_{i_1}, s_{i_2})$	$p(s_{i_1}, s_{i_2})$	$p(s_{i_1}, s_{i_2}, s_i)$
0	0	0	$4/5$	$5/14$	$2/7$
0	1	1	$1/5$	$5/14$	$1/14$
0	1	0	$1/2$	$1/7$	$1/14$
0	1	1	$1/2$	$1/7$	$1/14$
1	0	0	$1/2$	$1/7$	$1/14$
1	0	1	$1/2$	$1/7$	$1/14$
1	1	0	$1/2$	$5/14$	$1/14$
1	1	1	$4/5$	$5/14$	$2/7$

Tabel 4.1 Sandsynligheder for Markov-kilden. (Rabbani & Jones; 1991, side 19)

Ligesom for den diskrete hukommelsesløse kilde er entropien af en udvidet Markov-kilde $\mathcal{K}^N$, N gange entropien af den oprindelige kilde K (jævnfør sætning 4.4). De udvidede Markov-kilder vil ofte være attraktive i modelleringssammenhænge, da de ligesom de udvidede hukommelsesløse kilder håndterer blokke af symboler frem for enkelte symboler. Der foruden spiller de en betydelig rolle for de såkaldte sammensatte kilder, som vi skal se nærmere på herunder.

4.3 Sammensatte kilder

I mange applikationer er det ikke muligt kun at bruge én kilde til at beskrive symbolgeneratoren, hvorfor der anvendes en *sammensat kilde*. En sammensat kilde består således af et udvalg af forskellige kilder, med kun én kilde værende aktiv til et givet tidspunkt.

Betragt en N'te udvidelse $\mathcal{K}^N$ af en Markov-kilde med kildesymboler $\{\sigma_1, \sigma_2, \ldots, \sigma_{n^N}\}$ og stationære sandsynligheder $\{p(\sigma_1), p(\sigma_2), \ldots, p(\sigma_{n^N})\}$ og betragt dernæst en diskret hukommelsesløs kilde med samme alfabet og samme symbolsandsynligheder. Denne diskrete hukommelsesløse kilde kaldes den sammensatte kilde af $\mathcal{K}^N$ og benævnes $\overline{\mathcal{K}^N}$. Bemærk, at den sammensatte kilde af en diskret hukommelsesløs kilde selv er en diskret hukommelsesløs kilde.

En interessant egenskab ved en sammensat kilde er, at entropien $H(\overline{\mathcal{K}^N})$ af en sammensat kilde altid er større end eller lig med entropien af dens tilsvarende udvidede Markov-kilde, $H(\mathcal{K}^N)$, det vil sige

$$H(\overline{\mathcal{K}^N}) \geq H(\mathcal{K}^N).$$

Dette følger (intuitivt) i og med, at den sammensatte kilde, da denne er en diskret hukommelsesløs kilde, ikke er bundet af begrænsningerne påtvunget en Markov-kilde. Der gælder endvidere ifølge Rabbani & Jones (1991), at entropien per oprindeligt kildesymbol af den sammensatte kilde, kaldet $H_N(\mathcal{K})$, går mod entropien af Markov-kilden som blokstørrelsen N går mod uendelig, det vil sige

$$\lim_{N\to\infty} H_N(\mathcal{K}) = \lim_{N\to\infty} \frac{H(\overline{\mathcal{K}^N})}{N} = H(\mathcal{K}).$$

Eksempel 4.8
Lad os igen se på andenordens Markov-kilden $\mathcal{K}$ med tilstandsdiagrammet givet i figur 4.1. Den sammensatte kilde $\overline{\mathcal{K}}$ er en diskret hukommelsesløs kilde med alfabet $\mathcal{S} = \{0, 1\}$. Grundet symmetrien haves

$$p(0) = p(1) = \frac{1}{2}$$

hvorfor

$$H_1(\mathcal{K}) = H(\overline{\mathcal{K}}) = 1 \text{ bit per symbol.}$$

Det betyder, at en person, der modellerer denne symbolgenerator som en diskret hukommelsesløs kilde, og er uvidende om hukommelsen, observerer en entropi på 1 bit per symbol for kilden.

Den sammensatte kilde $\overline{\mathcal{K}^2}$, som svarer til den anden udvidelse af $\mathcal{K}$, er en diskret hukommelsesløs kilde med fire symboler. Symbolsandsynlighederne er givet i ligning (4.4). Entropien per oprindeligt kildesymbol, $H_2(\mathcal{K})$, fra denne kilde kan beregnes som

$$\begin{aligned} H_2(\mathcal{K}) &= -\frac{1}{2}\left[2\left(\frac{5}{14}\log_2\frac{5}{14}\right) + 2\left(\frac{1}{7}\log_2\frac{1}{7}\right)\right] \\ &= 0,93 \text{ bit per oprindeligt kildesymbol.} \end{aligned}$$

Den sammensatte kilde $\overline{\mathcal{K}^3}$ er en diskret hukommelsesløs kilde med et alfabet bestående af otte symboler, hvis sandsynligheder er givet ved sidste søjle i tabel 4.1. Dens entropi per oprindeligt kildesymbol, $H_3(\mathcal{K})$, bestemmes til

$$\begin{aligned} H_3(\mathcal{K}) &= -\frac{1}{3}\left[2\left(\frac{2}{7}\log_2\frac{2}{7}\right) + 6\left(\frac{1}{14}\log_2\frac{1}{14}\right)\right] \\ &= 0,89 \text{ bit per oprindeligt kildesymbol.} \end{aligned}$$

Bemærk, at som blokstørrelsen N tiltager, bevæger entropien per oprindeligt kildesymbol af den sammensatte kilde, $H_N(\mathcal{K})$, sig mod 0,80 – entropien af andenordens Markov-kilden i eksempel 4.6. (Rabbani & Jones; 1991, side 20) ◇

Sammensatte kilder udgør et kraftfuldt værktøj til modellering, idet sådanne modeller kan bruges til at beskrive meget komplekse datagenerende processer.

4.4 Afrunding

Vi har i dette kapitel introduceret modellering af en datagenererende proces, nærmere bestemt en model kaldet en 'kilde'.

Først har vi beskrevet den diskrete hukommelsesløse kilde. Dernæst har vi præsenteret en generalisering af denne, nemlig en diskret kilde med hukommelse – Markov-kilden. En Markov-kilde er i billedkomprimeringssammenhænge også kendt som en kontekst-model og det er i denne forbindelse, at kilden indgår i MERs komprimeringsteknikker. Markov-kilden er beskrevet ved hjælp af et såkaldt tilstandsdiagram, hvilket der også er givet et eksempel på. Vi har præsenteret udvidelser af de to typer af kilder samt beskrevet en sammensat kilde.

5 Entropikodning

> The entropy encoder is sort of the engine. Given some sort of probabilistic model of the source it compresses data or represents it in a more efficient way through something like a variable length code. (Kiely; 2005, side 35 i INTERVIEWS)

Med udgangspunkt i en datagenererende proces er ideen i entropikodning som tidligere nævnt, at tildele symboler, hvis sandsynlighed for at optræde som uddata fra processen er stor, en kortere kode end dem, hvis sandsynlighed for at optræde er lille, og som derfor tildeles en længere kode.

Vi skal i dette kapitel se eksempler på, hvordan de forbehandlede data kan indkodes med entropikodere, som udnytter en given model (kilde). Kapitlet bygger på et udvalg af forskellig litteratur, hvorfor denne vil blive nævnt løbende.

5.1 Entropi og kodning

Entropikoderens job består typisk i at komprimere data ved at udskifte symboler, der er repræsenteret ved koder med lige lange kodeord, med koder, hvis ord er af variabel længde. Typisk tildeles symbolerne kodeordslængder proportionale med den negative logaritme til sandsynligheden for, at de optræder, hvorfor de mest hyppige symboler anvender de korteste koder. Givet en tilfældig hændelse med et endelig antal udfald, her vores symbolalfabet $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, kan vi for hvert muligt udfald s_i definere mængden af information indeholdt i udfaldet. Hvis $p(s_i)$ er sandsynligheden for, at hændelsen indtræffer, defineres mængden af information, som også kaldes *selv-information*, tilknyttet s_i ved

$$\mathcal{I}(s_i) = \log \frac{1}{p(s_i)} = -\log p(s_i).$$

Det er denne størrelse, længden af koden for symbolet s_i skal være proportional med. (Sayood; 2000, side 13-14)

Entropien for vores hændelse ses at være summen af informationen i de respektive udfald vægtet i forhold til sandsynligheden for, at de indtræffer:

$$\sum_{i=1}^{n} p(s_i)\mathcal{I}(s_i) = -\sum_{i=1}^{n} p(s_i) \log p(s_i).$$

Når vi tager logaritmen med base 2, opnås målet for entropien i *bit* per kildesymbol. Tages logaritmen med base 10, kaldes enheden *hartleys*, og tages den naturlige logaritme, er enheden *nat*. I det følgende benyttes bit per kildesymbol.

Når vi bruger færre bit til at repræsentere symboler, der hyppigere forekommer, vil vi i gennemsnit bruge færre bit per symbol. Den gennemsnitlige længde af koden er imidlertid ikke det eneste, der beskriver en 'god' kode. En anden vigtig egenskab er som tidligere nævnt, at koden skal være entydigt dechifrerbar. Shannons teorem omhandlende informationskilden – *The Noiseless Source Coding Theorem* – (se kapitel 3 i HOVEDRAPPORTEN), lyder i en moderne formulering:

Sætning 5.1
Lad $\mathcal{K}$ være en kilde med entropi $H(\mathcal{K})$, da gælder

$$H(\mathcal{K}) \leq m(\mathcal{K}),$$

hvor $m(\mathcal{K})$ er den gennemsnitligt mindste kodeordslængde for alle entydigt dechifrerbare kodningsteknikker for $\mathcal{K}$.

En variation af sætning 5.1 omhandler indkodningen af blokke af kildesymboler til binære kodeord. Sætningen bringes her uden bevis:

Sætning 5.2
Lad $\mathcal{K}$ være en ergodisk kilde med et alfabet af størrelse n og med entropi $H(\mathcal{K})$. Lad endvidere indkodningsproceduren indkode blokke af N kildesymboler ad gangen til binære kodeord. For et vilkårligt $\delta > 0$ er det muligt, ved at vælge N stort nok, at konstruere en kode på en sådan vis, at det gennemsnitlige antal bit per oprindeligt kildesymbol, kaldet $\overline{L}$, opfylder

$$H(\mathcal{K}) \leq \overline{L} < H(\mathcal{K}) + \delta.$$

Ydermere må den venstre ulighed være opfyldt for en hvilken som helst entydig dechifrerbar kode for blokken af N kildesymboler.

Sætningen siger, at en hvilken som helst kilde kan indkodes tabsfrit med en kode, hvis gennemsnitlige antal bit per kildesymbol er vilkårligt tæt på entropien af kilden. (Rabbani & Jones; 1991, side 21)

Vi skal i den resterende del af dette kapitel betragte et udvalg af teknikker til entropikodning. Vi skal først se på Huffman-kodning, som er en af de mest 'berømte' komprimeringsalgoritmer. Huffman-kodning er en af bestanddelene i MERs tabsfri billedkomprimeringsteknik LOCO-I. Dernæst skal vi se på de teknikker og metoder, som ICER gør brug af i sin entropikoder.

5.2 Huffman-kodning

Inden vi kan præsentere Huffmans algoritme, er der nogle begreber, der skal præsenteres. Et mål for effektiviteten af en kode er dennes redundans.

Definition 5.3
Redundansen *af en kode $\mathcal{C}$ er givet ved forskellen mellem den gennemsnitlige længde l og entropien H.*

Definition 5.4
En optimum-kode *er en kode med minimum redundans.*

Des mindre redundansen er, des højere er optimaliteten af den pågældende kode. Teknikken kendt som Huffman-kodning producerer prefix-koder, som er optimale for en given kilde. (Solomon; 1998, side 25), (Huffman; 1952, side 1098)

Huffmans procedure er baseret på de to følgende egenskaber for optimum prefix-koder.

Sætning 5.5
En optimum prefix-kode har følgende to egenskaber:

1. *De symboler der optræder oftere – det vil sige dem, hvis sandsynlighed for at forekomme er stor – vil have kortere kodeord, og de, der optræder sjældnere, vil have længere kodeord.*
2. *De to symboler, der optræder sjældnest, vil have samme kodeordslængde.*

Bevis
Egenskab 1 følger nemt, for hvis symboler, der optræder oftere, havde kodeord, der var længere end kodeordene for symboler der optræder sjældnere, ville det gennemsnitlige antal bit per symbol være større, end hvis situationen var den omvendte. Dette er i strid med, at koden er en optimumkode.

For egenskab 2 antages det, at der eksisterer en optimum-kode $\mathcal{C}$, hvis to mindst sandsynlige symboler har kodeord af forskellige længde. Antag endvidere, at det længste kodeord er k bit længere end det korteste. Da $\mathcal{C}$ er en prefix-kode, kan vi afkorte det længste kodeord med k bit og stadig have, at de to kodeord er forskellige. Endvidere, da de to kodeord repræsenterer de mindst sandsynlige symboler, vil intet andet kodeord kunne være længere end disse, hvorfor der ingen fare er for, at det afkortede kodeord kan være et prefix af nogen andre kodeord. Koden er altså stadig en prefix-kode. Ydermere opnår vi ved at fjerne de k bit en kode, hvis gennemsnitlige længde er kortere end den af $\mathcal{C}$. Men dette er i strid med, at $\mathcal{C}$ er en optimum-kode. Antagelsen må altså være forkert, hvorfor de to mindst sandsynlige symboler må have samme kodeordslængde. □

Til beskrivelse af Huffman-kodning kan det med fordel benyttes, at Huffman-koder er prefix-koder og dermed kan beskrives ved hjælp af et binært træ, hvor de eksterne knuder svarer til symboler. Med udgangspunkt i dette faktum bliver Huffmans algoritme som følger:

Algoritme 5.6 (Huffman-kodning)
Inddata: *Kildealfabet $\mathcal{S}$ og sandsynlighedsfordeling $\mathcal{P}$ for en kilde $\mathcal{K}$.*
Opret en tabel over symboler ordnet således, at symbolerne med de største tilhørende sandsynligheder står øverst.

1. *Lad de to nederste symboler i tabellen være eksterne knuder i et binært træ.*
2. *Opret en ny intern knude, der forbinder disse to.*
3. *Opfat denne nye knude som et nyt symbol, hvis sandsynlighed er givet ved summen af sandsynlighederne af de eksterne knuder i dens undertræ.*

4. *Indsæt det nye symbol i tabellen, og sortér igen denne efter sandsynligheder.*

Denne procedure gentages, så længe antallet af symboler i tabellen er ≥ 2.
Uddata: *Binært Huffman-træ for prefix-koden* $\mathcal{C}$.

Som man måske vil have bemærket, er uddata fra ovenstående algoritme ikke entydig. Indføres derimod konventionen om, at det nederste symbol i tabellen altid placeres til venstre under opbygningen af træet, opnås entydighed. Lad os se på et eksempel.

Eksempel 5.7
Lad os som i eksempel 4.2 tage udgangspunkt i en kilde $\mathcal{K}$ med kildealfabet $\mathcal{S} = \{A, B, C, D, E\}$ og sandsynlighedsfordeling

$$\mathcal{P} = \left\{ \frac{2}{5}, \frac{1}{10}, \frac{1}{5}, \frac{1}{5}, \frac{1}{10} \right\}.$$

Vi ordner symbolerne efter sandsynligheden for, at de forekommer:

$$\{A, C, D, B, E\}.$$

Opbygningen af det binære Huffman-træ er som angivet på figur 5.1.

Huffman-koden for $\mathcal{K}$ bliver:

$$\begin{array}{lcl} \text{A} & \mapsto & 0 \\ \text{C} & \mapsto & 10 \\ \text{D} & \mapsto & 110 \\ \text{B} & \mapsto & 1110 \\ \text{E} & \mapsto & 1111 \end{array}$$

Det vil sige

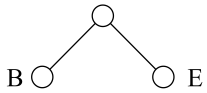
$$\mathcal{C} = \{0, 1110, 10, 110, 1111\}.$$

Den gennemsnitlige længde af denne Huffman-kode er på $2,2$ bit per symbol. Vi har tidligere bestemt entropien for denne kode til $2,12$ bit per symbol, hvorfor redundansen af koden er $2,2 - 2,12 = 0,08$ bit per symbol. ◇

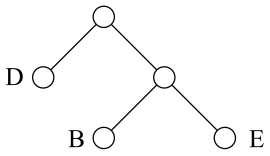
Vælger vi at placere knuderne i tabellen efter et andet mønster end det ovenfor beskrevne, kan vi opnå et andet Huffman-træ og dermed en anderledes kode. Nærmere bestemt kan vi, hvis vi ønsker at opnå en kode med *minimum varians*, indføre reglen om altid at placere de forbundne symboler så højt i tabellen som muligt i forhold til tabellens elementers respektive sandsynligheder (Sayood; 2000, side 45).

En Huffman-kode med minimum varians kan undertiden være at foretrække. Lad os for eksempel forestille os, at vi transmitterer symboler fra vores kildealfabet med en hastighed af 10.000 symboler per sekund, og at den kanal, som vi transmitterer over, har en kapacitet på 22.000 bit i sekundet. Det betyder, at kanalen vil forvente at modtage og transmittere 22.000 bit per sekund, intet mere og intet mindre. Raten, hvormed bittene genereres, vil ofte ikke være konstant, men vil variere omkring kanalkapaciteten. Af den grund vil man ofte i praksis i et kommunikationssystem

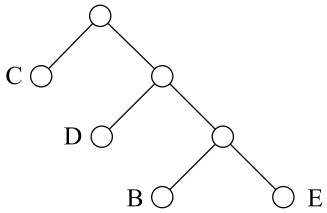
Symbol	Sandsynlighed
A	0,4
C	0,2
D	0,2
B	0,1
E	0,1



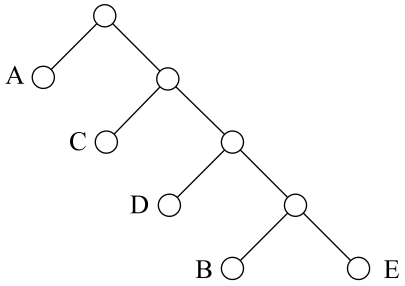
Symbol	Sandsynlighed
A	0,4
C	0,2
D	0,2
BE	0,2



Symbol	Sandsynlighed
A	0,4
DBE	0,4
C	0,2



Symbol	Sandsynlighed
CDBE	0,6
A	0,4



Figur 5.1 Opbygningen af det binære Huffman-træ.

placere uddata fra kildeindkoderen i en buffer (eventuelt vil man have buffere diverse andre steder i kommunikationssystemet også). Denne buffers funktion er at udjævne variationerne i raten, hvormed bittene genereres. Imidlertid er en buffer af en endelig størrelse, hvorfor variansen af koden bliver en betydende faktor; des større variansen er, des mere besværligt bliver det at designe bufferen. (Sayood; 2000, side 43-45)

Lad os betragte eksemplet fra før.

Eksempel 5.8
Med minimum varians Huffman-kodning bliver indkodningen af $\mathcal{K}$ som vist her nedenfor.

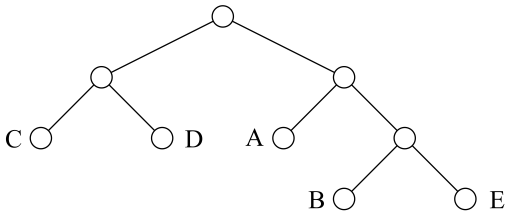
$$\begin{aligned} \text{A} &\mapsto 10 \\ \text{C} &\mapsto 00 \\ \text{D} &\mapsto 01 \\ \text{B} &\mapsto 110 \\ \text{E} &\mapsto 111 \end{aligned}$$

Det vil sige

$$\mathcal{C}_{\text{mv}} = \{10, 110, 00, 01, 111\}.$$

Det binære Huffman-træ og tabellens udseende umiddelbart før opbygningen af dette, kan ses på figur 5.2.

Symbol	Sandsynlighed
ABE	0,6
CD	0,4



Figur 5.2 Sidste skridt i opbygningen af det binære Huffman-træ for koden med minimum varians.

Den gennemsnitlige længde af denne Huffman-kode er ligeledes $2,2$ bit per symbol. Redundansen af koden er $2,2 - 2,12 = 0,08$ bit per symbol og altså dermed ikke forskellig fra koden i eksempel 5.7. Til gengæld er variansen af de to koder væsentligt forskellig. Lad os beregne variansen af $\mathcal{C}_{\text{mv}}$. Den stokastiske variabel X er i dette tilfælde en afbildning fra mængden af kodeordslængder $\{2, 3\}$ ind i $\mathbb{R}$. Det bemærkes, at middelværdien er lig den gennemsnitlige længde l af koden, som er 2,2 bit per symbol. Variansen af koden beregnes til

$$\text{Var}(X) = E(X^2) - (EX)^2 = 5,0 - 4,84 = 0,16.$$

Den stokastiske variabel af koden i eksempel 5.7 vil have udfaldsrummet $\{1, 2, 3, 4\}$, og koden $\mathcal{C}$ vil ligeledes have middelværdi 2,2. Variansen af denne kode bliver da

$$\text{Var}(X) = 6,2 - 4,84 = 1,36,$$

som er betydeligt større. ◇

Lad os se på betydningen af variansen af koderne $\mathcal{C}$ og $\mathcal{C}_{\text{mv}}$ fra henholdsvis eksempel 5.7 og eksempel 5.8 med udgangspunkt i den tidligere omtalte kanal. Antages det, at vi igennem længere tid sender en følge udelukkende bestående af B'er og E'er, vil koden $\mathcal{C}$ kræve en transmissionsrate på 40.000 bit per sekund. Koden $\mathcal{C}_{\text{mv}}$ vil kun kræve en rate på 30.000 bit per sekund. I første tilfælde skal bufferen gemme 18.000 bit, hvorimod bufferen i andet tilfælde kun skal håndtere 8.000 bit. Har vi derimod en følge bestående udelukkende af A'er, vil $\mathcal{C}$ give en rate på 10.000 bit per sekund. Da kanalen stadig forventer 22.000 bit per sekund, skal bufferen altså på en eller anden måde kompensere for de manglende 12.000 bit. Koden $\mathcal{C}_{\text{mv}}$'s kodeord for A er én bit længere, hvorfor raten bliver på 20.000, og bufferen skal således kun kompensere for 2.000 bit. En kode med minimum varians er således langt nemmere at håndtere for bufferen, idet udsvingene her er minimeret. (Sayood; 2000, side 45)

Det faktum, at de to forskellige Huffman-indkodninger i de foregående eksempler giver koder med samme redundans, er intet tilfælde. Det skyldes, som nævnt i begyndelsen af dette afsnit, at en Huffman-kode er optimal for en given kilde. Dette kan vises ved først at nedskrive de

nødvendige betingelser for en optimum-kode og derefter godtgøre, at opfyldelsen af disse fører til designet af en Huffman-kode. Vi vil godtgøre resultatet for det binære tilfælde. De to første betingelser er de i sætning 5.5 allerede godtgjorte egenskaber for en optimum-kode. Derudover er der to yderligere betingelser. Disse lyder som følger:

3. I træet svarende til optimum-koden skal der være to forgreninger for hver intern knude.
4. Antag, at vi ændrer en intern knude til en ekstern knude ved at forbinde alle de eksterne knuder udspringende fra denne til et sammensat symbol for et reduceret alfabet. Hvis det oprindelige træ var optimalt for det oprindelige alfabet, så skal også det reducerede træ være optimalt for det reducerede alfabet.

Sætning 5.9
Den binære Huffman-kode er en variabel-længde optimum-kode.

Bevis
Lad os begynde med at godtgøre betingelse 4. Hvis denne betingelse ikke var opfyldt, ville vi kunne finde en kode med mindre gennemsnitlig længde for det reducerede alfabet og dernæst udvide det sammensatte ord for således at opnå et træ med en tilhørende kortere gennemsnitlig længde end det oprindelige 'optimum'-træ. Men dette er i modstrid med påstanden om, at det oprindelige træ var optimalt.

Betingelse 3 skyldes, at hvis der fandtes en intern knude med kun én forgrening, ville vi kunne fjerne denne uden at påvirke dechifrerbarheden af koden, mens vi reducerede dens gennemsnitlige længde.

For at opfylde betingelserne 1, 2 og 3, må de to mindst sandsynligt forekommende symboler tildeles kodeord af maksimum længde. Ydermere må de eksterne knuder for disse symboler i træet udspringe fra samme interne knude. Altså må kodeordene for disse symboler være ens på nær på den sidste bit. Betragt nu det fælles prefix af kodeordene for disse symboler som kodeordet for det sammensatte symbol af det reducerede alfabet. Da koden for det reducerede alfabet skal være optimal for, at koden for det oprindelige alfabet kan være optimal, følger vi denne procedure igen. For at opfylde de nødvendige betingelser, må denne procedure gentages indtil vi har et reduceret alfabet, der kun indeholder ét sammensat symbol. Men dette er jo præcis den indlejrede procedure i Huffmans algoritme. Derfor må de ovenfor specificerede fire nødvendige betingelser for en optimum-kode – som alle opfyldes af proceduren i Huffmans algoritme – også være tilstrækkelige betingelser. □

Den gennemsnitlige længde l for en optimum-kode – og dermed også for en Huffman-kode – er begrænset såvel opad til som nedad til.

Sætning 5.10
Den gennemsnitlige længde l af en optimum-kode for en kilde $\mathcal{K}$ er begrænset nedad til af entropien og opad til af entropien plus én (bit), altså

$$H(\mathcal{K}) \leq l < H(\mathcal{K}) + 1.$$

Beviset for ovenstående sætning tager sit udgangspunkt i Kraft-McMillans ulighed (se kapitel 2) samt Jensens ulighed, som er givet i følgende lemma. For et bevis for Jensens ulighed henvises for eksempel til (Cover & Thomas; 1991, side 25-26).

Lemma 5.11 (Jensens ulighed)
Lad E angive det forventede (det vil i det diskrete tilfælde sige: $E[X] = \sum_{s\in\mathcal{S}} p(s)s$). Hvis f er en konveks funktion, og X er en tilfældig variabel, vil der gælde, at

$$E[f(x)] \geq f(E[X]).$$

Er f derimod konkav, gælder uligheden den anden vej

$$E[f(x)] \leq f(E[X]).$$

Nu til beviset for sætning 5.10.

Bevis
Antag, at vi har en kilde $\mathcal{K}$ med symbolalfabet $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ og sandsynlighedsfordeling $\mathcal{P} = \{p(s_1), p(s_2), \ldots, p(s_N)\}$, samt at den gennemsnitlige længde for optimum-koden er l.

Lad os begynde med den nedre grænse. Vi kan opskrive forskellen mellem entropien af kilden, $H(\mathcal{K})$, og den gennemsnitlige længde af koden, l, som følger:

$$\begin{aligned}
H(\mathcal{K}) - l &= -\sum_{i=1}^{N} p(s_i)\log_2 p(s_i) - \sum_{i=1}^{N} p(s_i)l_i \\
&= \sum_{i=1}^{N} p(s_i)\left(\log_2 p(s_i)^{-1} - l_i\right) \\
&= \sum_{i=1}^{N} p(s_i)\left(\log_2 p(s_i)^{-1} - \log_2 2^{l_i}\right) \\
&= \sum_{i=1}^{N} p(s_i)\log_2 \frac{2^{-l_i}}{p(s_i)} \\
&\leq \log_2\left(\sum_{i=1}^{N} 2^{-l_i}\right).
\end{aligned}$$

Da vi ved fra analyse, at log er en konkav funktion (den anden afledte af log er ≤ 0 overalt), følger ulighedstegnet ovenfor af Jensens ulighed (lemma 5.11). Da koden er en optimum-kode og dermed entydigt dechifrerbar, gælder Kraft-McMillans ulighed,

$$\sum_{i=1}^{N} 2^{-l_i} \leq 1,$$

hvorfor

$$H(\mathcal{K}) - l \leq 0 \quad \Leftrightarrow \quad H(\mathcal{K}) \leq l.$$

For at vise den øvre grænse vises det, at der eksisterer en entydig dechifrerbar kode med $l = H(\mathcal{K}+1)$. Derfor må en eventuel optimumkode have en gennemsnitlig længde, som er mindre end eller lig $H(\mathcal{K}+1)$. Vi definerer

$$l_i = \left\lceil \log_2 \frac{1}{p(s_i)} \right\rceil,$$

hvoraf følger at

$$\log_2 \frac{1}{p(s_i)} \leq l_i < \log_2 \frac{1}{p(s_i)} + 1. \tag{5.1}$$

Fra den venstre ulighed i ligning (5.1) har vi, at

$$2^{-l_i} \leq p(s_i) \quad \Rightarrow \quad \sum_{i=1}^{N} 2^{-l_i} \sum_{i=1}^{N} \leq p(s_i) = 1.$$

Ifølge sætning 2.12 eksisterer en entydigt dechifrerbar (prefix-) kode med kodeordslængder $\{l_1, l_2, \ldots, l_N\}$. Den gennemsnitlige kodeordslængde kan begrænses opad til ved at benytte højre ulighed i ligning (5.1):

$$l = \sum_{i=1}^{N} p(s_i) l_i < \sum_{i=1}^{N} p(s_i) \left(\log_2 \frac{1}{p(s_i)} + 1 \right)$$

eller

$$l < H(\mathcal{K}) + 1.$$

Den øvre grænse her er, som det fremgår af ovenstående tilnærmelse, en smule løs – for en stramning af grænsen samt et bevis for denne se (Gallager; 1978, side 671). □

Huffmans algoritme kan udvides til at danne koder for ikke-binære tilfælde, det vil sige for q-ære kodealfabeter. For en beskrivelse af dette se eventuelt Sayood (2000). Ligeledes findes der en dynamisk variant af Huffman-indkodning, altså en indkodningsprocedure, hvor man ikke på forhånd kender sandsynlighedsfordelingen af uddata fra kilden. Denne kaldes også gerne adaptiv Huffman-kodning.

Inden vi kommer til de adaptive eller tilpasningsdygtige varianter af entropi-indkodning, skal vi først se på endnu en teknik til entropikodning, nemlig Golomb-kodning, samt se på, hvordan denne finder sin anvendelse i MERs komprimeringsalgoritme ICER.

5.3 Golomb-kodning

Golomb-koden er en type kode designet til at indkode heltal under antagelsen, at des større heltallet er, des mindre er sandsynligheden for, at det forekommer.

Før vi beskriver Golomb-koden, skal vi have endnu en slags koder på banen, nemlig de *unære*.

Definition 5.12
En unær kode *for et positivt heltal n består af n 1-taller efterfulgt af et 0.*

Eksempelvis er den unære kode for $n = 4$ givet ved 11110. En unær kode defineres undertiden i stedet ved n 0'er efterfulgt af et 1-tal. Den unære kode er den samme som Huffman-koden for det semi-uendelige alfabet $\{1, 2, 3, \ldots\}$ med sandsynlighedsmodel $\mathcal{P}[k] = 1/2^k$. Da Huffman-koden er optimal, er den unære kode for denne sandsynlighedsmodel altså også optimal. (Sayood; 2000, side 61)

Ideen i Golomb-koden er at splitte et heltal n i to dele; en del repræsenteret ved en unær kode og en del ved en anden kode. Golomb-koden er rent faktisk en familie af koder parametriseret ved et heltal $m > 0$, hvorfor den ofte benævnes $\mathcal{G}_m$. For en Golomb-kode med parameter m, repræsenteres et heltal $n > 0$ ved

$$n = qm + r \quad (0 \leq r < m),$$

hvor q er kvotienten og r resten, når n divideres med m. Kvotienten q kan antage værdierne $0, 1, 2, \ldots$ og repræsenteres ved den unære kode for q. Resten r kan antage værdierne $0, 1, 2, \ldots, m - 1$. Resten r repræsenteres ofte ved en $\lceil \log_2 m \rceil$-bit binær repræsentation, men antallet af krævede bit kan reduceres ved at anvende en $\lfloor \log_2 m \rfloor$-bit binær repræsentation for r for de første $2^{\lceil \log_2 m \rceil} - m$ værdier og en $\lceil \log_2 m \rceil$-bit binær repræsentation af $r + 2^{\lceil \log_2 m \rceil} - m$ for de resterende værdier. Bemærk det simple tilfælde, hvor $m = 2^k$, for $k \in \mathbb{N}_0$. Golomb-koder $\mathcal{G}_{2^k}$ kaldes ofte for Golomb-Rice koder eller bare Rice-koder. (Sayood; 2000, side 63), (Golomb; 1966, side 400), (Weinberger et al.; 1996, side 5)

Lad os se på et eksempel for Golomb-koden $\mathcal{G}_5$.

Eksempel 5.13 ($\mathcal{G}_5$)
Lad os konstruere en Golomb-kode med $m = 5$. Da $\lceil \log_2 5 \rceil = 3$ og $\lfloor \log_2 5 \rfloor = 2$, vil de første $8 - 5 = 3$ værdier af r, det vil sige $r = 0, 1, 2$ være givet ved de to-bit binære repræsentationer af r, de næste to værdier, altså $r = 3, 4$, vil være givet ved tre-bit repræsentationen af $r + 3$. Kvotienten q vil altid være repræsenteret ved den unære kode for q.

Eksempelvis vil kodeordet for $n = 2$ være 010, hvor det første 0 er den unære kode for q og 01 er den to-bit binære repræsentation af $r = 2$. Kodeordet for $n = 3$ er 0110, hvor 0 igen udgør den unære kode for q, men hvor 110 er den tre-bit binære repræsentation af $r + 3$. Kodeordene for $n = 0, 1, \ldots, 21$ er vist i tabel 5.1. For $n = 21$ vil kodeordet være 1111001. ◇

Det kan vises, at Golomb-koder er optimale for fordelinger af formen

$$Q(n) = (1 - \rho)\rho^n,$$

hvor $0 < \rho < 1$ og $n > 0$, det vil sige geometriske sandsynlighedsfordelinger af ikke-negative heltal. For hver fordeling af denne form eksisterer der en værdi af m således, at $\mathcal{G}_m$ giver den kortest mulige gennemsnitlige kodelængde over alle entydigt dechifrerbare koder for ikke-negative heltal. Den optimale værdi af m er ifølge Gallager & van Voorhis (1978) givet ved

$$m = \left\lceil \frac{\log(1 + \rho)}{\log \rho^{-1}} \right\rceil.$$

Valget af $m = 2^k$ fører som sagt til de såkaldte Golomb-Rice koder, $\mathcal{G}_{2^k}$, hvis ind- og afkodningsprocedurer er mere simple: Koden for n består af de

n	q	r	kodeord	n	q	r	kodeord
0	0	0	000	11	2	1	11001
1	0	1	001	12	2	2	11010
2	0	2	010	13	2	3	110110
3	0	3	0110	14	2	4	110111
4	0	4	0111	15	3	0	111000
5	1	0	1000	16	3	1	111001
6	1	1	1001	17	3	2	111010
7	1	2	1010	18	3	3	1110110
8	1	3	10110	19	3	4	1110111
9	1	4	10111	20	4	0	1111000
10	2	0	11000	21	4	1	1111001

Tabel 5.1 Golomb-koden $\mathcal{G}_5$ for $n = 0, 1, \ldots, 21$.

k mindst betydende bit af n efterfulgt af tallet dannet af de tilbageværende bit af n, angivet i unær repræsentation. Længden af indkodningen er

$$k + 1 + \left\lfloor \frac{n}{2^k} \right\rfloor .$$

(Weinberger et al.; 1996, side 5), (Gallager & van Voorhis; 1978, side 228-230)

Givet de rette inddata er det altså langt nemmere at generere en optimal variabel-længde kode med Golomb-indkodning end at gennemgå hele Huffmans algoritme.

ICER i MER anvender et større udvalg af Golomb-koder med forskellige parametre, herunder $\mathcal{G}_5$, i en sammenflettet struktur kendt under navnet sammenflettet entropikodning.

5.4 Sammenflettet entropikodning

Forskellige entropikodere har forskellige styrker og svagheder. Ifølge Howard (1998) gør følgende sig gældende: Huffman-indkodning er hurtig, konstruerer prefix-koder og er optimal blandt disse. Til gengæld er den *adaptive* Huffman-indkodningsprocedure, som vi senere beskriver, forholdsvis langsom. JPEG 2000s aritmetiske entropikodning – som vi i denne rapport kun behandler ved navns nævnelse – giver næsten-optimal komprimering, er god til indkodning af data med skæv sandsynlighedsfordeling og den adaptive modellering håndteres forholdsvist nemt. Aritmetisk kodning er til gengæld langsom og anvender ikke prefix-koder. Golomb-kodning er særdeles hurtig, anvender prefix-koder og er god til skæve sandsynlighedsfordelinger. Desværre er Golomb-koderne dog ofte kun delvist optimale. (Howard; 1998, side 45-46)

Overvejelser i stil med ovenstående er i særdeleshed væsentlige til applikationer som data- og billedkomprimering i MER. ICER i MER anvender således ikke aritmetisk kodning grundet denne tekniks langsommelighed, men anvender i stedet en teknik kendt som *sammenflettet entropikodning*[1]. Sammenflettet entropikodning er en metode, der forsøger at samle

1 Fra engelsk: Interleaved entropy coding.

'det bedste' fra et udvalg af entropi-indkodningsteknikker ved netop, i en og samme applikation, at sammenflette uddata fra de forskellige entropikoder, som kaldes *komponent-kildekoder*. Et vigtigt aspekt ved en sådan sammenfletning er at sikre, at den dannede følge af bit stadig kan afkodes. Dette gøres ved hjælp af en teknik kaldet *afkoder-synkroniseret indkodning*. Her arrangerer indkoderen bittene i den indkodede følge således, at afkoderen ved at læse disse sekventielt automatisk kan trække de bit ud den behøver, præcis når den behøver dem. Der er altså tale om, at indkoderen manipulerer inddata-strømmen, således at afkodning i sidste øjeblik bliver muligt. (Howard; 1998, side 45-46)

Sammenflettet entropikodning tager sit udgangspunkt i *Block Melcode*, som er en teknik til sammenfletning af Golomb-koder med forskellige parametre. I det følgende vil vi med *MPS* forstå et symbol, hvis sandsynlighed for at forekomme er stor, og med *LPS*[2] et symbol, hvis sandsynlighed for at forekomme er lille.

5.4.1 Block Melcode

Som tidligere beskrevet kan vi til indkodning af ikke-negative heltal n anvende en Golomb-kode med parameter m ($\mathcal{G}_m$). Heltallet n udtrykkes som bekendt på formen $qm + r$, hvor $0 \leq r < m$ og q udskrives unært (q 1-taller efterfulgt af ét 0). Til indkodning med Block Melcode indføres følgende konvention: Hvis $m = 2^k$ udskrives r som et k-bit binært tal. Hvis $m = 2^k + t$ ikke er en potens af 2, $0 < t < 2^k$, udskrives r som et k-bit binært tal, mens hvis $2^k - t \leq r < 2^k$ udskrives $r + 2^k - t$ som et $(k + 1)$-bit binært tal. (Howard; 1998, side 45-46)

Formålet med en Block Melcode er, at tillade en sammenfletning af uddata fra Golomb-koder med forskellige parametre. Block Melcode gør brug af den ovenfor nævnte afkoder-synkroniserede indkodning samt en *indkodningsbuffer*. Indkodningsbufferen gør brug af en *kø-struktur* hvilket betyder, at data fjernes fra bufferen i samme rækkefølge, som de er blevet placeret i bufferen. Hvis et antal af n forekomster af MPS'er kodes med $\mathcal{G}_m$, vil en blok af m på hinanden følgende MPS'er blive tilføjet et 1-tal til den unære del af udtrykket for n. Dette 1-tal kan umiddelbart placeres i bufferens kø af uddata. Efter et antal af disse MPS-blokke vil der forekomme en blok med r MPS'er efterfulgt af et LPS, og indkodningen vil være slut. Dette tillader terminering af den unære del med et 0 samt indkodning af r som et binært tal. Disse $k + 1$ eller $k + 2$ bit kan derefter placeres i køen af uddata som en LPS-blok. (Howard; 1998, side 46), (Ono et al.; 1989, side 0255-0256)

For at kunne afkodes korrekt må en bloks indkodning – én bit for en MPS-blok og $k + 1$ eller $k + 2$ bit for en LPS-blok – optræde i uddatafølgen, når afkoderen er klar til at afkode det *første* symbol i blokken. For at afkode et symbol indkodet med parameter m læser afkoderen enten (a) et 1 eller (b) et 0 efterfulgt af k eller $k + 1$ bit. Tilfælde (a) indikerer forekomsten af m på hinanden følgende MPS'er. I tilfælde (b) afkodes r's værdi udfra de efterfølgende bit, og denne værdi indikerer forekomsten af r på hinanden følgende MPS'er efterfulgt af et LPS. Ved indkodningens afslutning kan delvist-dannede blokke indkodes ved at opfylde de resterende pladser med MPS'er, hvilket blot efterlader afkoderen med oversky-

2 More Probable Symbol (MPS) og Less Probable Symbol (LPS).

dende symboler. En lignende mekanisme kan anvendes til at rydde ud i uddata-køen, hvis denne indeholder for mange blokke. I dette tilfælde kan afkoderen spore indkoderens kø-størrelse og kan således undlade at tage de unødvendige symboler i betragtning. (Howard; 1998, side 46), (Ono et al.; 1989, side 0255-0256)

Den væsentligste del af komponent-kildekoderne i ICERs sammenflettede entropikoder udgøres af Golomb-koder med forskellige parametre. De resterende udgøres af andre prefix-koder.

5.4.2 Udvidelse af Block-Melcode til andre prefix-koder

Ideen fra Block Melcode kan udvides fra kun at omfatte Golomb-koder til at omfatte et større udvalg af prefix-koder. Nedenstående beskrivelse stammer fra Howard (1998).

Først indføres en klasse af såkaldte *UP/S-koder*, unære prefix/suffix-koder.

Definition 5.14 (UP/S-kode)
I en UP/S-kode *indeholder kodeordet for et heltal n en unær prefix-del indikerende et interval i hvilket n falder, samt en suffix-del identificerende n i det pågældende interval.*

Motivationen for at anvende UP/S-koder til sammenflettet entropikodning er, at vi for at reducere den krævede kø-størrelse er interesserede i at udskrive en del af kodeordet, før indkodningen er helt tilendebragt. Dette gøres ved at udskrive hvert 1-tal i den unære prefix-del, så snart vi kender det, og udskrive det efterfølgende 0 af prefix-delen samt suffix-delen når løbet ender.

Vi definerer en specifik UP/S-kode ved at inddele heltallene $0, 1, 2, \ldots$ i mængder $U_0, U_1, U_2, \ldots$ af størrelser $u_0, u_1, u_2, \ldots$. Hver mængde fyldes helt, inden vi fortsætter til den næste. For at beregne prefix-delen af kodeordet for heltallet n begynder vi med U_0 og udskriver et 1 for hver mængde U_i op til, men ikke med, mængden, som indeholder n. Derefter udskrives et enkelt 0.

Hver mængde U_i associeres med et kodningstræ, T_i, for indkodning af suffix-delen. Der kan være tale om en hvilken som helst entydigt dechifrerbar kode indeholdende u_i kodeord. Suffix-delen af kodeordet for n tages direkte fra dens mængdes kodningstræ.

En kodning af ikke-negative heltal er tilendebragt når u_i og T_i er specificeret for samtlige i. En redegørelse for, hvorledes diverse koder til indkodning af heltal kan udtrykkes som UP/S-koder er givet i (Howard; 1998, side 49).

Elementer af såvel Block Melcode som udvidelsen af denne til andre prefix-koder kan genfindes i ICERs sammenflettede entropikoder, som er beskrevet i kapitel 5 af Hovedrapporten.

5.5 Adaptiv entropikodning overordnet

De to ovenfor gennemgåede komprimeringsteknikker; Huffman- og Golomb-kodning er såkaldte *ikke-adaptive* teknikker. Ikke-adaptive entropikoder er karakteriseret ved, at de er rigide, det vil sige, at de ikke løbende

modificerer deres indkodning (og afkodning) i forhold til de data, der komprimeres. I modsætning til disse teknikker er de *adaptive* teknikker, som løbende 'studerer' de rå data og tilpasser sig efter disse.

Nogle komprimeringsteknikker anvender en algoritme, som gennemløber datasættet to gange; ét gennemløb til bestemmelse af statistikken og ét gennemløb til udførelse af komprimeringen. Sådanne teknikker kaldes for *semi-adaptive* komprimeringsteknikker. Et eksempel på en sådan teknik kunne være en Huffman-indkodning af en specifik tekststreng, hvor sandsynlighedsfordelingen af symbolerne ikke på forhånd er givet. Algoritmen udfører først et gennemløb til bestemmelse af frekvensen af de i strengen indgående symboler og dernæst et andet gennemløb til den egentlige Huffman-indkodning. (Solomon; 1998, side x)

En komprimeringsteknik kan også være *lokal adaptiv*, hvilket vil sige, at den tilpasser sig til de lokale omstændigheder i inddata-strømmen og varierer sin tilpasning efterhånden, som den flytter sig fra et område af inddata-strømmen til et andet. (Solomon; 1998, side x)

Mange komprimeringsteknikker findes i en adaptiv udgave, eksempler er Huffman-kodning, den i JPEG 2000 anvendte aritmetiske entropikodning og ikke mindst Golomb-Rice kodning, som anvendes i MERs tabsfri billedkomprimeringsalgoritme LOCO-I. De adaptive tilsnit af enkelte teknikker er dog i høj grad bundet til selve strukturen af algoritmerne. Som et eksempel på en adaptiv tilgang skal vi her nedenfor skitsere hovedideen i adaptiv Huffman-kodning.

5.5.1 Adaptiv Huffman-kodning

Ideen i adaptiv Huffman-kodning er at lade såvel ind- som afkoder begynde med at have et tomt Huffman-træ og så simultant lade dem opbygge ens træer. Til et hvilket som helst tidspunkt i processen, skal ind- og afkoder altså have ens kopier af træet og dermed ens koder, selv om disse koder løbende modificeres i takt med de nye frekvensbestemmelser for kildesymbolerne. Ind- og afkoder er altså på en måde synkroniserede. Man kan også sige, at afkoderen afspejler operationerne udført af indkoderen. En af hovedpointerne er, at der kun kræves ét gennemløb af algoritmen, idet frekvensbestemmelserne foregår løbende og symbolerne indkodes med det samme.

Efterhånden som træet (træerne) opbygges undersøges det løbende, om de repræsenterer en optimum-kode for de indkodede symboler og disses foreløbige sandsynlighedsfordeling, hvis ikke omarrangeres træet. Dette medfører en ændring af koden.

Vi skal her overordnet skitsere en algoritme til binær adaptiv Huffman-kodning kaldet FGK-algorimen[3]. Følgende gennemgang er baseret på Vitter (1987).

Algoritme FGK gør brug af *søskende-egenskaben*[4] og sætning 5.16. Begge er første gang formuleret af Gallager (1978). For et bevis for sætning 5.16 se (Gallager; 1978, side 670). I det følgende skal vi ved s_j forstå det j'te symbol i kildealfabetet af størrelse n og ved w_j, kaldet vægten,

3 Navnet skyldes følgende personers bidrag til algoritmen: Faller (1973), Gallager (1978) og Knuth (1985).

4 Fra engelsk: The sibling property.

antallet af forekomster af a_j indtil videre – altså den fortløbende frekvensbestemmelse.

Definition 5.15 (Søskende-egenskaben)
Et binær kode beskrevet ved et binært kodetræ med p eksterne knuder af ikke-negative vægte har søskende-egenskaben *hvis:*

1. *De p knuder har ikke-negative vægte $w_1, w_2, \ldots, w_p$, og vægten af hver knude er summen af vægtene af dens børn.*
2. *Knuderne kan nummereres i ikke-aftagende orden efter vægte, således at knude $2j-1$ og $2j$ er søskende, $1 \leq j \leq p-1$, og deres fælles forælder-knude er højere i nummereringen.*

Sætning 5.16
En binær prefix-kode er en Huffman-kode, hvis og kun hvis dens kodetræ opfylder søskende-egenskaben.

I algoritme FGK nummereres knuderne i Huffman-træet i den rækkefølge, hvori de er sammensat ved hjælp af Huffmans algoritme. Ideen er, at det $(t+1)$'ste symbol, $s_{i_{t+1}}$, i den foreløbige genererede følge fra kilden behandles ved hjælp af Huffman-træet for de t foregående symboler, $\mathscr{H}_t = s_{i_1}, s_{i_2}, \ldots, s_{i_t}$. Efter at have kodet og afkodet $s_{i_{t+1}}$ ved hjælp af Huffman-træet for $\mathscr{H}_t$ består problemet i hurtigt at få udført opdateringen af Huffman-træet for $\mathscr{H}_{t+1}$, inden det næste symbol modtages. Algoritmen er som følger:

Algoritme 5.17 (FGK)
Inddata: *Symbol $s_{i_{t+1}}$ samt Huffman-træ for følgen $\mathscr{H}_t$.*

1. *Indkoderen koder følgens $(t+1)$'ste symbol $s_{i_{t+1}}$ ved hjælp af 0'er og 1-taller.*
2. *Afkoderen tilvejebringer det oprindelige symbol ved hjælp af sin egen kopi af træet for $\mathscr{H}_t$.*
3. *Såvel ind- som afkoder opdaterer deres kopier af træet før det næste symbol sendes, således at det bliver et Huffman-træ for $\mathscr{H}_{t+1}$. Denne opdatering udføres i to faser:*
   a. *Sæt q lig $s_{i_{t+1}}$'s eksterne knude. Ombyt successivt indholdet af q (inklusiv dens undertræ) med indholdet af den højst nummererede knude af samme vægt, og sæt forælderen til denne knude lig q.*
   b. *Opnå Huffman-træet for $\mathscr{H}_{t+1}$ ved at tælle vægtene for $s_{i_{t+1}}$'s eksterne knuder og dets forfædre op med 1.*

Uddata: *Huffman-træ for følgen $\mathscr{H}_{t+1}$.*

Sætning 5.18
Uddata fra algoritme FGK er et Huffman-træ.

Bevis
Dette godtgøres ved hjælp af søskende-egenskaben: Antallet af knuder er det samme før og efter optællingen. Betingelse 1 og anden del af betingelse 2 af søskende-egenskaben er trivielt bevaret efter optællingen. Altså må vi skulle fokusere på de specifikke knuder, hvis vægte er blevet udsat for en optælling. Før hver sådan knudes vægt tælles op, er det den højst

nummererede knude af sin vægt. Ergo kan vægten forøges med 1 uden at overskride vægten af den næste knude i nummereringen, hvorfor søskendeegenskaben er bevaret. □

En sidste detalje omhandler introduktionen af nye symboler i det dynamiske Huffman-træ. Huffman-træet indeholder hele tiden en tom (ekstern) knude, der repræsenterer de endnu ikke forekommende symboler fra alfabetet. Hver gang, der introduceres et nyt symbol, splittes denne knude i to nye tomme knuder. Det $(t+1)$'ste symbol indkodes ved den sti i træet, der går fra roden til den ene af de to tomme knuder, efterfulgt af nogle bit, der specificerer, hvilket af de endnu ikke forekommende symboler, der er tale om. Disse bit er givet ved en simpel prefix-kode.

Komprimeringsalgoritmen ICER i MER baserer sig på adaptiv entropikodning, mere præcist på en adaptiv udgave af sammenflettet entropikodning. Ligesom i tilfældet med den adaptive Huffman-kodning skal også her konstrueres de samme estimater til den foreløbige sandsynlighedsfordeling på ind- og afkodersiden. Denne opgave stiller store krav til designet af entropikoderen.

5.6 Afrunding

I dette kapitel har vi behandlet den del af en komprimeringsteknik, som udfører selve komprimeringen, nemlig entropikoderen.

Indledningsvis har vi præsenteret visse generelle aspekter af entropi og kodning, herunder begrebet selv-information samt Shannons 'Noiseless Source Coding Theorem'. Derefter har vi behandlet et udvalg af for MER relevante metoder til entropikodning: Huffman-kodning, Golomb-kodning og sammenflettet entropikodning, herunder Block Melcode – en teknik til sammenfletning af uddata fra Golomb-indkodere med forskellige parametre.

Vi har givet en metode til konstruktion af Huffman-koder med minimum varians samt vist, at Huffmans algoritme fører til konstruktionen af optimum prefix-koder. Ydermere har vi opstillet en nedre og en øvre grænse for den gennemsnitlige længde af en optimum-kode.

Endelig har vi kort behandlet de grundlæggende elementer i adaptiv entropikodning og eksemplificeret disse ved den adaptive Huffman-kodning.

Del II

Kanalkodning i MER-missionen

6 Introduktion til kanalkodning

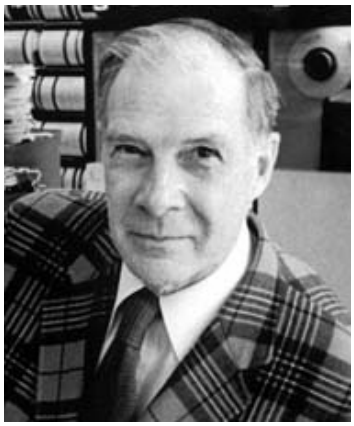
> There are the most insane codes out there. (Kötter; 2005, side 28 i Interviews)

I denne del af rapporten vil vi beskæftige os med den del af kommunikationssystemet, som omhandler kanalkodning.

Som beskrevet i kapitel 3 i Hovedrapporten udgør Shannons 1948-arbejde grundlaget for kodningsteorien. De første fejlkorrigerende koder tilskrives Richard Hamming, eksempelvis den såkaldte $(7, 4)$-Hammingkode, som Shannon med reference til Hammings arbejde selv anvendte i sin artikel. Flere af de begreber, som i dag anvendes indenfor kodningsteorien, er da også navngivet efter Hamming, herunder Hammingafstand og Hammingsfære. Disse begreber skal vi se nærmere på nedenfor.

Hamming var ansat ved Bell Telephone Laboratories, da han udviklede sine koder, og grundet diverse ophavsrettigheder blev hans arbejde derfor først udgivet i 1950. På dette tidspunkt havde Marcel Golay inspireret af Shannons arbejde allerede udviklet sine egne koder. Golays koder vil vi også nævne i det følgende.

Vi vil i kapitlet præsentere en stribe elementære koncepter inden for kodningsteorien. Kapitlet er baseret på flere forskellige kilder, som vil blive nævnt løbende.



Richard Wesley Hamming (1915-1998)

Hamming studerede matematik ved universiteterne i Chicago og Nebraska frem til 1939. I 1942 modtag han en ph.d. i matematik fra University of Illinois, og i 1945 blev han en del af Manhattan-projektet og tilknyttet Los Alamos. I 1946 begyndte Hamming at arbejde for Bell Telephone Laboratories, sammen med blandt andet Shannon. Herfra udgav han i 1950 sin berømte artikel om kodningsteori. I 1956 arbejdede Hamming på den tidlige 650 IBM-computer, og hans arbejde der førte blandt andet til udviklingen af højniveau programmeringssprog. I 1976 accepterede Hamming en stilling i datalogi ved Naval Postgraduate School i Monterey. Hamming modtag i sine sene leveår adskillige priser, ikke mindst for sit arbejde inden for kodningsteori. Hamming var iøvrigt kendt for sin røde skotskternede sportsjakke og sine vandede vittigheder. (Davis; 2005, side 21 i Interviews) http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Hamming.html

6.1 Elementære kodningsteoretiske begreber

Det helt overordnede princip i kanalkodning er, at man ved transmission af data tilføjer en mængde redundans, som gør det muligt at opdage fejl i transmissionen. Vi skal senere se, præcist hvordan denne redundans tilføjes.

Data, der skal transmitteres, består i vores tilfælde af en lang følge af symboler fra et endeligt alfabet. Hvis alfabetet $\{0, 1\}$ benyttes kaldes symbolerne *bit*. Følgen af symboler tilføjes redundans i form af *paritetssymboler* ved en process, der kaldes *indkodning*. Herefter foretages modulation, og tilsidst transmitteres det resulterende signal. Dette er overordnet beskrevet i kapitel 3 i HOVEDRAPPORTEN.

Den modulation, vi er interesserede i, er binær modulation, så vi kan tænke på det transmitterede signal som en følge af bit. Under transmissionen kan der opstå to typer fejl. En enkelt bit, som forandres, kaldes en *bitfejl*. En længere sekvens af bitfejl kaldes for en *burstfejl*. I kapitel 3 i HOVEDRAPPORTEN har vi nævnt, at man normalt i praksis anvender soft-decision demodulation, for hvilken et transmitteret symbol ledsages af to bit, der fortæller noget om transmissionens kvalitet. For at gøre vores fremstilling mere overskuelig vil vi se bort fra disse bit og arbejde med de transmitterede data, som om der anvendes hard-decision demodulation. Når transmissionen modtages, foretages demodulationen, og paritetssymbolerne fjernes ved en process kaldet *afkodning*. Tilstedeværelsen af paritetssymbolerne gør det muligt at opdage og korrigere for eventuelle fejl i transmissionen.

Der findes to basale klasser af koder, *blokkoder* og *trelliskoder*[1]. For en blokkode indkodes en tupel af q-ære symboler kaldet et *dataord* til en større tupel af q-ære symboler kaldet et *kodeord*. For en trelliskode ordnes datasymboler i tupler, som kaldes *datarammer*, og kodesymboler ordnes i tupler, der kaldes *koderammer*. Trelliskoden indkoder en i princippet uendelig *følge* af datarammer til en *følge* af koderammer.

Den store forskel på de to klasser af koder er, at for en blokkode afhænger et kodeord kun af et enkelt dataord, mens en koderamme i en trelliskode afhænger af den pågældende dataramme samt flere af de allerede indkodede datarammer. Af denne årsag er trelliskoder en smule mere komplicerede end blokkoder, og vi skal derfor introducere de grundlæggende kodningsteoretiske begreber gennem blokkoder. Trelliskoder omfatter såkaldte *foldningskoder*, som behandles i kapitel 9. (Blahut; 2003, side 10)

Fra nu af vil vi altså interessere os for blokkoder, data*ord* og kode*ord*.

Definition 6.1
En q-ær blokkode $\mathcal{C}$ af størrelse M og bloklængde *n er en mængde af M ord bestående af n symboler fra et alfabet med q bogstaver. Ordene kaldes* kodeord.

Vi vil nøjes med at se på koder med $M = q^k$. En sådan kode kaldes en (n, k)-kode. For sådanne koder er det praktisk at tænke på dataord som q-ære ord af længde k (dem findes der q^k af) og kodeord som q-ære ord af længde n. Indkodningsproceduren, som blev nævnt ovenfor, består i at opstille en en-til-en korrespondance mellem dataordene og kodeordene og benytte sidstnævnte i transmissionen.

1 For yderligere informationer om trelliskoder se Vardy (1998).

Definition 6.2
Raten *for en* (n, k)*-blokkode defineres som forholdet* k/n.

Raten for en blokkode angiver således forholdet mellem længden af kodens dataord og kodeord, altså hvor stor en del af et kodeord, der udgøres af data, det vil sige reel information.

Eksempel 6.3
Lad $\mathcal{C}$ være en binær blokkode ($q = 2$) af størrelse 4 og bloklængde 6, der indeholder kodeordene

$$\{000000, 000111, 111000, 111111\}.$$

Koden giver mulighed for at indkode 2^2 dataord. Disse kan vi kalde

$$\{00, 01, 10, 11\}.$$

Raten af koden er $2/6 = 1/3$ svarende til, at hver gang vi transmitterer 6 symboler, er det kun 2 symboler, som er egentlig data. ◇

Det er klart, at man umiddelbart er interesseret i en stor rate svarende til meget data per transmitteret symbol. Der er imidlertid også andre vigtige egenskaber at tage hensyn til (ellers kunne man jo bare lade være med at kode dataene). For kodeordene indfører vi følgende begreb:

Definition 6.4 (Vægt)
Vægten *af et kodeord defineres som antallet af symboler i kodeordet forskellige fra* 0.

For kodeordene i eksempel 6.3 er de tilhørende vægte $0, 3, 3, 6$. Udover vægten ønsker vi at indføre et andet begreb knyttet til kodeordene, som gør det muligt at beskrive, hvor forskellige disse er.

Definition 6.5 (Hammingafstand)
Hammingafstanden $d(x, y)$ *mellem to* q*-ære følger* x *og* y *af længde* n *er lig antallet af pladser, hvor* x *og* y *er forskellige.*

For koden defineret i eksempel 6.3 er Hammingafstanden for par af kodeord tydeligvis enten 3 eller 6. Det ses også, at Hammingafstanden mellem nul-kodeordet og et andet kodeord c er lig vægten af c.

Sætning 6.6
Hammingafstanden definerer en metrik.

Bevis
Hammingafstanden er tydeligvis ikke-negativ og symmetrisk, og at trekantsuligheden er opfyldt indses let: Antag, at $d(x, y) > d(x, z) + d(z, y)$. Dette medfører, at for enhver plads j med $x_j \neq z_j$ eller $z_j \neq y_j$ gælder også $x_j \neq y_j$. Desuden findes en plads i med $x_i \neq y_i$ samtidigt med, at $x_i = z_i$ og $z_i = y_i$. De to sidste betingelser giver tilsammen $x_i = y_i$, hvilket er en modstrid. Hammingafstanden definerer altså en metrik. □

Hammingafstanden giver anledning til en ny definition.

Definition 6.7
Lad $\mathcal{C} = \{c_\ell | \ell = 0, \ldots, M-1\}$ *være en kode. Da er* minimumsafstanden $d_{\min}$ *for* $\mathcal{C}$ *defineret som*

$$d_{\min} = \min_{c_i, c_j \in \mathcal{C}, i \neq j} d(c_i, c_j).$$

En (n, k)-blokkode med minimumsafstand $d_{\min}$ kendes i litteraturen også som en $(n, k, d_{\min})$-blokkode. (Justesen & Høholdt; 2004a, side 5), (Blahut; 2003, side 11)

En mulig procedure til afkodning af et modtaget ord v er at afkode til det dataord, hvis kodeord $\tilde{v}$ har den mindste Hammingafstand til v. En sådan afkodning kaldes en *'nærmeste nabo'*-afkodning. For en kode med en stor minimumsafstand skal der opstå mange fejl, før man ved 'nærmeste nabo'-afkodning kommer så langt væk fra det rigtige kodeord, at man afkoder til et forkert dataord, det vil sige får en *afkodningsfejl*. Af denne årsag er man generelt interesseret i at benytte koder med stor minimumsafstand. Jo større bloklængde en kode har, jo bedre er mulighederne for at få en stor minimumsafstand, fordi der er flere positioner på hvilke, kodeordene kan være forskellige. Vi er imidlertid nødt til at holde k så lav som mulig, ellers fylder vi jo bare den 'ekstra plads' i rummet op. Vi har altså to modstridende interesser:

1. Vi vil gerne have stor minimumsafstand, det vil sige, at n skal være stor, og at k skal være lille.
2. Vi vil gerne have høj rate, det vil sige, at n ikke må være stor i forhold til k.

Et af hovedproblemerne inden for kodningsteori er at finde koder, som så vidt muligt tilfredsstiller ovenstående behov. Samtidigt vil man gerne have at koderne skal være simple at ind- og afkode.

Dette forhold er en rød tråd gennem hele kodningsdelen (del II) i INDFØRELSEN.

6.2 Fejldetektion versus fejlkorrektion

Vi vil nu gå lidt dybere ned i beskrivelsen af, hvordan koderne kan bruges til at opdage og rette fejl. For en kode med minimumsafstand 2 vil én enkelt fejl i et kodeord være helt synlig, fordi fejlen ikke vil ændre det sendte kodeord til et andet kodeord. Mere generelt er det klart, at hvis minimumsafstanden for en kode er t, så kan $t-1$ fejl i et kodeord opdages eller *detekteres*. Denne observation er en sætning værdig.

Sætning 6.8
En kode $\mathcal{C}$ *er* t*-fejldetekterende, hvis og kun hvis* $d_{\min} \geq t + 1$.

For at kunne rette (eller *korrigere*) t fejl ved 'nærmeste nabo'-afkodning kræves imidlertid en større minimumsafstand.

Sætning 6.9
En kode $\mathcal{C}$ *er* t*-fejlkorrigerende hvis og kun hvis* $d_{\min} \geq 2t + 1$.

Bevis
Antag at $d_{\min} \geq 2t+1$. Hvis et kodeord c pådrager sig mellem 1 og t fejl, så vil det modtagne ord v opfylde $1 \leq d(c,v) \leq t$. Fra trekantsuligheden følger, at for et andet kodeord $\tilde{c}$ gælder at

$$d(c,v) < t+1 \leq d_{\min} - t \leq d(c,\tilde{c}) - d(c,v) \leq d(v,\tilde{c}),$$

så c er altså det kodeord, som ligger tættest på v. Det modtagne ord v vil derfor blive afkodet som det dataord, der hører til c. Koden kan altså rette t fejl.

Antag nu at $\mathcal{C}$ er t-fejlkorrigerende, men at der findes to kodeord c og $\tilde{c}$ med $d(c,\tilde{c}) = d_{\min} \leq 2t$. Antag for nemheds skyld, at

$$c_i \neq \tilde{c}_i \text{ for } i = 1, \ldots, d_{\min} \text{ og } c_i = \tilde{c}_i \text{ ellers.}$$

Lad v være ordet fastlagt ved

$$v_i = \tilde{c}_i \text{ for } i = 1, \ldots, t \text{ og } v_i = c_i \text{ ellers.}$$

v er altså ordet

$$v = \underbrace{\tilde{c}_0 \ldots \tilde{c}_t}_{t} \underbrace{c_{t+1} \ldots c_d}_{d_{\min}-t} \underbrace{\tilde{c}_{d+1} \ldots \tilde{c}_n}_{n-d_{\min}}.$$

Så er $d(v,c) = t$. Da koden er t-fejlkorrigerende er $d(v,\tilde{c}) \geq t+1$ for ellers er c jo ikke det nærmeste kodeord. Der gælder imidlertid, at $v_i \neq \tilde{c}_i$ på

$$n - t - (n-d) = d - t \leq 2t - t = t$$

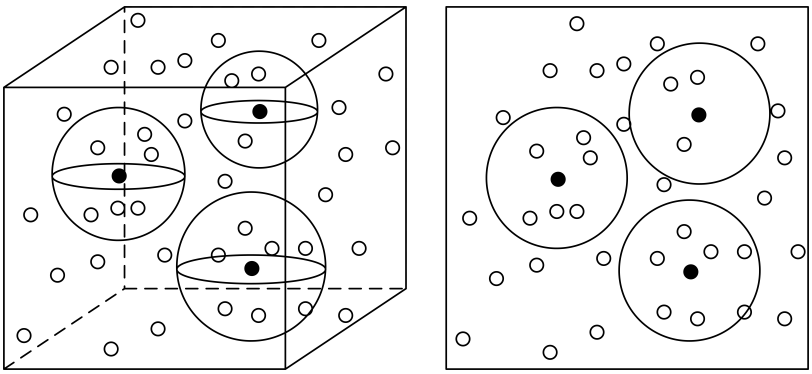
pladser. Men dette er en modstrid da $d(v,\tilde{c}) \geq t+1$. □

Sætning 6.9 er vigtig, da den knytter information om anvendelsen – i form af et antal fejl, som skal kunne korrigeres – sammen med information om kodens matematiske struktur – minimumsafstanden. (Roman; 1997, side 111-112)

I forbindelse med fejl-detektion og fejl-korrektion er det vigtigt at påpege, at de nævnte grænser for kodernes ydelse på disse områder kun gælder, når koderne *udelukkende* anvendes til detektion eller korrektion. For en kode med minimumsafstand $2t+1$, som anvendes til at korrigere t fejl gælder nemlig, at et modtaget ord v, som indeholder flere end t fejl, kan risikere at komme for tæt på et *forkert* kodeord c, altså $d(c,v) < t$ og fejlagtigt blive afkodet til c. Dette vil blive opfattet som en succesfuld afkodning. Koden mister altså nogle af sine fejldetekterende egenskaber, hvis den samtidigt anvendes til fejlkorrektion. I praksis anvender man typisk en såkaldt blandet strategi af såvel fejlkorrektion som fejldetektion. Den blandede strategi går overordnet set ud på at begrænse både fejlkorrektion og fejldetektion, sådan at begge dele kan fungere samtidigt. En sådan strategi er for eksempel at finde i Roman (1997).

6.3 En geometrisk tilgang til blokkoder

Definitionen af Hammingafstanden gør det muligt at benytte en mere intuitiv, geometrisk tilgang til blokkoderne.



Figur 6.1 Hammingsfærer afbildet som henholdsvis rummelige og plane sfærer. ∘ angiver vilkårlige n-tupler (mulige modtagne ord) i det pågældende rum, • angiver kodeord. (Blahut; 2003, side 12), (Roman; 1997, side 120)

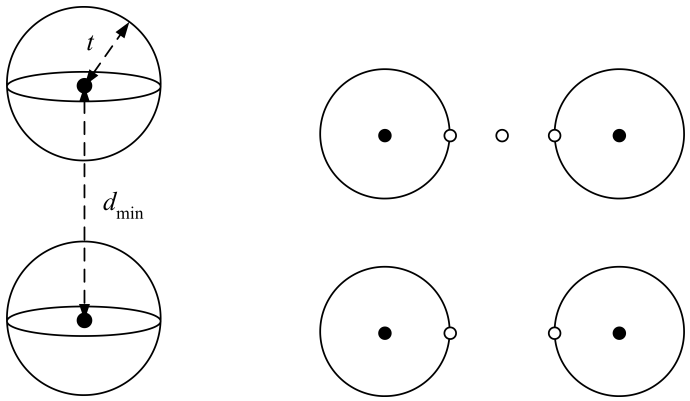
Definition 6.10
Lad V være det metriske rum bestående af alle q-ære n-tupler med Hammingafstanden d som metrik . En Hammingsfære *i V med radius $r > 0$ og med centrum i $v \in V$ defineres da som mængden*

$$S_q^n(v, r) = \{x \in V \mid d(v, x) \leq r\}.$$

Hammingsfærerne giver os mulighed for at illustrere betydningen af mindsteafstanden. For en kode $\mathcal{C}$ med minimumsafstand $2t + 1$ gælder, at hvis man for hvert kodeord $c \in \mathcal{C}$ placerer en Hammingsfære med radius t centreret i c, altså $S_q^n(c, t)$, så er disse Hammingsfærer disjunkte. Dette er illustreret på figur 6.1.

Afkodningen for en t-fejlkorrigerende kode kan ligeledes forklares ved hjælp af sfærerne. Et modtaget ord, der befinder sig i en Hammingsfære, afkodes som det dataord, der svarer til kodeordet i Hammingsfærens centrum. Figur 6.1 illustrerer tydeligt, at man ikke kan være sikker på, at et modtaget ord ligger i en sfære. Alt afhængigt af anvendelsen kan dette afkodningsmæssigt behandles på to måder:

1. En *begrænset afstandsafkoder* afkoder kun modtagne ord, som befinder sig i en Hammingsfære. Et modtaget ord uden for en sådan afkodes *ikke*. I stedet rapporteres et *afkodningssvigt*. En sådan afkoder kaldes en *ufuldstændig afkoder*, fordi den ikke afkoder alle modtagne ord.
2. En *fuldstændig afkoder* afkoder derimod ethvert modtaget ord til det nærmeste kodeord. I geometriske termer er der tale om, at afkoderen opdeler rummet mellem sfærerne ved at allokere hver n-tupel til den sfære, der ligger nærmest. Tupler med lige stor afstand til flere sfærer allokeres vilkårligt. For en t-fejlkorrigerende kode gælder, at hvis der modtages et ord, hvor der er opstået *lidt* flere en t fejl, vil dette ord ofte blive afkodet korrekt (der er dog ingen garanti), fordi afstanden mellem kodeordene typisk er større end minimumsafstanden.



Figur 6.2 Til venstre: Geometrisk fremstilling af minimumsafstanden $d_{\min}$ mellem rumlige sfærer. (Blahut; 2003, side 12) Øverst til højre: $d_{\min} = 4$, så $p_{\mathcal{C}} = 1$. Nederst til højre: $d_{\min} = 3$, så $p_{\mathcal{C}} = 1$. I begge tilfælde kan $\mathcal{C}$ korrigere én fejl. (Roman; 1997, side 120)

Store koder har en tendens til at give en kompleks fuldstændig afkoder, og man vælger derfor ofte i praksis at benytte en begrænset afstandsafkoder. Her igennem formindsker man sandsynligheden for at få en *afkodningsfejl* på bekostning af større sandsynlighed for et *afkodningssvigt*. (Blahut; 2003, side 13)

6.4 Perfekte koder

For en kode med minimumsafstand $2t + 1$ gælder som tidligere nævnt, at Hammingsfærer med radius t placeret med centrum i kodens kodeord vil være disjunkte. Denne observation giver anledning til definitionen af et nyt begreb.

Definition 6.11
Lad $\mathcal{C}$ være en kode med minimumsafstand $d_{\min}$. Kodens pakningsradius*, $p_{\mathcal{C}}$, er det største tal $r \in \mathbb{N}$ for hvilket mængden af sfærer med radius r centreret i $\mathcal{C}$'s kodeord er disjunkte.*

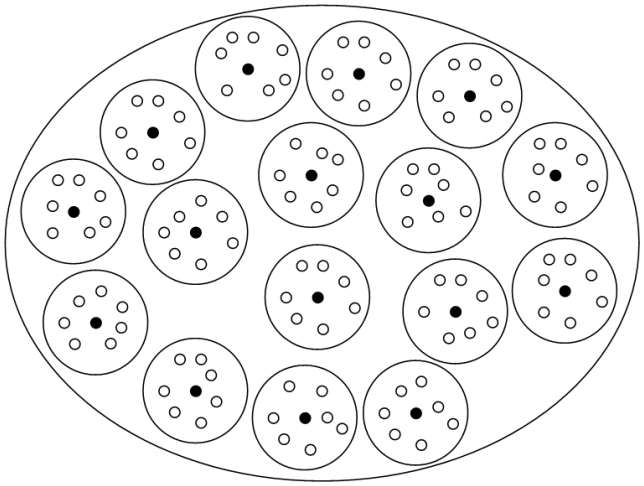
Pakningsradiusen for en kode med $d_{\min} = 2t + 1$ er ifølge det tidligere t. Mere generelt er pakningsradiusen for en kode med minimumsafstand d lig med

$$\left\lfloor \frac{d-1}{2} \right\rfloor .$$

Dette er illustreret på figur 6.2. Sfærerne centreret i kodeordene kaldes for *pakningssfærer*. (Roman; 1997, side 119-120)

For nogle meget få, men vigtige koder gælder et forhold, som er af så stor betydning, at koderne tildeles en særlig betegnelse.

Definition 6.12
Lad $\mathcal{C} = \{c_1, \ldots, c_M\}$ være en q-ær kode med bloklængde n og pakningsradius r. Lad A være kodens alfabet. Koden $\mathcal{C}$ kaldes perfekt*, hvis ethvert*



Figur 6.3 En illustration af en perfekt kode – nærmere bestemt en binær $(7,4)$-Hammingkode. Antallet af kodeord for denne er $2^4 = 16$ og antallet af binære strenge af længde 7 er $2^7 = 128$. Alle disse strenge er indeholdt i en pakningssfære. (Roman; 1997, side 122)

element $v \in A^n$ er indeholdt i en pakningssfære, det vil sige

$$A^n \subseteq \bigcup_{i=1}^{M} S_r^n(c_i, r).$$

Da sfærerne er disjunkte, er definitionen ækvivalent med at foreningen af pakningssfærerne giver hele rummet, det vil sige

$$A^n = \bigcup_{i=1}^{M} S_r^n(c_i, r).$$

På figur 6.3 er 'geometrien' for en sådan perfekt kode illustreret. Perfekte koder forekommer af flere årsager sjældent i praktiske anvendelser: For det første er kodernes parametre (n, M, k) helt fastlåste, hvorfor de ikke kan tilpasses særlige krav i anvendelserne. For det andet har koderne den svaghed, at hvis der opstår for mange fejl ved transmissionen vil afkodningen med garanti føre til en afkodningsfejl. Perfekte koder spiller da heller ingen rolle for kodningen i MER-missionen. (Roman; 1997, side 121)

Som figur 6.3 illustrerer, kan en perfekt kode med bloklængde n og minimumsafstand $2t + 1$ opfattes som en opskrift til pakning af kugler med radius t i et n-dimensionalt rum. Opdagelsen og betydningen af disse relationer mellem kodning og kuglepakning er beskrevet i Thompson (1983).

Repetitionskoden, som vi skal se på i næste afsnit, er et eksempel på en perfekt kode. Koder som består af kun ét kodeord eller af samtlige q-ære strenge af en fast længde udgør ligeledes perfekte koder. Disse kaldes trivielle koder. (Roman; 1997, side 229)

De vigtigste perfekte koder blev beskrevet første gang i Hamming (1950) og Golay (1949).

Familien af Hammingkoder benævnes undertiden $\mathcal{H}_r(h)$, hvor r angiver antallet af symboler i kodens alfabet og h er en parameter, der påvirker bloklængden. Der gælder, se Roman (1997), at for hver primtalspotens r og for hver $h > 0$ er Hammingkoden $\mathcal{H}_r(h)$ en r-ær $(n, k, d_{\min})$-kode med parametre

$$n = \frac{r^h - 1}{r - 1}, \quad k = n - h, \quad d_{\min} = 3.$$

I næste kapitel skal vi se et eksempel på en $(7, 4)$-Hammingkode. (Roman; 1997, side 116-117)

Golaykoderne består af blot fire koder, hvoraf kun de to er perfekte; en binær og en ternær kode. Den binære kode $\mathcal{G}_{23}$ er en $(23, 12, 7)$-kode, og den ternære kode $\mathcal{G}_{11}$ er en $(11, 6, 5)$-kode. Golaykoderne er unikke i den forstand, at en vilkårlig lineær kode med samme parametre som Golaykoden er ækvivalent til denne. Desuden er det vist, at den eneste ikke-trivielle, perfekte kode med minimumsafstand større end 3 er $\mathcal{G}_{23}$ (Roman; 1997, side 229). Resultatet er vigtigt, fordi det overflødiggør yderligere søgning efter perfekte koder, og samtidigt angiver det hvilke kuglepakninger, der kan lade sig gøre i hvilke rum. Golaykoderne har rent faktisk været i spil i rumsammenhæng, nemlig om bord på Voyager. (Roman; 1997, side 117, 224-229), (Wicker; 1998, side 2136)

For Hammingkoderne kan man vise, at de er unikke i en særlig klasse af koder, nemlig klassen af *lineære* koder (Roman; 1997, side 229). Denne klasse skal vi se nærmere på i næste kapitel.

6.5 Eksempler på simple koder

Vi vil her med udgangspunkt i de begreber, som tidligere er beskrevet, give et par eksempler på simple koder; en paritetscheckkode og en repetitionskode. Disse koder er yderst simple, men illustrerer ikke desto mindre en vigtig pointe.

En *paritetscheckkode* er et eksempel på en fejldetekterende kode. Indkodning af et dataord, som består af k bit, foregår ved at dataordet tilføjes en ekstra bit, således at antallet af 1-taller i kodeordet er lige[2]. En paritetscheckkode er altså en $(n, n-1)$-kode med minimumsafstand 2.

Eksempel 6.13 (Paritetscheckkode)
Dataord bestående af 4 bit indkodes til kodeord bestående af 5 bit på følgende vis:

$$\begin{aligned} 0000 &\mapsto 00000 \\ 0001 &\mapsto 00011 \\ &\vdots \quad\quad \vdots \\ 1111 &\mapsto 11110 \end{aligned}$$

Koden kan detektere én fejl, men ikke korrigere nogen. Til gengæld har koden en høj rate, nemlig $(n-1)/n$, i dette tilfælde $4/5$. ◇

2 Fastsættes antallet af 1-taller i koden til at være et ulige tal kaldes koden for en *ulige* paritetscheckkode. Den her beskrevne kode er en *lige* paritetscheckkode.

En binær *repetitionskode* kan som udgangspunkt kun indkode to dataord, nemlig 0 og 1. Indkodningen foregår ved at dataordet gentages n gange. Herved får man en $(n, 1)$-kode, med minimumsafstand n.

Eksempel 6.14 (Repetitionskode)
For $n = 5$ fås en kode, som giver følgende indkodning:

$$\begin{aligned} 0 &\mapsto 00000 \\ 1 &\mapsto 11111 \end{aligned}$$

Repetitionskoden har i modsætning til paritetscheckkoden fantastiske fejlkorrigerende egenskaber. Der kan nemlig korrigeres $\lfloor (n-1)/2 \rfloor$ fejl. Til gengæld har repetitionskoden en forfærdelig lav rate, nemlig $1/n$. ◇

For en repetitionskode gælder, at hvis minimumsafstanden er ulige, er koden perfekt. I eksemplet ovenfor kan der placeres sfærer med radius 2 omkring de to kodeord, således at sfærerne er disjunkte samtidigt med, at de dækker hele rummet.

Disse to eksempler viser med al tydelighed, at vi ikke uden videre kan få både høj rate og gode fejlkorrigerende egenskaber. Jo flere fejl vi ønsker at kunne rette, desto mere redundans er vi altså nødt til at tilføje.

6.6 Afrunding

I dette kapitel har vi søgt at forklare de helt grundlæggende præmisser og begreber for kanalkodningen.

Helt overordnet er der to modstridende interesser, som er gældende for koderne. Man ønsker på den ene side at benytte en kode med en høj rate, sådan at de beskeder, der transmitteres, indeholder meget information i forhold til redundans. På den anden side er der i mange sammenhænge et ønske om at fremme kodernes fejlkorrigerende egenskaber. Dette gøres ved at øge kodernes bloklængde, men herved formindsker (forringer) man kodernes rater. En stor del af forskningen inden for kodningsteori består af den grund i at finde kanalkoder, som udgør et godt kompromis mellem rate og fejlkorrigering.

I de følgende kapitler vil vi forsøge at matematisere ovenstående diskussion. Vi skal i næste kapitel se på en klasse af koder, der kaldes lineære koder. Afslutningsvis skal lige nævnes, at alle koder, som vi indtil videre har præsenteret, er lineære.

7 Lineære koder

> There was a unifying goal in coding theory and now that this is achieved it is not entirely clear what is left. (Kötter; 2005, side 27 i INTERVIEWS)

Hvis man ønsker at forstå de koder, der anvendes i rummissionerne, er man nødt til at kende til en række grundlæggende begreber og klasser af koder inden for kodningsteorien. I dette kapitel vil vi præsentere og forklare disse begreber og koder nærmere. Fremstillingen foregår oppefra og ned forstået på den måde, at vi indleder med at præsentere en meget generel og grundlæggende klasse af koder, de *lineære* koder. Derefter indsnævrer vi vores fremstilling til en særlig delmængde af de lineære koder, de *cykliske* koder. I næste kapitel vil vi føre denne indsnævring videre og indføre *Reed-Solomon* koder.

En effektiv og brugbar kodning af en mængde data kræver på et helt overordnet plan en kode samt algoritmer til indkodning og afkodning af data. Som vi skal se i dette kapitel er indkodning af data ligetil for de koder, vi præsenterer. Effektive afkodningsalgoritmer er imidlertid ikke så simple at beskrive. Ej heller kan man pege på en enkelt algoritme som den bedste. Hvilken algoritme, der er bedst, afhænger i høj grad af de konkrete koder og anvendelser. Af samme årsag har søgning efter effektive afkodningsalgoritmer udgjort et stort forskningsområde inden for kodningsteorien, og der findes derfor et stort antal forskellige algoritmer. Foreløbig vil vi forbigå afkodningen og nøjes med at koncentrere os om kodernes struktur samt indkodningen af data.

Den følgende fremstilling er baseret på (Roman; 1997, side 89-208) samt (Blahut; 2003, side 67-130)

7.1 Grundlæggende for lineære koder

Vi vil indlede beskrivelsen af lineære koder med at give den abstrakte matematiske definition af begrebet.

Definition 7.1 (Lineær kode)
En lineær kode $\mathcal{C}$ er et underrum af $\mathbb{F}_q^n$.

I ovenstående definition udtrykkes klart, at der tales om en *lineær* kode, men da vi kun skal se på lineære koder, vil vi i det følgende undlade at angive, at koderne er lineære.

Et element $c \in \mathcal{C}$ kaldes et *kodeord*. En kode består altså af kodeord, som er n-tupler over et endeligt legeme $\mathbb{F}_q$. Størrelsen n kaldes kodens *bloklængde*. At koden udgør et underrum betyder, at kodeordene opfylder følgende:

1. Hvis c_1 og c_2 er kodeord, så er også $c_1 + c_2$ et kodeord.

2. Hvis c er et kodeord, og a er et element i $\mathbb{F}_q$, da er ac et kodeord.

Underrumsbeskrivelsen af en kode betyder, at vi kan tale om dimensionen k af koden. Hvis $\{b_1, b_2, \ldots b_k\}$ udgør en basis for $\mathcal{C}$, siges koden at have dimension k. En kode med bloklængde n og dimension k kaldes en (n,k)-kode. Underrumsbetingelsen betyder, at en (n,k)-kode over $\mathbb{F}_q$ indeholder præcist q^k kodeord.

Når vi ønsker at karakterisere en kode, er det ikke nødvendigt at opskrive alle kodens kodeord. I stedet kan vi udnytte at koden udgør et underrum og nøjes med at angive basisvektorerne. Dette forhold motiverer følgende definition:

Definition 7.2
Lad $\mathcal{C}$ være en (n,k)-kode over $\mathbb{F}_q$. Hvis $\{c_1, c_2, \ldots, c_k\}$ udgør en basis for $\mathcal{C}$, og $c_i = c_{i1}c_{i2}\ldots c_{in}$, så kaldes $k \times n$-matricen G defineret ved

$$G = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \ldots & c_{kn} \end{pmatrix},$$

for en generatormatrix for $\mathcal{C}$.

Generatormatricen giver en simpel metode til indkodning af dataord.

Definition 7.3
Lad $\mathcal{C}$ være en (n,k)-kode over $\mathbb{F}_q$ og lad $\mathcal{D}$ være en mængde bestående af q^k dataord. En bijektion mellem $\mathcal{D}$ og $\mathcal{C}$ kaldes en indkodning af elementerne fra $\mathcal{D}$.

Enhver bijektion kan benyttes til indkodning, men generatormatricen giver mulighed for at anvende en særlig simpel bijektion:

1. Opskriv alle k-tupler over $\mathbb{F}_q$.
2. Par tuplerne med dataordene således, at hvert dataord d svarer til en tupel a_d.
3. Indkodningen c_d af dataordet d er da givet ved:

   $$c_d = a_d G,$$

   hvor G er en generatormatrix for koden.

Dataordene er ikke så praktiske at arbejde med matematisk og parringen af dataord og tupler er normalt underforstået. Derfor vil vi i det følgende, når vi nævner dataord, tænke på de tilhørende tupler.

Sætning 7.4
Afbildningen $c_d = a_d G$ er en bijektion.

Bevis
Rækkerne i G udgør en basis $\mathcal{B}$ for $\mathcal{C}$. Læg mærke til, at a_d blot er c_d's repræsentation i forhold til $\mathcal{B}$. Heraf følger bijektiviteten. □

For enhver $k \times n$-matrix G over $\mathbb{F}_q$ med lineært uafhængige rækker gælder, at matricen er en generatormatrix for en kode $\mathcal{C}$, hvis kodeord kan bestemmes ved at multiplicere k-tupler over $\mathbb{F}_q$ med G.

Eksempel 7.5
Lad $\mathcal{C}$ være en $(7,4)$-Hammingkode over $\mathbb{F}_2$, det vil sige en *binær* kode med generatormatrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{7.1}$$

og lad $a = (1\ 0\ 1\ 1)$ være et dataord. Det til a hørende kodeord c er da givet ved

$$c = aG = (1\ 0\ 1\ 1) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} = (1\ 0\ 1\ 1\ 1\ 0\ 0),$$

◇

Ovenstående generatormatrix er på formen $G = (I \quad P)$, hvor I er en $k \times k$-enhedsmatrix. Denne form kaldes *systematisk* form. Det kan vises, at enhver lineær blokkode er ækvivalent med en kode, hvis generatormatrix er på systematisk form (Blahut; 2003, side 53).

Vi har nu set, at en lineær blokkode kan karakteriseres ved kodens generatormatrix G. Der findes også en anden matrix, som knyttes sammen med koden. Lad $\mathcal{C}$ være en (n,k) lineær kode. Så udgør $\mathcal{C}$ et underrum. Til underrummet $\mathcal{C}$ hører et ortogonalt komplement $\mathcal{C}^\perp$, som også udgør et underrum og dermed en lineær kode. Denne kode kaldes den *duale* kode til $\mathcal{C}$.

Definition 7.6 (Paritetscheck-matrix)
Lad $\mathcal{C}$ være en (n,k) lineær kode og lad $c_1, c_2, \ldots, c_{n-k}$ være $n-k$ basisvektorer for $\mathcal{C}^\perp$. Matricen H hvis i'te række er lig c_i kaldes da en Paritetscheck-matrix *for $\mathcal{C}$.*

For et kodeord $c \in \mathcal{C}$ gælder, at c er ortogonal med hver række i H og dermed

$$cH^T = \begin{pmatrix} \langle c, c_1 \rangle \\ \langle c, c_2 \rangle \\ \vdots \\ \langle c, c_{n-k} \rangle \end{pmatrix} = 0,$$

hvor $\langle \cdot\ , \cdot \rangle$ angiver indre produkt[1] defineret ved $\langle b, c \rangle = \sum_{i=0}^{n} b_i c_i$ for $b, c \in \mathbb{F}_q^n$. Paritetscheck-matricen giver en simpel metode til at undersøge, om en vektor c er et kodeord.

7.1.1 Singleton-grænsen

Lad $\mathcal{C}$ være en (n,k)-kode med en generatormatrix G på systematisk form og lad $a = (1, 0, 0, \ldots, 0)$ være et dataord med tilhørende kodeord $c = aG$.

1 Egentlig er der ikke tale om et indre produkt, da afbildningen kun er positiv semidefinit. Vi vil alligevel benytte afbildningen som et indre produkt.

De første k komponenter i c er da lig a og indeholder derfor $k-1$ nuller. Denne observation leder direkte til en vigtig grænse for minimumsafstanden for en kode.

Sætning 7.7 (Singleton-grænsen)
Minimumsafstanden for en lineær (n,k)-kode, $\mathcal{C}$, opfylder

$$d_{\min} \leq n - k + 1.$$

Bevis
Da $\mathcal{C}$ er et underrum gælder, at hvis $c \in \mathcal{C}$, så er

$$c + \mathcal{C} = \mathcal{C}.$$

Lad c_1, c_2 være de kodeord, der ligger tættest i $\mathcal{C}$ og dan koden

$$\mathcal{C}^* = \{c - c_1 | c \in \mathcal{C}\}.$$

Minimumsafstanden for $\mathcal{C}^*$ er lig med afstanden fra nulkodeordet til $\tilde{c} = c_2 - c_1$, det vil sige lig med den mindste vægt for et kodeord i $\mathcal{C}^*$. Da der imidlertid gælder, at $\mathcal{C}^* = \mathcal{C}$ gælder det samme for $\mathcal{C}$.

Vi så før, at vi kan indkode et dataord, sådan at det tilhørende kodeord har $k-1$ komponenter lig nul. Mindstevægten for et kodeord kan derfor højest være $n-(k-1) = n-k+1$. Dermed er minimumsafstanden ligeledes lig $n-k+1$. □

En kode med $d_{\min} = n - k + 1$ har altså størst mulig minimumsafstand svarende til at kodeordene ligger i underrummet på en sådan måde, at der er størst mulig afstand imellem dem. En sådan kode siges at have *maksimal afstand*. I næste kapitel skal vi se på en særlig klasse af sådanne koder – Reed-Solomon koderne.

7.2 Cykliske koder

Som sagt udgør de lineære koder en meget stor og generel samling af koder, som vi vil indsnævre lidt i dette afsnit, i hvilket vi skal betragte de *cykliske* koder. En cyklisk kode er en lineær kode med en række yderligere egenskaber, som kan beskrives i en mere generel algebraisk ramme. De cykliske koders egenskaber gør det muligt at benytte simple og hurtige algoritmer til ind- og afkodning, hvilket gør koderne interessante. Vi vil indføre koderne med underrumstilgangen og derfra gå over til at se dem som algebraiske strukturer.

Definition 7.8 (Cyklisk kode)
En lineær kode $\mathcal{C}$ er cyklisk, *hvis den opfylder, at når $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$ så også $c' = (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$.*

Givet et kodeord c kan der altså opnås et nyt kodeord ved at flytte komponenterne i c én plads mod højre. Denne procedure kaldes i denne sammenhæng et *cyklisk skift*, og svarer egentlig til en rotation (se definition A.8 i appendiks A). En cyklisk kode udgør et *cyklisk underrum*. Koden fra eksempel 7.5 er rent faktisk en cyklisk kode.

Umiddelbart er det ikke åbenlyst at kodens cykliske struktur skulle give den nogle fordele frem for andre koder, og vi er da også nødt til at indføre koden algebraisk før dette bliver klart. Vi vil begynde med at se lidt på ringen $\mathbb{F}_q[x]/\langle x^n-1\rangle$ bestående af alle polynomier over $\mathbb{F}_q$ reduceret modulo x^n-1. I denne ring er

$$\begin{aligned} x^n &= 1, \\ x^{n+1} &= x, \\ x^{n+2} &= x^2 \\ &\vdots \end{aligned}$$

hvilket betyder, at ethvert polynomium $p(x)$ over $\mathbb{F}_q$ har en *principal*[2] repræsentant $f(x)$ i ringen.

Ethvert polynomium

$$p(x) = \sum_{i=0}^{n-1} p_i x^i$$

i ringen kan opfattes som en vektor $p \in \mathbb{F}_q^n$ givet ved

$$p = (p_0, p_1, \ldots, p_{n-1}).$$

Polynomiet $\tilde{p}(x) = xp(x)$ er da givet ved:

$$\begin{aligned} \tilde{p}(x) = xp(x) &= \sum_{i=0}^{n-1} p_i x^{i+1} \\ &= p_{n-1}x^n + \sum_{i=0}^{n-2} p_i x^{i+1} \\ &= p_{n-1} + p_0 x + \ldots + p_{n-2}x^{n-1}, \end{aligned}$$

hvoraf ses, at $\tilde{p}_0 = p_{n-1}$ og $\tilde{p}_i = p_{i-1}$ for $i = 0, \ldots, n-2$. Multiplikationen $x \cdot p(x)$ svarer altså til at flytte alle komponenterne i p én plads mod højre.

Lad $\mathcal{C}$ være en cyklisk kode. Hvis vi opfatter kodeordene i $\mathcal{C}$ som elementer i $\mathbb{F}_q[x]/\langle x^n-1\rangle$, kan vi beskrive det cykliske skift af et kodeord c ved multiplikationen $x \cdot c(x)$. Dette forhold antyder, at det kunne være interessant med en algebraisk tilgang til de cykliske koder.

Eksempel 7.9
Lad $\mathcal{C}$ være en $(6,k)$-kode over $\mathbb{F}_2$ og lad c være kodeordet

$$c = (1\ 0\ 0\ 1\ 1\ 1).$$

Til c svarer da polynomiet

$$c(x) = x^5 + x^4 + x^3 + 1.$$

Polynomiet

$$\tilde{c}(x) = x \cdot c(x) = x^5 + x^4 + x + 1$$

svarer da til kodeordet

$$\tilde{c} = (1\ 1\ 0\ 0\ 1\ 1),$$

som er det kodeord, der fremkommer, når der foretages et cyklisk skift på c. ◇

2 En principal repræsentant $f(x)$ opfylder $\text{grad}(f) < n$.

For at opdage hvad ovenstående betyder for kodens struktur, er vi nødt til at indføre endnu et begreb.

Definition 7.10 (Ideal)
Lad R være en kommutativ ring. En delmængde S af R kaldes et ideal hvis

1. *Når $a, b \in S$ gælder, at $a + b \in S$.*
2. *For alle $r \in R$ og $a \in S$ gælder, at $ra \in S$.*

Delmængden S er altså lukket overfor addition og lukket overfor multiplikation *med elementer fra* R. Hvis $a \in S$, så kaldes mængden $\langle a \rangle = \{ra | r \in R\}$ for idealet *genereret af* a.

Vi kan nu karakterisere de cykliske koder på en anden måde.

Sætning 7.11
En lineær kode $\mathcal{C}$ er cyklisk, hvis og kun hvis den svarer til et ideal i $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$.

Bevis
Antag, at $\mathcal{C}$ er en cyklisk kode, som er repræsenteret ved en delmængde af $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$. Da $\mathcal{C}$ er lineær gælder, at

$$a(x), b(x) \in \mathcal{C} \Rightarrow a(x) + b(x) \in \mathcal{C}.$$

Da koden er lukket overfor cykliske skift gælder, at

$$a(x) \in \mathcal{C} \Rightarrow x \cdot a(x) \in \mathcal{C}$$

og endvidere $x^i \cdot a(x) \in \mathcal{C}$, det vil sige, at $\mathcal{C}$ er lukket overfor multiplikation med x og med potenser af x. Da $\mathcal{C}$ er lineær er $\mathcal{C}$ tilmed lukket overfor multiplikation med linearkombinationer af potenser af x, det vil sige $\mathcal{C}$ er lukket overfor multiplikation med et vilkårligt polynomium i $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$. Kravene til et ideal er altså opfyldt af $\mathcal{C}$.

Antag modsat, at $\mathcal{C}$ er et ideal i $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$. Så er $\mathcal{C}$ lukket for addition. Koden er desuden lukket for multiplikation med elementer fra $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$, herunder skalarer og x, og der gælder at

$$a(x) \in \mathcal{C} \Rightarrow x \cdot a(x) \in \mathcal{C}.$$

Ovenstående viser, at $\mathcal{C}$ er lukket overfor cykliske skift og dermed, at $\mathcal{C}$ er en cyklisk kode. □

Med udgangspunkt i det foregående kan vi give en algebraisk definition af en cyklisk kode:

Definition 7.12
En cyklisk kode $\mathcal{C}$ med bloklængde n er et ideal i ringen $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$.

Den algebraiske ramme vil vi udnytte til at se nærmere på koderne.

7.2.1 Egenskaber for cykliske koder

Vi vil indlede dette afsnit med en sætning, som opsummerer en række vigtige egenskaber ved cykliske koder.

Sætning 7.13
Lad $\mathcal{C}$ være en cyklisk kode med bloklængde n. Da gælder:

§1. Der findes et entydigt monisk polynomium g af mindste grad i $\mathcal{C}$. Dette polynomium genererer $\mathcal{C}$, det vil sige $\langle g \rangle = \mathcal{C}$. Polynomiet g kaldes for generatorpolynomiet *for $\mathcal{C}$.*

§2. Generatorpolynomiet g er en divisor i $x^n - 1$.

§3. Hvis $\text{grad}(g) = r$ så har $\mathcal{C}$ dimensionen $k = n - r$, og endvidere gælder $\mathcal{C} = \{q(x)g(x) \mid \text{grad}(q) < n - r\}$.

§4. Hvis $g(x) = g_0 + g_1 x + g_2 x^2 + \ldots + g_r x^r$, så har $\mathcal{C}$ generatormatrix

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \ldots & g_r & 0 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & g_2 & \ldots & g_r & 0 & \ldots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \ldots & g_r & \ldots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & g_0 & g_1 & g_2 & \ldots & g_r \end{pmatrix}.$$

Bevis
Beviset inddeles i fire trin, der svarer til de fire paragraffer:

§1. Vælg et polynomium med lavest grad i $\mathcal{C}$. Dette er muligt, da $\mathcal{C}$ indeholder et endeligt antal polynomier. Multiplicer polynomiet med et element fra $\mathbb{F}_q$, så polynomiet bliver monisk og kald det resulterende polynomium g. Dette moniske polynomium har mindst grad.
Antag nu, at $\mathcal{C}$ indeholder *to* moniske polynomier g_1, g_2 af mindste grad r. Da $\mathcal{C}$ er lineær gælder, at $\tilde{g}(x) = g_1 - g_2 \in \mathcal{C}$. Da g_1 og g_2 er moniske gælder, at $\text{grad}(\tilde{g}) < r$, hvilket er en modstrid, da g_1, g_2 har mindste grad. Altså er g et entydigt polynomium.
Da $\mathcal{C}$ er en cyklisk kode og dermed et ideal gælder, at $\langle g \rangle \subset \mathcal{C}$. Antag omvendt, at $p \in \mathcal{C}$ og lad $p(x) = Q(x)g(x) + s(x)$ med $\text{grad}(s) < r$. Da $\langle g \rangle \subset \mathcal{C}$ gælder $Q(x)g(x) \in \mathcal{C}$ og da $\mathcal{C}$ er et underrum gælder derfor også, at $s(x) = p(x) - Q(x)g(x) \in \mathcal{C}$. Dette medfører imidlertid, at $s(x) = 0$ da $\text{grad}(s) < \text{grad}(g)$. Altså er $p(x) = Q(x)g(x)$ og dermed $p \in \langle g \rangle$. Samlet fås, at $\langle g \rangle = \mathcal{C}$.

§2. Lad

$$x^n - 1 = Q(x)g(x) + s(x),$$

med $\text{grad}(s) < \text{grad}(g)$. I ringen $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$ er venstresiden lig nul hvilket medfører, at $s(x) = -Q(x)g(x)$, det vil sige, at $s(x)$ er et kodeord. Heraf følger, da $\text{grad}(s) < \text{grad}(g)$, at $s(x) = 0$ og dermed, at $x^n - 1 = Q(x)g(x)$.

§3. Vi viser sidste del først. Det skal vises, at for idealet

$$\langle g \rangle = \{f(x)g(x) \mid f(x) \in \mathbb{F}_q[x]/\langle x^n - 1 \rangle\}$$

er det nok at se på f'er af grad mindre end $n - r$. Vi bemærker først, at der findes et polynomium $h(x)$ med $\text{grad}(h) = n - r$, så

$x^n - 1 = h(x)g(x)$. Lad nu $\text{grad}(f) \geq n - r$. Så har vi ifølge divisionsalgoritmen fra grundlæggende algebra, at

$$f(x) = Q(x)h(x) + s(x),$$

hvor $\text{grad}(s) < \text{grad}(h) = n - r$. Endvidere har vi, at

$$\begin{aligned} f(x)g(x) &= (Q(x)h(x) + s(x))g(x) \\ &= Q(x)h(x)g(x) + s(x)g(x) \\ &= Q(x)(x^n - 1) + s(x)g(x). \end{aligned}$$

I ringen $\mathbb{F}_q[x]/\langle x^n - 1\rangle$ er $Q(x)(x^n - 1) = 0$ og dermed $f(x)g(x) = s(x)g(x)$. Da $\text{grad}(s) < n - r$ følger, at det er tilstrækkeligt at multiplicere $g(x)$ med polynomier af grad mindre end $n - r$, når man danner idealet (koden).
Lad $p(x) \in \mathcal{C}$. Så er

$$p(x) = a(x)g(x) = a_0 g(x) + a_1 x g(x) + \ldots + a_{n-r-1}x^{n-r-1}g(x),$$

det vil sige, at $\{g(x), xg(x), \ldots x^{n-r-1}g(x)\}$ udspænder $\mathcal{C}$. Da

$$a_0 g(x) + a_1 x g(x) + \ldots + a_{n-r-1}x^{n-r-1}g(x) = 0$$

medfører, at a_i'erne alle er nul, er vektorerne lineært uafhængige og

$$\{g(x), xg(x), \ldots x^{n-r-1}g(x)\}$$

udgør derfor en basis for $\mathcal{C}$. Altså $\dim(\mathcal{C}) = n - r$.

§4. Da $\{g(x), xg(x), \ldots, x^{n-r-1}g(x)\}$ udgør en basis for $\mathcal{C}$ er rækkerne i G lineært uafhængige, og G er dermed en generatormatrix for koden.

□

Som tidligere beskrevet kan vi vælge at benytte generatormatricen til at indkode dataord på simpel vis. Sætning 7.13 ovenfor giver en anden metode. Ifølge §1 kan vi nemlig i stedet benytte generatorpolynomiet til indkodning. Indkodningen af et dataord a foregår så ved indkodning af det til a hørende datapolynomium $a(x)$ givet ved $a(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}$. Til dette datapolynomium knyttes kodepolynomiet $c(x)$ givet ved $a(x)g(x)$. I stedet for at multiplicere matricer kan vi altså multiplicere polynomier.

Tidligere i kapitlet så vi, at enhver $k \times n$-matrix med lineært uafhængige rækker definerer en lineær (n, k)-blokkode. Sætning 7.13 §2 kan let udvides sådan, at vi får et lignende resultat for cykliske koder.

Sætning 7.14
Der findes en cyklisk kode med bloklængde n og generatorpolynomium $g(x)$, hvis og kun hvis $g(x)$ er en faktor i $x^n - 1$.

Bevis
Den ene vej viste vi i sætning 7.13. Den anden vej vises her.

Antag, at $g(x)$ er en faktor i $x^n - 1$. Så findes $Q(x)$ så $x^n - 1 = g(x)Q(x)$. Koden $\mathcal{C} = \langle g(x)\rangle$ er tydeligvis en cyklisk kode, men vi skal vise, at $g(x)$ er generatorpolynomium for koden. Lad $s(x) \in \mathcal{C}$, det vil sige

$$s(x) \equiv a(x)g(x) \bmod x^n - 1,$$

hvor $a(x) \in \mathbb{F}_q[x]/\langle x^n - 1\rangle$. Antag, at $\text{grad}(s) < \text{grad}(g)$. Så gælder, at

$$s(x)Q(x) \equiv a(x)g(x)Q(x) \equiv a(x)(x^n - 1) \bmod x^n - 1,$$

og da $\text{grad}(s) < \text{grad}(g)$ er $\text{grad}(sQ) < n$. Det eneste polynomium p af grad mindre end n, hvor

$$p \equiv a(x)(x^n - 1) \equiv 0 \bmod x^n - 1$$

er imidlertid nulpolynomiet, altså må $s(x) = 0$. Hvis $\text{grad}(s) = \text{grad}(g)$ er $a(x) = 1$, det vil sige, at $s(x) = g(x)$. Altså er $g(x)$ polynomiet med lavest grad i $\mathcal{C}$ og dermed generatorpolynomium. □

Vi har altså, at ethvert monisk polynomium, som går op i $x^n - 1$, er generatorpolynomium for en cyklisk kode. Ifølge sætning 7.13 §3 har denne kode dimension $k = n - r$, hvor $r = \text{grad}(g)$.

§4 i samme sætning etablerer en sammenkædning mellem dette kapitels indledende underrumsbeskrivelse og den algebraiske beskrivelse, idet paragraffen gør det muligt at bestemme en generatormatrix for koden, når generatorpolynomiet kendes.

Eksempel 7.15
Lad $g(x) = x^3 + x + 1$ være et polynomium over $\mathbb{F}_2$. Da

$$\begin{aligned} g(x)(x^4 + x^2 + x + 1) &= (x^3 + x + 1)(x^4 + x^2 + x + 1) \\ &= x^7 + x^5 + x^4 + x^3 + x^5 + x^3 \\ &\quad + x^2 + x + x^4 + x^2 + x + 1 \\ &= x^7 + 1, \end{aligned}$$

som i $\mathbb{F}_2$ er lig $x^7 - 1$ gælder, at $g(x)$ er generatorpolynomium for en cyklisk kode med dimension $7-3 = 4$. Bemærk til senere, at $g(x)$ ikke har nogen rødder i $\mathbb{F}_2$. Generatormatricen for koden er ifølge sætning 7.13 §4 givet ved

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Koden $\mathcal{C}$ med ovenstående generatormatrix er som tidligere beskrevet underrummet udspændt af G's rækkevektorer. Fra lineær algebra (eller (Blahut; 2003)) ved vi, at et sådant underrum er upåvirket af elementære rækkeoperationer på G, hvorfor vi kan bringe denne på systematisk form. Ved sådanne operationer kan vi opnå

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} r_1 + r_2 + r_3 \\ r_2 + r_3 + r_4 \\ r_3 + r_4 \end{matrix},$$

hvilket ses at være generatormatrix for koden fra eksempel 7.5. Med andre ord er der tale om den samme kode i de to eksempler. ◇

7.3 Afrunding

I dette kapitel har vi indført og kort beskrevet lineære koder samt cykliske koder. Disse koder udgør en stor klasse af koder med mange forskellige egenskaber, og vi har derfor indtil videre ikke analyseret disse så dybtgående.

Vi har vist, hvordan man på simpel vis kan konstruere en cyklisk kode og indkode denne ved hjælp af kodens generatorpolynomium. Som antydet i eksempel 7.15 spiller generatorpolynomiets rødder en rolle for kodens struktur og egenskaber. I næste kapitel skal vi se hvilken.

På spørgsmålet om hvorvidt ikke-lineære koder forefindes i anvendelser svarede Tom Høholdt:

> Nej. No way. Det er teori. Men en charmerende teori, for nogen gange kan man lave ikke-lineære koder med dobbelt så mange kodeord som en lineær kode. Men der er ikke nogen, der bruger dem til noget, det er alt for kompliceret. (Høholdt; 2004, side 2 i Interviews)

For vores speciale synes det altså rimeligt at nøjes med at betragte lineære koder.

8 Reed-Solomon koder

> This technique has been used for thirty years and there are still missions signing up today that are planning to use concatenated Reed-Solomon codes. (Hamkins; 2005, side 49 i Interviews)

I dette kapitel vil vi indsnævre vores interesser yderligere og betragte på en særlig klasse af cykliske koder kaldet *primitive Reed-Solomon* koder. I forhold til de cykliske koder generelt har de primitive Reed-Solomon koder mere struktur, hvilket gør dem interessante matematisk set. Vi har derfor valgt at præsentere koderne ved flere forskellige tilgange og gennemgå dem i større detalje end sidste kapitels koder. For at spare unødige gentagelser vil vi blot omtale koderne som Reed-Solomon koder uden at nævne at der rent faktisk er tale om et specialtilfælde, som har en særlig *primitiv* struktur.

Kapitlet er baseret på et udvalg af kilder, som angives i begyndelsen af hvert afsnit.

8.1 Reed-Solomon koder som cykliske koder

Vi vil til at begynde med indføre Reed-Solomon koderne som cykliske koder. Fremstillingen er inspireret af (Roman; 1992, side 369-382).

Definition 8.1
Lad α være et primitivt element i legemet $\mathbb{F}_q$. En Reed-Solomon kode $\mathcal{C}$ *er en cyklisk kode med bloklængde $n = q - 1$ og generatorpolynomium*

$$g(x) = \prod_{i=b}^{r-1+b} (x - \alpha^i),$$

hvor $b \geq 0$ og $r < n$.

I ovenstående definition har vi krævet, at bloklængden skal være præcis $q - 1$. Når dette krav stilles kaldes koden en *primitiv* Reed-Solomon kode. Af definitionen ses, at generatorpolynomiet bliver monisk, hvilket jo ifølge sætning 7.13 §1 var et krav til en cyklisk kode. Parameteren r bestemmer graden af generatorpolynomiet. Parameteren b har størst betydning for implementering af koden.

Generatorpolynomiet for koden i eksempel 7.15 har ingen rødder i det legeme, polynomiet er defineret over. Af definitionen ovenfor ses, at Reed-Solomon koderne er defineret således, at generatorpolynomiet har *alle* sine rødder i det legeme, der benyttes som kodens alfabet. I stedet for at specificere et polynomium og konstruere koden ved hjælp af dette kan vi i stedet specificere en samling af rødder og konstruere generatorpolynomiet herfra.

Eksempel 8.2
Betragt legemet $\mathbb{F}_8$ med følgende regneregler:

$+$	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

$\cdot$	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	5	7	1	3
3	0	3	6	5	1	2	7	4
4	0	4	5	1	7	3	2	6
5	0	5	7	2	3	6	4	1
6	0	6	1	7	2	4	3	5
7	0	7	3	4	6	1	5	2

Ved en simpel udregning kan vi se, at vi kan skrive legemets elementer forskellige fra nul som

$$(1, 2, 4, 5, 7, 3, 6) = (\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6),$$

hvor $\alpha = 2$. Det vil sige, at 2 er et primitiv element.

Lad os konstruere en $(7, 3)$-Reed-Solomon kode. Denne kode skal ifølge definitionen være defineret med $\mathbb{F}_8$ som alfabet og med et generatorpolynomium g af grad $7 - 3 = 4$, altså $r = 4$. Vi sætter $b = 0$ og definerer generatorpolynomiet g ved

$$\begin{aligned} g(x) &= \sum_{i=0}^{3}(x - 2^i) \\ &= (x-1)(x-2)(x-4)(x-5) \\ &= (x+1)(x+2)(x+4)(x+5) \\ &= (x^2+3x+2)(x^2+x+3) \\ &= x^4 + 2x^3 + 2x^2 + 7x + 6. \end{aligned}$$

Den resulterende kode er en kode med $8^3 = 512$ kodepolynomier, som svarer til g multipliceret med polynomier af grad 2 eller mindre over $\mathbb{F}_8$. $\diamond$

Resultaterne fra sætning 7.13 gælder naturligvis også for Reed-Solomon koderne, da de er cykliske.

8.2 En moderne tilgang til Reed-Solomon koder

Ovenfor så vi, hvordan en algebraisk tilgang til Reed-Solomon koderne gør det muligt at definere disse på en forholdsvis elegant måde. I dette afsnit vil vi kaste yderligere lys over koderne ved at vælge en tilgang, som går gennem Fouriertransformationen på $\mathbb{F}_q^n$. Denne del af gennemgangen er baseret på (Blahut; 2003, side 131-143).

I appendiks A har vi udforsket den diskrete Fouriertransformation på $\mathbb{C}^N$. I den forbindelse er givet en definition af transformationen (se ligning (A.1)), som involverer begrebet *enhedsrødder*. Vi vil benytte dette begreb til at indføre Fouriertransformationen på $\mathbb{F}_q^n$. Først defineres enhedsrødder i denne sammenhæng.

Definition 8.3
Lad $\mathbb{F}_q$ være et endeligt legeme, og lad $\mathbb{F}_Q$ være et udvidelseslegeme for

$\mathbb{F}_q$, hvor $p(x) = x^n - 1$ kan skrives som et produkt af lineære faktorer $(x - \beta_i)$, $i = 1, \ldots, n$. Rødderne β_i i p kaldes da n'te enhedsrødder over $\mathbb{F}_Q$.

Definitionen antyder, at vi ikke generelt kan være sikre på, at rødderne til polynomiet ligger i legemet selv. For $n = q - 1$ *kan* vi imidlertid være sikre, hvilket redegøres for herunder.

Lad $\mathbb{F}_q^*$ være elementerne i $\mathbb{F}_q$ forskellige fra nul. Elementer i $\mathbb{F}_q^*$ udgør da en cyklisk gruppe af orden $q - 1$ under multiplikation. Lad $a \in \mathbb{F}_q^*$. Så gælder, at $\text{ord}(a)$ går op i $q - 1$, det vil sige

$$a \in \mathbb{F}_q^* \quad \Rightarrow \quad a^{q-1} = a^{\text{ord}(a)k} = 1^k = 1.$$

Ovenstående betyder, at alle elementer a i $\mathbb{F}_q^*$ opfylder ligningen

$$a^{q-1} - 1 = 0,$$

hvilket vil sige, at de er rødder i polynomiet $x^{q-1} - 1$. Da $\mathbb{F}_q$ er et legeme kan dette polynomium højest have $q - 1$ rødder[1] og der gælder derfor, at

$$x^{q-1} - 1 = (x - a_1)(x - a_2) \ldots (x - a_{q-1}),$$

hvor a_i'erne betegner de $q - 1$ elementer i $\mathbb{F}_q^*$. Alle elementer er altså rødder i polynomiet $x^{q-1} - 1$ og dermed $(q - 1)$'te enhedsrødder. Dette resultat fortjener en sætning:

Sætning 8.4
Lad $\mathbb{F}_q$ være et endeligt legeme og lad $\omega \in \mathbb{F}_q^*$. Så gælder, at ω er en $q-1$'te enhedsrod.

Vi kan nu definere Fouriertransformationen på $\mathbb{F}_q$.

Definition 8.5
Lad $z \in \mathbb{F}_q^n$, $n = q - 1$ og lad $\omega \in \mathbb{F}_q$ være et primitivt element. Den Fouriertransformerede $\hat{z}$ af z er vektoren $\hat{z} = \big(\hat{z}(0), \ldots, \hat{z}(n-1)\big)$ med

$$\hat{z}(m) = \sum_{k=0}^{n-1} z(k)\omega^{mk}.$$

Fra sætning 8.4 følger, at når ω er et primitivt element og dermed tilhører F_q^*, så er ω også en n'te enhedsrod, hvorfor ovenstående definition stemmer overens med definitionen af transformationen i appendiks A. Den invers transformerede defineres ved

$$\check{z}(m) = \frac{1}{n} \sum_{k=0}^{n-1} \omega^{-mk} z(k),$$

hvilket også er i overensstemmelse med definitionen i appendiks A.

Med baggrund i ovenstående er vi nu klar til at definere Reed-Solomon koderne på endnu en måde:

1 Dette er ikke nødvendigvis tilfældet, hvis der ikke er tale om et legeme. Eksempelvis har polynomiet x^3 over ringen $\mathbb{Z}_8$ rødderne $0, 2, 4$ og 6.

Definition 8.6 (Reed-Solomon kode)
En Reed-Solomon kode $\mathcal{C}$ *over et endeligt legeme* $\mathbb{F}_q$ *er defineret som mængden* $\{c \in \mathbb{F}_q^{q-1}\}$ *hvis Fouriertransformerede* $\hat{c}$ *har* r *på hinanden følgende komponenter lig 0.*

At koden defineret ovenfor er et cyklisk underrum indses let: Lad c_1, c_2 være kodeord som ovenfor defineret og lad $a_1, a_2 \in \mathbb{F}_q$. Da Fouriertransformationen er lineær gælder, at for en komponent k lig nul i $\hat{c}_1, \hat{c}_2$ er

$$(a_1c_1 + a_2c_2)\hat{}(k) = a_1\hat{c}_1(k) + a_2\hat{c}_2(k) = 0,$$

hvoraf følger at koden udgør et underrum. Lad c være et kodeord. Så er

$$(R_kc)\hat{}(m) = \omega^k\hat{c}(m).$$

Hvis $\hat{c}(m) = 0$ gælder også $\omega^k\hat{c}(m) = 0$. Derfor gælder, at R_kc er et kodeord. Underrummet er altså cyklisk.

Den algebraiske tilgang til Reed-Solomon koderne og Fourier-tilgangen giver sammenlagt et effektivt analyseapparat, som vi vil benytte i det følgende. De to tilgange sammenknyttes elegant i følgende sætning:

Sætning 8.7
Lad $c \in \mathbb{F}_q^n$. *Om polynomiet* $c(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1}$ *gælder*

1. $c(x)$ *har* ω^k *som rod netop hvis* $\hat{c}(k) = 0$.
2. $\hat{c}(x)$ *har* ω^{-k} *som rod netop hvis* $c(k) = 0$.

Bevis

$$\begin{aligned} c(\omega^i) &= \sum_{j=0}^{q-2} c(j)\omega^{ji} = \hat{c}(i), \quad \text{og} \\ \hat{c}(\omega^{-i}) &= \sum_{j=0}^{q-2} \hat{c}(j)\omega^{-ji} = (q-1)c(i). \end{aligned}$$

□

Lad $\mathcal{C}$ være en Reed-Solomon kode defineret som i definition 8.6. Lad $k_0, \ldots, k_{r-1}$ være de r komponenter, som skal være 0 i de transformerede kodeord. Et kodeord c opfylder da

$$c(\omega^{k_i}) = 0,$$

for $i = 0, \ldots, r-1$. Fra den algebraiske tilgang så vi, at ethvert kodeord $c(x)$ i $\mathcal{C}$ kan skrives som

$$c(x) = a(x)g(x),$$

hvor g er kodens generatorpolynomium og a er et datapolynomium. Sættes $a(x) = 1$ får vi

$$c(x) = g(x)$$

og herfra sluttes, at

$$g(\omega^{k_i}) = 0.$$

Blokken bestående af r nuller giver altså rødderne i generatorpolynomiet, som kan skrives

$$g(x) = (x - \omega^{k_0})(x - \omega^{k_1}) \cdots (x - \omega^{k_{r-1}}). \tag{8.1}$$

Sætning 8.7 gør os altså i stand til at bestemme generatorpolynomiet for koden, hvis vi kender blokken med nuller for de transformerede kodeord og omvendt.

8.2.1 Minimumsafstanden for en Reed-Solomon kode

Den kombinerede tilgang gør det nemt at vise, at Reed-Solomon koderne har maksimal afstand.

Sætning 8.8
Lad $\mathcal{C}$ være en Reed-Solomon kode. Så har $\mathcal{C}$ minimumsafstand

$$d_{\min} = n - k + 1.$$

Bevis
Multiplicer koden med ω^j således at blokken med nuller i de Fouriertransformerede udgør de sidste r pladser. Dette er muligt ifølge sætning A.11 og ændrer ikke koden, da den udgør et underrum. Lad $\hat{c}$ være den transformerede til kodeordet c og lad $C(x)$ være polynomiet defineret ved $\hat{c}$'s koordinater. Så er

$$C(x) = \sum_{j=0}^{q-2-r} \hat{c}(j)x^j = \sum_{j=0}^{q-2} \hat{c}(j)x^j.$$

Den sidste lighed følger af at $\hat{c}(j) = 0$ for $j > q - 2 - r$. Det til $\hat{c}$ hørende kodeord c har komponenterne

$$c(m) = \frac{1}{q-1} \sum_{j=0}^{q-2} \hat{c}(j)\omega^j = \frac{1}{q-1} C(\omega^j).$$

Polynomiet $C(x)$ har grad $q - 2 - r$ eller mindre og kan derfor højest have $q - 2 - r$ rødder. Ifølge sætning 8.7 betyder dette, at c højst har $q - 2 - r$ komponenter, som er nul (ellers er c nulvektoren), det vil sige, c har *mindst* $r + 1$ komponenter forskellige fra nul. Minimumsafstanden for koden opfylder derfor

$$d_{\min} \geq r + 1 = n - k + 1.$$

Fra sætning 7.7 har vi, at

$$d_{\min} \leq n - k + 1.$$

Sammen viser de to uligheder at

$$d_{\min} = n - k + 1.$$

□

Eksempel 8.9
Lad $\mathcal{C}$ være en cyklisk kode med bloklængde 7 over $\mathbb{F}_8$ med samme regneregler som tidligere angivet. Vi ønsker at konstruere en Reed-Solomon kode, der kan rette to fejl. Koden skal altså have minimumsafstand 5. Vi vælger det primitive element 2 og vælger, at de første 4 pladser i de Fouriertransformerede skal være nul. Så er generatorpolynomiet for koden ifølge ligning (8.1) givet ved

$$g(x) = \sum_{i=0}^{3}(x - 2^i) = x^4 + 2x^3 + 2x^2 + 7x + 6,$$

hvilket tydeligvis er generatorpolynomiet fra eksempel 8.2. På figur 8.1 ses et udsnit af kodens 512 kodeord. ◇

$a(x)$	$c(x)$	Oktalord
1	$x^4 + 2x^3 + 2x^4 + 7x + 6$	0012276
2	$2x^4 + 4x^3 + 4x^4 + 3x + 1$	0024431
3	$3x^4 + 6x^3 + 6x^4 + 4x + 7$	0036647
$\vdots$	$\vdots$	$\vdots$
x	$x^5 + 2x^4 + 2x^3 + 7x^2 + 6x$	0122760
$x + 1$	$x^5 + 3x^4 + 5x^2 + x + 6$	0130516
$x + 2$	$x^5 + 4x^4 + 6x^3 + 3x^2 + 5x + 1$	0146351
$\vdots$	$\vdots$	$\vdots$
x^2	$x^6 + 2x^5 + 2x^4 + 7x^3 + 6x^2$	1227600
$x^2 + 1$	$x^6 + 2x^5 + 3x^4 + 5x^3 + 4x^2 + 7x + 6$	1235476
$x^2 + 2$	$x^6 + 2x^5 + 4x^4 + 3x^3 + 2x^2 + 3x + 1$	1243231
$\vdots$	$\vdots$	$\vdots$
$7x^2 + 7x + 5$	$7x^6 + 4x^5 + 5x^4 + 6x^3 + 4x + 4$	7456044
$7x^2 + 7x + 6$	$7x^6 + 4x^5 + 6x^4 + 6x^2 + 3$	7460603
$7x^2 + 7x + 7$	$7x^6 + 4x^5 + 7x^4 + 4x^3 + 5x^2 + 7x + 5$	7474575

Figur 8.1 Et udsnit af de 512 kodeord for Reed-Solomon koden i eksempel 8.9. I tabellen angiver $a(x)$ et datapolynomium og $c(x)$ angiver et kodepolynomium. Den sidste kolonne angiver kodepolynomiet opskrevet som en vektor over $\mathbb{F}_8$.

8.3 Reed-Solomon koder og bit

Indtil videre har vi nøjes med at interessere os for kodernes matematiske struktur og ikke interesseret os så meget for, hvordan de egentlig skal anvendes til at transmittere data i form af bit. En Reed-Solomon kode over $\mathbb{F}_q$ skal ifølge definition 8.1 have bloklængde $q - 1$ og en binær Reed-Solomon kode vil derfor have bloklængde 1, altså være koden bestående af kodeordene 0 og 1. En sådan kode er naturligvis ikke interessant i anvendelser.

For et endeligt legeme med karakteristik 2, altså $\mathbb{F}_{2^p}$, kan vi imidlertid udnytte, at alle legemets elementer kan opfattes som vektorer over $\mathbb{F}_2$ med p komponenter. Hvis vi opskriver koden fra eksemplet ovenfor med binær notation, opnår vi således en kode, med en bloklængde, der er 3 gange så

stor som den ikke-binære kodes bloklængde. På figur 8.2 er vist, hvordan kodeordene for koden fra eksempel 8.9 ser ud i denne notation.

Oktalord	Binært ord						
0012276	000	000	001	010	010	111	110
0024431	000	000	010	100	100	011	001
0036647	000	000	011	110	110	100	111
⋮				⋮			
0122760	000	001	010	010	111	110	000
0130516	000	001	011	000	101	001	110
0146351	000	001	100	110	011	101	001
⋮				⋮			
1227600	001	010	010	111	110	000	000
1235476	001	010	011	101	100	111	110
1243231	001	010	100	011	010	011	001
⋮				⋮			
7456044	111	100	101	110	000	100	100
7460603	111	100	110	000	110	000	011
7474575	111	100	111	100	101	111	101

Figur 8.2 Reed-Solomon koden fra eksempel 8.9 i binær notation. I stedet for at transmittere et oktalt ord, transmitteres det tilhørende binære ord. Når et binært ord modtages omformes det til oktal notation inden afkodning.

Kodeordene i figur 8.2 illustrerer en vigtig egenskab ved Reed-Solomon koderne. Hvis man transmitterer et af ovenstående binære kodeord, vil en enkelt bitfejl under transmissionen resultere i en ændring i et af de oktale symboler. Bitfejl, der er spredt jævnt ud over et kodeord, vil resultere i ændringer i mange oktale symboler, og disse ændringer vil højst sandsynligt betyde en afkodningsfejl. Hvis bitfejlene i stedet er ordnet i *bursts*, vil dette resultere i langt færre fejl i de oktale symboler, fordi flere bit falder inden for samme oktale symbol. Eksempelvis vil et burst på fire bit forårsage fejl i præcist to oktale symboler.

For en Reed-Solomon kode over $\mathbb{F}_{2^p}$ vil hvert element i legemet kræve p bit til repræsentation. Herfra er det simpelt at udlede, at et burst af længde m højest forårsager fejl i $1 + \lceil (m-1)/p \rceil$ symboler fra $\mathbb{F}_{2^p}$. Dette resultat fortjener en sætning:

Sætning 8.10
For en (n, k)-Reed-Solomon kode, med $n = 2^p - 1$ vil et burst af længde m højst forårsage fejl i

$$1 + \left\lceil \frac{m-1}{p} \right\rceil$$

symboler fra $\mathbb{F}_{2^p}$.

Bevis
Antag, at den første bit i burst'et ødelægger den sidste bit i et symbol. Dette giver det mest ødelæggende burst. De resterende $m-1$ bit ødelægger da $\lceil (m-1)/p \rceil$ symboler. I alt påvirkes $1 + \lceil (m-1)/p \rceil$ symboler. □

8.3.1 Et oplagt valg

I computersammenhænge foregår de fleste operationer ved manipulation af bytes. En byte består som bekendt af otte bit, og det er derfor oplagt at se på en Reed-Solomon kode defineret over et legeme med 2^8 elementer, $\mathbb{F}_{256}$. For en sådan kode vil hvert symbol i et kodeord blive repræsenteret ved en enkelt byte. MER-missionen benytter netop en kode over $\mathbb{F}_{256}$.

8.4 Afrunding

I dette kapitel har vi indført Reed-Solomon koder som en særlig klasse af cykliske koder, der er karakteriseret ved at generatorpolynomiets rødder ligger i det legeme, som koden er defineret over. Vi har desuden indført Reed-Solomon koderne gennem Fouriertransformationen og vist, hvordan man kan skifte mellem de to tilgange.

Konstruktion af en Reed-Solomon kode er let. Man vælger hvilke rødder generatorpolynomiet skal have, og så er koden fastlagt. Den konstruerede kode har flere egenskaber, som er ønskværdige i anvendelser: For det første har koden maksimal minimumsafstand, og for det andet er koden god til at håndtere burst-fejl. Reed-Solomon koderne forekommer derfor i mange forskellige anvendelser, herunder MER-missionen.

Irving S. Reed (1923-) og Gustave Solomon (1931-1996)

Irving S. Reed var oprindeligt uddannet inden for teoretisk fysik, men modtog i 1949 sin ph.d. i matematik fra Caltech. Hans opdagelse af en algebraisk struktur i Mullers koder – i dag kendt som Reed-Muller koder – førte til en ny afkodningsalgoritme for disse. I midten af 1950'erne arbejdede Reed med at udvikle automatiske digitale processorer til brug i radar-applikationer. Arbejdet med processorerne fik ham sidst i 50'erne til at overveje at udnytte endelige legemer til byteoperationer. Overvejelserne kulminerede i opdagelsen af Reed-Solomon koderne i 1959. Reeds arbejde i 60'erne med fejl-detektion i processorer fik betydning for JPL's design af computersystemerne til blandt andet Viking- og Voyager-missionerne og dermed for rumforskningen. Fra 1960-63 arbejdede han desuden for RAND Corporation, hvor han studerede missile forsvarssystemer. I nyere tid har Reed blandt andet arbejdet med udvikling af neurale netværk til afkodning af fejlkorrigerende koder. (Reed & Solomon; 1994, side 18-19, 22-24)

Gustave Solomon var matematiker med speciale i algebra og en ph.d. fra MIT. Sidst i 1950'erne blev Solomon introduceret til kodningsteorien af Reed. Deres arbejde sammen resulterede i Reed-Solomon koderne. Solomon arbejdede frem til 1961 ved Lincoln Laboratory, MIT, hvorefter han blev rekrutteret til JPL. Fra '67 til '89 arbejdede Solomon med udvikling af nye kommunikationssystemer til rumfartsindustrien dog afbrudt af midlertidige professorater ved UCLA og Caltech. Han fungerede i sine sidste leveår som konsulent indenfor digital kommunikation og undertiden også som rådgiver for de studerende ved Caltech. Indlæring lå Solomon dybt på sinde, og han gav privatundervisning i såvel tekniske aspekter som i formidling. Solomon udgav en række for kodningsteorien vigtige afhandlinger, blandt andet sammen med Mattson i 1961, McEliece i 1966 og med van Tilborg i 1977, hvor trellisafkodning af algebraiske koder blev introduceret. (Reed & Solomon; 1994, side 18-19, 22-24)

9 Foldningskoder

> The entire community took off into a different direction following convolutional codes and sequential decoding that was supposed to be the savior and make information theory practical. (Kötter; 2005, side 25 i INTERVIEWS)

Til kanalkodningen i MER-missionen anvendes som tidligere nævnt en konkateneret kode bestående af en ydre Reed-Solomon kode og en indre foldningskode. Foldningskoder er såkaldte *trelliskoder* i modsætning til de tidligere beskrevne koder, der alle går under betegnelsen blokkoder.

På trods af at foldningskoder har været kendt og studeret siden midten af 1950'erne, er der stadig mange aspekter af disse koder, der ikke er velforståede, og der findes i dag adskillige tilgange til teorien for disse koder (Justesen & Høholdt; 2004a, side 83). Af den grund har vi valgt at bringe et udvalg af disse forskellige tilgange. Store dele af den følgende fremstilling er baseret på (Blahut; 2003, side 270-312). Når der undervejs er benyttet andre kilder, vil dette blive angivet.

9.1 Fra blokkoder til foldningskoder

Den mest ligefremme måde at skelne blok- og foldningskoder fra hinanden er ved at sammenligne deres indkodere. Lad os for blokkodernes vedkommende kort rekapitulere: En indkoder for en lineær (n, k)-blokkode kan opfattes, som en en-til-en afbildning fra $\mathbb{F}_q^k$ ind i $\mathbb{F}_q^n$. Altså, hvis vi har en vektor $x \in \mathbb{F}_q^k$, som udgør et k-dimensionalt dataord, vil vi ved afbildningen opnå et n-dimensionalt kodeord, $y \in \mathbb{F}_q^n$. Hvis indkoderen præsenteres for en følge af dataord, $x_1, x_2, \ldots$, producerer den en følge af kodeord, $y_1, y_2, \ldots$. Vi kan tænke på x_i som dataordet sendt til kanalindkoderen til tid i, eller opfatte i som et 'tidsindex' for x_i.

En indkoder for en foldningskode kan ligeledes opfattes som en afbildning fra en følge af k-tupler, som kaldes *datarammer*, til en følge af n-tupler, som kaldes *koderammer*, det vil sige

$$\varphi : x_1, x_2, \ldots \rightarrow y_1, y_2, \ldots .$$

Bemærk, at man for foldningskoderne anvender betegnelserne dataramme og koderamme i stedet for dataord og kodeord. Dette gøres dels for at adskille blok- og foldningskoder fra hinanden, men også fordi foldningskodernes rammer typisk er betydeligt mindre end blokkodernes ord. Den væsentligste forskel på en blokkode og en foldningskode er, at for en foldningskode er den i'te koderamme y_i ikke kun en funktion af x_i, men også af et fastlagt antal af tidligere indkodede datarammer, $x_{i-1}, \ldots, x_{i-m}$. Heltallet m kaldes rammehukommelsen eller blot *hukommelsen* af indkoderen. Raten for en foldningskode er defineret ved forholdet k/n.

Fra det her givne synspunkt kan en blokkode betragtes som en foldningskode uden hukommelse, hvilket får teorien for blokkoder til at fremstå som et 'specialtilfælde' af teorien for foldningskoder. Til trods for at et sådant synspunkt kan være forståelsesmæssigt hjælpsomt rummer det ikke desto mindre en mulighed for misforståelse. Dette skyldes ifølge McEliece (1998), at blokkodning omfatter koder, hvor k og n er store heltal, hvorimod foldningskodning ofte handler om koder med små, fastlagte værdier af k og n. Da m for en foldningskode kan variere er studiet af foldningskoder med $n = 2$ og $k = 1$ på ingen måde trivielt, hvilket studiet af klassen af blokkoder med samme parametre derimod er. Ifølge Blahut (2003) er kun få konstruktive klasser af foldningskoder kendt i dag. De fleste er opdaget ved brug af computersøgninger. Der kendes ingen tilfredsstillende konstruktive metoder til at finde generelle klasser af gode foldningskoder. (McEliece; 1998, side 1067-1068), (Blahut; 2003, side 273)

Vi skal senere give en helt formel definition af en foldningskode, men først skal vi se på foldningsindkoderen.

9.2 Foldningsindkoderen

Vi skal i dette afsnit betragte indkoderen for en foldningskode. Indkoderen den *regel*, der definerer hvordan følgen af koderammer bestemmes udfra følgen af datarammer. Nedenstående beskrivelse af foldningsindkoderen er i et vist omfang baseret på (Forney, Jr.; 1970b, side 720-725).

Lad os introducere foldningsindkoderen med et simpelt eksempel.

Eksempel 9.1
Inddata til indkoderen er en 'følge' af binære datarammer

$$x = \ldots, x_{-1}, x_0, x_1, x_2, \ldots,$$

og uddata er de to følger af binære koderammer y_0 og y_1. Den første uddata-følge, y_0, er blot lig x. Elementerne y_{1i} af den anden uddata-følge, y_1, er givet ved

$$y_{1i} = x_i + x_{i-1} + x_{i-2},$$

hvor addition foregår modulo 2. Bemærk, at indkoderen skal huske (gemme) de to foregående rammer af datafølgen. (Forney, Jr.; 1970b, side 720-721) ◇

Ovenstående eksempel giver anledning til følgende generelle definition.

Definition 9.2 (Tvungen længde)
Antallet af gemte q-ære symboler fra datafølgen i indkoderen for en foldningskode kaldes den tvungne længde *af koden og betegnes ν.*

Da indkoderen beregner sine koderammer på basis af de foregående m datarammer, er det tilstrækkeligt kun at huske disse rammer. Det vil sige, at når hver ramme består af k symboler, er det tilstrækkeligt kun at huske mk symboler, hvilket indikerer at $\nu \leq mk$. (Blahut; 2003, side 272)

Til beskrivelse af foldningsindkodningen anvendes i litteraturen gerne frembringende funktioner[1] også kendt som z-transformationer, som gør

1 For en beskrivelse af frembringende funktioner se eventuelt Biggs (1989).

brug af den ubestemte z^{-1}. Til foldningskodning anvendes, hvad der undertiden kaldes $\mathcal{D}$-transformationen, da denne i stedet for z^{-1} gør brug af et D, som i denne sammenhæng kaldes en *forsinkelsesgenerator*[2]. Til en foldningsindkoders inddata-følger knyttes således rækker i $\mathcal{D}$-transformationen. En forsinkelsesoperator D^i angiver, at et q-ært symbol fra datafølgen er i rammer forsinket i forhold til det først indkodede symbol. (Justesen & Høholdt; 2004a, side 91), (McEliece; 1998, side 1069)

Eksempel 9.3
Med udgangspunkt i det ovenstående eksempel haves således

$$\begin{aligned} x(D) &= \ldots + x_{-1}D^{-1} + x_0 + x_1 D + x_2 D^2 + \ldots \\ y_0(D) &= \ldots + y_{0-1}D^{-1} + y_{00} + y_{01} D + y_{02} D^2 + \ldots \\ y_1(D) &= \ldots + y_{1-1}D^{-1} + y_{10} + y_{11} D + y_{12} D^2 + \ldots \end{aligned}$$

Forholdet mellem inddata og uddata kan nu udtrykkes som

$$\begin{aligned} y_0(D) &= g_0(D)x(D), \\ y_1(D) &= g_1(D)x(D), \end{aligned}$$

hvor $g_0(D)$ og $g_1(D)$ er generatorpolynomier givet ved

$$\begin{aligned} g_0(D) &= 1, \\ g_1(D) &= 1 + D + D^2. \end{aligned}$$

(Forney, Jr.; 1970b, side 721) ◇

Dette giver anledning til en definition af et *generatorpolynomium*.

Definition 9.4
Det ℓ'te generatorpolynomium *for en foldningskode har formen*

$$g_\ell(D) = g_{\ell 0} + g_{\ell 1} D + g_{\ell 2} D^2 + \ldots + g_{\ell m} D^m,$$

hvor alle operationer foregår modulo q. (Lin & Costello, Jr.; 1983, side 292-293)

Med udgangspunkt i ovenstående kan vi på nuværende tidspunkt give følgende definition af en foldningsindkoder.

Definition 9.5 (Foldningsindkoder)
Vi definerer en (n, k, m)-foldningsindkoder *ved en $k \times n$ matrix, $G(D)$, af generatorpolynomier $g_{ij}(D)$ af grad (højst) m med koefficienter fra et endeligt legeme $\mathbb{F}_q$. Forholdet mellem indkoderens ind- og uddata er givet ved*

$$y_j(D) = \sum_{i=0}^{k-1} x_i(D) g_{ij}(D),$$

hvor $x_i(D)$ udgør de k-dimensionale datarammer og $y_i(D)$ de n-dimensionale koderammer. Såvel data- som koderammer har symboler fra $\mathbb{F}_q$.

2 Fra engelsk: Delay operator. Den ubestemte D kan tænkes på som et element i et udvidelseslegeme af symbollegemet.

Eksempel 9.6
Generatormatricen $G(D)$ fra ovenstående eksempel er givet ved

$$G(D) = \begin{pmatrix} 1 & 1 + D + D^2 \end{pmatrix}.$$

$\diamond$

Vi skal undertiden i det følgende undertrykke D'et i referencerne til rammerne, altså $x_i = x_i(D)$, $y_i = y_i(D)$ og så videre, hvor det faktum, at et bogstav repræsenterer en ramme, fremgår klart af konteksten. Lad os nu se på hvilke krav, der stilles til en sådan foldningsindkoder.

Det kræves af en foldningsindkoder, at den opfylder en række bestemte egenskaber; den skal være *tidsinvariant*, den skal være *lineær*, og generatormatricen skal være *forsinkelsesfri* og *realiserbar*. Kravet om reliserbarhed skal vi vende tilbage til i næste afsnit, men de øvrige krav uddybes herunder.

Med tidsinvariant forstås, at hvis samtlige til indkoderen indgående datafølger forskubbes i tid, så skal samtlige udgående kodefølger tilsvarende forskubbes. Udtrykt ved forsinkelsesoperatoren D haves

$$G(D^n x) = D^n G(x).$$

Med linearitet forstås at en linearkombination af to datarammer har en koderamme, som er den samme linearkombination af koderammerne for de to datarammer. Det vil sige, at hvis x_1 og x_2 er to datarammer, gælder der

$$G(\alpha x_1 + \beta x_2) = \alpha G(x_1) + \beta G(x_2) \quad \alpha, \beta \in \mathbb{F}_q.$$

Er kravene om, at indkoderen er tidsinvariant og lineær ikke opfyldt, kaldes den resulterende kode blot en *trelliskode* – altså er en tidsinvariant, lineær trelliskode en foldningskode (Blahut; 2003, side 271).

Det kræves, at mindst et af generatorpolynomierne i $G(D)$ ikke har D som faktor, for hvis alle elementer af $G(D)$ havde D som faktor, ville D kunne faktoriseres ud af $G(D)$ og dermed blive betragtet som en forsinkelse på én tidsenhed eller ramme, også kaldet en *rammeforsinkelse*. Et generatorpolynomium hvorom gælder, at det ikke har D som faktor, kaldes et *forsinkelsesfrit polynomium*.

Definition 9.7
En generatormatrix, der indeholder et forsinkelsesfrit polynomium, kaldes en forsinkelsesfri generatormatrix*.*

Det tidligere givne eksempel på foldningsindkodning er en såkaldt *systematisk* indkodning. Ved en systematisk indkodning af en foldningskode forstås en indkodning, hvori hver dataramme optræder uændret som de første k symboler ud af de n symboler i den første koderamme. I en sådan systematisk indkodning er data synlige i den indkodede følge af symboler og kan tilmed aflæses direkte i det tilfælde, hvor fejl ikke forekommer under transmissionen. Bemærk, at da foldningskoden eksisterer uafhængigt af indkodningsmetoden, er der ikke tale om en systematisk foldningskode, men kun om en systematisk indkoder.

9.3 Polynomial beskrivelse af foldningskoder

Et matematisk grundlag for foldningskoderne kan opstilles ved hjælp af algebraen, hvorfor koderne også betragtes som tilhørende klassen af algebraiske koder. Vi skal i det følgende antyde indførelsen af foldningskoderne gennem Laurent-rækker, men i øvrigt begrænse os til et specialtilfælde af koderne. I overenstemmelse med konventionen på området[3] skal vi i det følgende tage udgangspunkt i $\mathcal{D}$-transformationen. Den formelle indførelse af foldningskoderne er delvist baseret på (McEliece; 1998, side 1068-1073).

En beskrivelse af foldningskodning kan opnås ved at lade en følge x af elementer (rammer) i en ring $\mathbb{O}$ være repræsenteret ved de tilsvarende uendelige formelle *Laurent-rækker*

$$x(D) = \sum_{i \in \mathbb{Z}} x_i D^i,$$

i det ubestemte symbol D (forsinkelsesoperatoren). Hvis vi nøjes med at se på de følger, som har $x_i = 0$ for $i < N \in \mathbb{Z}$, det vil sige

$$x(D) = \sum_{i \geq N}^{\infty} x_i D^i,$$

også kaldet *en-sidede Laurent-rækker*, fås en mængde som skrives $\mathbb{O}((D))$. En-sidede Laurent-rækker af den specielle form $i \geq 0$, det vil sige formelle potensrækker, kaldes *kausale Laurent-rækker*. Endelige kausale Laurent-rækker er *polynomier*. Vi har, at mængden af polynomier med koefficienter fra $\mathbb{O}$, kaldet $\mathbb{O}[D]$, på oplagt vis er indeholdt i $\mathbb{O}((D))$.

Vi kan udstyre $\mathbb{O}((D))$ som en ring således, at multiplikationen er en udvidelse af den oplagte multiplikation af polynomier[4]. Vi får, at

$$\left(\sum_{j \geq N_j}^{\infty} p_j D^j \right) \left(\sum_{l \geq N_l}^{\infty} q_l D^l \right) = \sum_{i \geq N_i}^{\infty} z_i D^i,$$

for vilkårlige $N_j, N_l, N_i \in \mathbb{Z}$, netop hvis

$$z_i = \sum_{j+l=i} p_j q_l \quad (z = p * q).$$

Hvis $\mathbb{O}$ er lig $\mathbb{F}^\ell$, hvor $\mathbb{F}$ er et legeme, så har vi identifikationen

$$\mathbb{F}^\ell((D)) = \mathbb{F}((D))^\ell$$

og i denne situation vil $\mathbb{F}((D))$ være et legeme.

Helt generelt kan en foldningsindkodning betragtes som en lineær afbildning

$$\begin{aligned} \varphi : \mathbb{F}_q((D))^k &\rightarrow \mathbb{F}_q((D))^n, \\ x(D) &\rightarrow y(D), \end{aligned}$$

og koden selv som et k-dimensionalt underrum af $\mathbb{F}_q((D))^n$. Vi skal imidlertid indsnævre dette en smule.

3 Se eventuelt Forney, Jr. (1970b) og McEliece (1998).
4 For yderligere beskrivelse se eventuelt (Fraleigh; 1999, side 319-320).

Med udgangspunkt i ovenstående situation, hvor $\mathbb{F}((D))$ er et legeme, vil mængden af rationale udtryk af formen $p(D)/q(D)$, hvor $p(D)$ og $q(D)$ er polynomier med koefficienter fra $\mathbb{F}$ (og $q(D) \neq 0$), selv udgøre et legeme, kaldet $\mathbb{F}(D)$. Altså

$$\mathbb{F}(D) = \left\{ \frac{p(D)}{q(D)} \middle| p(D), q(D) \in \mathbb{F}[D], q(D) \neq 0 \right\}.$$

Dette legeme er det kanoniske brøklegeme for integritetsområdet $\mathbb{F}[D]$. Antallet af koefficienter forskellige fra nul i en Laurent-række kaldes for *vægten* af rækken. Enhver Laurent-række af endelig vægt er af formen $p(D)/D^l$, hvor $p(D)$ er et polynomium og l er et ikke-negativt heltal. Ifølge (McEliece; 1998) kan vi begrænse vores studie af foldningskoderne til sådanne endelige Laurent-rækker, dels da der i praksis kun optræder koderammer med endelig vægt og dels da generatormatricer, som svarer til realiserbare indkodere (forklares nedenfor) er dem, hvis elementer er kausale og består af de ovenfor angivne rationale udtryk. Med udgangspunkt i dette defineres en foldningskode på følgende vis.

Definition 9.8 (Foldningskode)
En (n, k)-foldningskode $\mathcal{C}$ *er et* k*-dimensionalt underrum af* $\mathbb{F}_q(D)^n$.

En foldningskode indkodet med en (n, k, m)-foldningsindkoder kaldes for en (n, k, m)-*foldningskode*. En matrix, hvis elementer stammer fra $\mathbb{F}(D)$, kaldes en *rational matrix*. Vi begrænser generatormaticerne for foldningskoder til at være sådanne rationale matricer.

Enhver basis for det k-dimensionale underrum kan bruges til at definere en generatormatrix $G(D)$ ved at anvende basisvektorerne som rækker i $G(D)$. Det er kutyme at kræve, at en generatormatrix for en foldningskode udover at være rational er såvel forsinkelsesfri som realiserbar. Et element $p(D)/q(D)$ kaldes *realiserbart*, hvis D ikke er en faktor i $q(D)$.

Definition 9.9
En (rational) generatormatrix hvis elementer alle er realiserbare kaldes en realiserbar generatormatrix.

Kravet om realiserbarhed er automatisk opfyldt i kraft af ovenstående definition af $\mathbb{F}(D)$, men vi skal alligevel dvæle lidt herved. Grunden til kravet er, at hvis alle tællerne af elementerne i $G(D)$ har D som faktor, så kan D faktoriseres ud af $G(D)$ og dermed betragtes som en simpel rammeforsinkelse. Hvis alle nævnere i matricens elementer har D som faktor, kan D^{-1} faktoriseres ud af $G(D)$ og blive betragtet som et simpelt rammeforspring.

Det kan vises, at enhver foldningskode har en polynomial generatormatrix.

Sætning 9.10
Enhver foldningskode $\mathcal{C}$ *har en polynomial generatormatrix* G.

Bevis
Lad $G(D)$ være en vilkårlig forsinkelsesfri (og realiserbar) generatormatrix for en foldningskode og lad $b(D) \in \mathbb{F}_q(D)$ være det mindste fælles multiplum af alle nævnerne i elementerne i $G(D)$. Da $b(D)$ ikke har D som faktor, fordi $G(D)$ er realiserbar, vil matricen

$$G'(D) = b(D)G(D)$$

være en polynomial forsinkelsesfri matrix. Rent faktisk er den en generatormatrix for koden, for hvis

$$y(D) = x(D)G(D),$$

hvor $y(D)$ er en koderamme, og $x(D)$ en dataramme, så vil

$$y(D) = x'(D)G'(D),$$

hvor $x'(D)$ er en dataramme, da $x(D)b(D)^{-1}$ blot svarer til multiplikation med en skalar. □

En foldningskode $\mathcal{C}$ kan have mange forskellige generatormatricer. To generatormaticer $G(D)$ og $G'(D)$ genererer den samme kode, hvis og kun hvis

$$G'(D) = T(D)G(D), \tag{9.1}$$

hvor $T(D)$ er en rational invertibel $k \times k$-matrix (Blahut; 2003, side 281). Dette resultat kan fortolkes på følgende vis: Lad de to generatormatricer G og G' være relateret til hinanden som i ligning (9.1). Kodefølgen $y(D) = x(D)G(D)$ kan genereres ved $G'(D)$ som

$$\begin{aligned} y(D) &= x(D)G(D) \\ &= x(D)T^{-1}(D)T(D)G(D) \\ &= x'(D)G'(D), \end{aligned}$$

hvor $x'(D) = x(D)T^{-1}(D)$. Alle kodefølger, som kan genereres ved den ene generatormatrix, kan også genereres ved den anden, men ikke nødvendigvis ved den samme datafølge. To generatormatricer, som er ens på nær ombytning af søjler og elementære rækkeoperationer, siges at være *ækvivalente* generatormatricer. Det skyldes, at de vil generere den samme eller ækvivalente koder, hvormed forstås at koderne er ens bortset fra rækkefølgen af komponenterne i en ramme.

En generatormatrix for en foldningskode på formen

$$G(D) = \left(I \;\; P(D)\right),$$

hvor I er en $k \times k$-enhedsmatrix og $P(D)$ er en $k \times (n-k)$-matrix af polynomier, kaldes en *systematisk* generatormatrix og betegnes G_{sys}. Da en generatormatrix for en foldningskode indeholder rationale udtryk på formen $p(D)/q(D)$, kan generatormatricen altid bringes på systematisk form.

Eksempel 9.11
Generatormaticen for en foldningskode givet ved

$$G(D) = \left(1 + D^2 \;\; 1 + D + D^2\right)$$

er ækvivalent med den systematiske rationale generatormatrix

$$G_{\text{sys}}(D) = \left(1 \;\; \frac{1 + D + D^2}{1 + D^2}\right) = \left(1 \;\; 1 + \frac{D}{1 + D^2}\right).$$

(Blahut; 2003, side 281) ◇

Definitionen af tvungen længde kan skærpes ved at udtrykke den med udgangspunkt i $n \times k$-generatormatricen $G(D)$. Bemærk, at den tvungne længde er en egenskab ved koden og ikke ved generatormatricen. Indførelsen af en mindre række begreber er påkrævet, inden vi kan give definitionen. For en $k \times n$-matrix $G(D)$ er der $\binom{n}{k}$ forskellige måder at danne en $k \times k$-undermatrix på ved at 'fjerne' $(n-k)$ søjler. Determinanterne af disse $k \times k$-undermatricer kaldes for *underdeterminanter*. Givet en generatormatrix $G(D)$ af polynomier er den *interne grad* af denne, $\text{grad}_{\text{int}} G(D)$, defineret som den højeste grad af $k \times k$-underdeterminanterne af $G(D)$. Den eksterne grad af generatormatricen $G(D)$ defineres som

$$\text{grad}_{\text{ext}} G(D) = \sum_{i=0}^{k-1} \max_j \left(\text{grad}(g_{ij}(D))\right).$$

Hvert led i summen, kaldet en *rækkegrad*, er den største grad af et hvilket som helst polynomium i den række. Hvert led i udregningen af en vilkårlig $k \times k$-determinant er produktet af k elementer fra $G(D)$, én fra hver række. Elementerne fra den i'te række har højst grad $\max_j (\text{grad}(g_{ij}(D)))$ således, at hver $k \times k$-determinant ikke har grad større end $\text{grad}_{\text{ext}} G(D)$, altså haves følgende resultat:

Sætning 9.12
Om den interne og den eksterne grad af en generatormatrix $G(D)$ gælder uligheden

$$\text{grad}_{\text{int}} G(D) \leq \text{grad}_{\text{ext}} G(D).$$

Definition 9.13
En basal[5] *generatormatrix er en polynomial generatormatrix for en foldningskode, for hvilken ingen polynomial generatormatrix af mindre intern grad kan opnås ved multiplikation af matricen med en rational invertibel matrix $T(D)$.*

Nu kan vi give definitionen af tvungen længde ν i termer af generatormatricen, omend ν er uafhængig af valget af denne.

Definition 9.14 (Tvungen længde genbesøgt)
Den tvungne længde ν *af en foldningskode $\mathcal{C}$ er defineret som den interne grad af en vilkårlig basal generatormatrix for $\mathcal{C}$.*

Hukommelsen for en foldningskode er givet ved

$$m = \max_i \max_j \left(\text{grad}(g_{ij}(D))\right),$$

hvoraf kan sluttes at $\nu \leq km$.

5 *Nærværende definition af en basal generatormatrix stammer fra Blahut (2003). Alternative definitioner, der tager udgangspunkt i en højre-invers matrix til generatormaticen, kan også findes i litteraturen. Se eksempelvis (Forney, Jr.; 1970b, side 730).*

9.4 Egenskaber for foldningskoder

Til studiet af foldningskoder som kan dannes ved hjælp af polynomiale generatormatricer indføres to nye begreber. Studiet af disse er nyttigt i forbindelse med en afklaring af foldningskoders strukturelle egenskaber.

Definition 9.15 (Højre-invers matrix)
En polynomial højre-invers matrix $G^{-1}(D)$ *for en polynomial generatormatrix* $G(D)$ *er en* $k \times n$*-matrix af polynomier således, at*

$$G(D)G^{-1}(D) = I,$$

hvor I *er* $k \times k$*-enhedsmatricen.*

Definition 9.16
En polynomial checkmatrix $H(D)$ *for en polynomial generatormatrix* $G(D)$ *er en* $(n-k) \times n$*-matrix af polynomier der opfylder, at*

$$G(D)H(D)^T = 0,$$

hvor 0 *angiver nul-matricen.*

Sætning 9.17
Hvis $G(D)$ *er en systematisk polynomial generatormatrix,* $G_{\text{sys}}(D)$*, kan checkmatricen skrives som*

$$H(D) = \left(-P(D)^T \;\; I\right),$$

hvor I *her er en* $(n-k)\times(n-k)$*-enhedsmatrix og* $P(D)$ *er en* $(n-k)\times k$*-matrix.*

Bevis
Dette kan verificeres ved

$$G(D)H(D)^T = (I \;\; P(D)) \begin{pmatrix} -P(D) \\ I \end{pmatrix} = 0.$$

□

En indkoder for en foldningskode må være invertibel således, at data kan genvindes efter, at fejlene er blevet korrigeret. Ydermere må selve koden have den egenskab, at ingen ukorrigerbare fejlmønstre af endelig vægt vil ødelægge en i princippet uendelig mængde data. Disse egenskaber ved en polynomial generatormatrix er tæt forbundet med eksistensen af en polynomial højre-invers matrix. Som vi skal se om lidt, fører denne diskussion til definitionerne af såkaldte af *katastrofale*[6] og *ikke-katastrofale* generatormatricer. Katastrofal-egenskaben er en egenskab relateret til indkodningen – ikke til selve koden. Det 'katastrofale' ved en indkoder, der er baseret på en katastrofal generatormatrix er, at fejlmønstre, som indeholder et endeligt antal kanalfejl, i princippet vil producere et uendeligt antal fejl i de afkodede følger. I det generelle tilfælde, hvor matricen $G(D)$ har k rækker, tager definitionen af en ikke-katastrofal generatormatrix udgangspunkt i underdeterminanter af $G(D)$. Hvis $k \times k$-undermatricerne af $G(D)$ indekseres med ℓ og betegnes $G_\ell(D)$, angives determinanten af den ℓ'te $k \times k$-undermatrix med $|G_\ell(D)|$. Definitionen lyder da som følger:

6 Fra engelsk: Catastrophic og non-catastrophic.

Definition 9.18
En polynomial generatormatrix $G(D)$, hvis underdeterminanter $|G_\ell(D)|$ opfylder, at

$$\gcd\left(|G_\ell(D)|; \ell = 1, \ldots, \binom{n}{k}\right) = 1,$$

kaldes en ikke-katastrofal *generatormatrix. I andet tilfælde kaldes den en* katastrofal *generatormatrix.*

En systematisk generatormatrix for en foldningskode vil have en $k \times k$-undermatrix, der er lig indentitetsmatricen, og denne undermatrix har determinant lig 1. Heraf følger, at enhver systematisk generatormatrix er ikke-katastrofal. Der gælder for ikke-katastrofale generatormatricer, at de kan inverteres, da en højre-invers $k \times k$-undermatrix $G^{-1}(D)$ eksisterer (Blahut; 2003, side 286, 308-309).

Eksempel 9.19
Lad generatormatricen $G(D)$ være givet ved

$$G(D) = \begin{pmatrix} 1 & D^2 & 1 \\ 0 & 1 & D + D^2 \end{pmatrix}.$$

$G(D)$ har her tre 2×2-undermatricer. Vektoren bestående af determinanterne af disse tre undermatricer er

$$\Delta(D) = \begin{pmatrix} 1 & D + D^2 & 1 + D^3 + D^4 \end{pmatrix}$$

over $\mathbb{F}_2$. Da $\gcd \Delta(D) = 1$ eksisterer en højre-invers matrix $G^{-1}(D)$, som er givet ved

$$G^{-1}(D) = \begin{pmatrix} D^3 + D^4 & 1 + D^2 + D^3 + D^4 \\ D + D^2 & 1 + D + D^2 \\ 1 & 1 \end{pmatrix}.$$

(Blahut; 2003, side 286) ◇

En procedure til udregning af $G^{-1}(D)$ kan for eksempel findes i (Blahut; 2003, afsnit 9.10).

9.4.1 Afstandsmål for foldningskoder

Foldningskoder har mange minimumsafstande afhængigt af længden af det først indkodede kodesegment over hvilket afstanden måles. Afstandsmålet er defineret på en sådan vis, at det kun er egnet til afkodning af første ramme.

Definition 9.20
Den ℓ'te minimumsafstand *d_ℓ af en foldningskode er lig den mindste Hammingafstand mellem to vilkårlige først indkodede kodeordssegmenter, som er ℓ rammer lange, og som er forskellige i initialrammen.*

Følgen $d_1, d_2, d_3, \ldots$ kaldes afstandsprofilen for en foldningskode.

Da en foldningskode er lineær, kan vi lige så godt vælge et af de to kodeord til at være følgen bestående udelukkende af nuller. Altså haves at den ℓ'te minimumsafstand er lig den *ℓ'te minimumsvægt*. Kodeordet med den ℓ'te minimumsvægt er dét afkortede kodeord, der har mindst vægt, er ℓ rammer langt, og som er forskellig fra nulfølgen i første ramme.

Definition 9.21 (Fri Hammingafstand)
Den frie Hammingafstand *af en foldningskode $\mathcal{C}$ er givet ved*

$$d_{\text{fri}} = \max_{\ell} d_{\ell}.$$

Hvis en lineær blokkode betragtes som en speciel form for foldningskode, er den frie Hammingafstand for koden den samme som den mindste Hammingafstand normalt defineret. Den frie Hammingafstand er altså minimumsafstanden mellem to forskellige kodeord. Den frie Hammingafstand er den parameter, der afgør, hvad en kode kan præstere; des større fri Hammingafstand, des bedre er kodens fejl-korrigerende egenskaber. Afstandsprofilen for en foldningskode vokser indtil den er lig den frie Hammingafstand, det vil sige

$$d_1 \leq d_2 \leq d_3 \ldots \leq d_{\text{fri}}.$$

Der kan for foldningskodernes vedkommende formuleres et resultat i stil med det for blokkoderne:

Sætning 9.22
En foldningskode kan ved 'nærmeste-nabo afkodning' korrigere et hvilket som helst mønster af t fejl sålænge

$$t < \frac{d_{\text{fri}}}{2}.$$

Bevis
For tilstrækkelig stor værdi af ℓ følger resultatet af det lignende resultat for blokkoderne. □

Foldningskoder kan dog også korrigere mange andre mønstre, da mønstre med $2\nu > d_{\text{fri}}$ stadig kan have et entydigt nærmeste kodeord. Ifølge Blahut (2003) kan foldningskoder korrigere et uendeligt antal af sådanne mønstre, hvis de vel og mærke forefindes på tilstrækkeligt spredt vis i det modtagne ord.

9.5 Matrixbeskrivelse af foldningskoder

Som et alternativ til en polynomial beskrivelse af foldningskoder kan også gives en matrixbeskrivelse. En foldningskode er lineær og kan beskrives ved hjælp af en uendelig generatormatrix over legemet $\mathbb{F}_q$.

En generatormatrix for en foldningskode opnås på følgende vis: Generatorpolynomiernes *koefficienterne* $(g_\ell)_{ij}$, givet ved

$$g_{ij}(D) = \sum_{\ell} (g_\ell)_{ij} D^{\ell},$$

arrangeres i matricer indekseret efter ℓ. Lad for hvert ℓ, G_ℓ være givet ved $k \times n$-matricen

$$G_\ell = \left((g_\ell)_{ij} \right).$$

Generatormatricen for foldningskoden har da formen

$$G = \begin{pmatrix} G_0 & G_1 & G_2 & \dots & G_m & & & \\ & G_0 & G_1 & \dots & G_{m-1} & G_m & & \\ & & G_0 & \dots & G_{m-2} & G_{m-1} & G_m & \\ & & & \ddots & & & & \ddots \end{pmatrix}. \tag{9.2}$$

Bortset fra det diagonale bånd bestående af m undermatricer, som alle er forskellige fra 0, består resten af matricen udelukkende af nuller. Med udgangspunkt i dette kan vi opfatte foldningsindkoderen, som givet ved

$$y = xG,$$

hvor x er følgen af k-dimensionale dataramme, y er følgen af n-dimensionale koderamme, og G er givet som i ligning (9.2). (Lin & Costello, Jr.; 1983, side 292)

For en systematisk foldningskode kan generatormatricen skrives som

$$G = \begin{pmatrix} I & P_0 & 0 & P_1 & 0 & P_2 & \dots & 0 & P_m & & & \\ & I & P_0 & 0 & P_1 & 0 & \dots & P_{m-1} & 0 & P_m & & \\ & & I & P_0 & 0 & P_1 & \dots & 0 & P_{m-1} & 0 & P_m & \\ & & & \ddots & & & & & & & & \ddots \end{pmatrix}.$$

Matricen I er en $k \times k$-enhedsmatrix, 0 er en $k \times k$-matrix af nuller og $P_0, P_1, \dots, P_m$ er $k \times (n-k)$-matricer. Første række i matricen beskriver indkodningen af første dataramme til de første m koderammer og så videre.

En checkmatrix for G er en vilkårlig matrix H, som opfylder

$$G^{(\ell)} H^{(\ell)T} = 0,$$

hvor $\ell = 0, 1, 2, \dots$ og hvor $G^{(\ell)}$ og $H^{(\ell)}$ angiver øvre venstre undermatricer af henholdsvis G og H, svarende til ℓ rammer. En checkmatrix H er en uendeligdimensional matrix og kan konstrueres udfra G. Ved nærmere eftersyn ses, at

$$H = \begin{pmatrix} P_0^T & -I & & & & & & & & \\ P_1^T & 0 & P_0^T & -I & & & & & & \\ P_2^T & 0 & P_1^T & 0 & P_0^T & -I & & & & \\ \vdots & & \vdots & & \vdots & & & & & \\ P_m^T & 0 & P_{m-1}^T & 0 & P_{m-2}^T & 0 & \dots & P_0^T & -I & \\ & & & & \ddots & & & & & \ddots \end{pmatrix}.$$

Lad os se på et par eksempler.

Eksempel 9.23
En binær systematisk $(2, 1, 1)$-foldningskode er beskrevet ved 1×1-matricerne $P_0 = 1$ og $P_1 = 1$. Altså haves

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & & & & \\ & & 1 & 1 & 0 & 1 & & \\ & & & & 1 & 1 & 0 & 1 \\ & & & & & & & \ddots \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 1 & & & & & \\ 1 & 0 & 1 & 1 & & & \\ & & 1 & 0 & 1 & 1 & \\ & & & & & & \ddots \end{pmatrix}.$$

(Blahut; 2003, side 290) ◇

Eksempel 9.24
En binær systematisk $(3, 1, 3)$-foldningskode er beskrevet ved 2×1-matricerne $P_0 = (1\ \ 1)$, $P_1 = (0\ \ 1)$, $P_2 = (1\ \ 0)$ og $P_3 = (1\ \ 1)$. Altså haves

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ & & & & & & & & & & & & & & & & & \ddots \end{pmatrix}$$

og

$$H = \begin{pmatrix} 1 & 1 & 0 & & & & & & & & & & & & & \\ 1 & 0 & 1 & & & & & & & & & & & & & \\ 0 & 0 & 0 & 1 & 1 & 0 & & & & & & & & & & \\ 1 & 0 & 0 & 1 & 0 & 1 & & & & & & & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & & & & & & & \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & & & & & & & \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & & & & \\ & & & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \\ & & & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & \\ & & & & & & & & & & & & & & & \ddots \end{pmatrix}.$$

(Blahut; 2003, side 291) ◇

Betegnelsen *foldning*[7] i forbindelse med foldningskoderne skyldes altså, at der findes en følge $\mathscr{G}$ af matricer således, at det for alle $i \in \mathbb{Z}$ gælder

$$y_i = x * \mathscr{G} = \sum_{j+l=i} x_j \mathscr{G}_l,$$

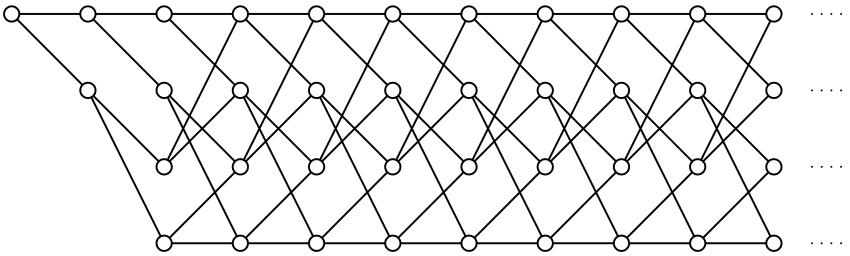
hvor j og l er heltal. Bemærk, at foldningens multiplikation er en multiplikation mellem vektor og matrix.

9.6 Trellis-beskrivelse af foldningskoder

Lad os her tilsidst se på en indførelse af foldningskoderne, som ikke tager sit matematiske udgangspunkt i algebraen, men, hvis den har et, snarere i grafteorien.

7 Foldning er også beskrevet i appendiks A i forbindelse med Fouriertransformationen på $\mathbb{Z}^N$.

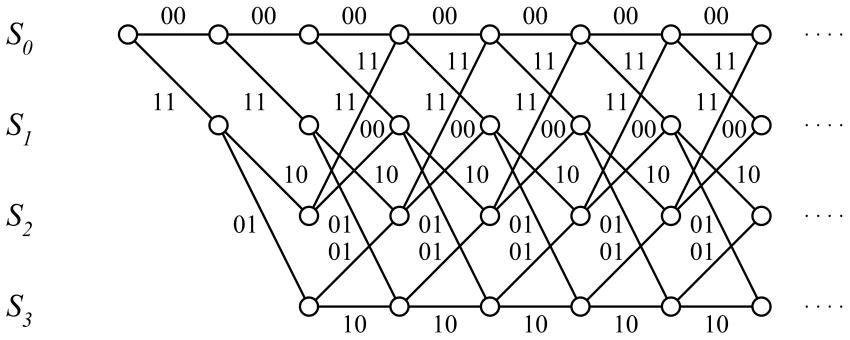
Som nævnt hedder trelliskoder, som de gør, fordi de ofte beskrives ved en form for semi-uendelig graf kaldet en *trellis*. En trellis er en *orienteret graf*, i hvilken hver knude er i en veldefineret dybde fra den begyndende knude. Det vil sige, at samtlige stier fra den begyndende knude til en sådan veldefineret knude indeholder det samme antal kanter. Vi vil i det følgende kun betragte trellis-diagrammer som er semi-uendelige mod højre, indeholdende q^ν knuder i hver søjle. Et eksempel på en trellis med 4 knuder i hver søjle og et binært kodealfabet er vist i figur 9.1. Kun knuder, der kan nås ved at starte ved den øverste venstre knude og derefter bevæge sig mod højre, er vist.



Figur 9.1 Trellis for en foldningskode. (Blahut; 2003, side 275)

Hver søjle af knuder i trellis-diagrammet indekseres med et heltal ℓ kaldet *tiden*. Til tid ℓ er hver af de forskellige tilstande repræsenteret ved forskellige knuder. En indkoder af tvungen længde ν over et q-ært symbolalfabet kan antage en hvilken som helst af q^ν tilstande, hvorfor der er q^ν knuder i en søjle, som viser mængden af tilstande til en given tid. Bemærk, at på hinanden følgende søjler viser mængden af tilstande til på hinanden følgende tider. Er der k datasymboler per ramme, må der være q^k forskellige kanter, der forlader hver knude; én kant for hver mulig dataramme. Bemærk, at når en trellis anvendes til at specificere koden, beskrives strukturen af foldningskoden uden at angive en indkoder. Lad os se et eksempel.

Eksempel 9.25
Et eksempel på en trellis for en binær foldningskode med $k = 1$, $n = 2$ og $\nu = 2$ er vist på figur 9.2. Foldningskoden er mængden af alle semi-uendelige binære ord, der kan læses ved at følge en hvilken som helst sti fra venstre mod højre gennem trellis-diagrammet og læse kodebittene langs stien. Indkodningsreglen er ikke givet. Trellis-diagrammet i figur 9.2 definerer en lineær kode, og dermed en foldningskode, da $\mathbb{F}_2$-summen af symbolerne i rækkerne langs et hvilket som helst par af stier er en række af symboler på en anden sti. Der er fire tilstande S_0, S_1, S_2 og S_3 i figuren svarende til de fire mulige 2-bit mønstre i indkoderens hukommelse – altså er $\nu = 2$. Hver knude i trellis-diagrammet har to grene gående ud fra sig; en gren for databit 0 og en gren for databit 1 – altså er $k = 1$. Hver gren fører til den nye tilstand forårsaget af databitten. Ved siden af hver gren står de to kodebit, som vil blive genereret, hvis vejen af denne gren følges – altså er $n = 2$. (Blahut; 2003, side 275) ◇



Figur 9.2 Trellis-diagram for en foldningskode med tvungen længde 2. Kodeordene dannes ved at følge en sti fra den begyndende knude i øverste venstre hjørne af diagrammet. (Blahut; 2003, side 275).

Hvilken gren, der følges på den videre vej fra de enkelte knuder, afhænger af den anvendte indkoder og dennes implementerede regelsæt (Blahut; 2003, side 275). Afstandsprofilen for en foldningskode kan aflæses af kodens trellis-diagram som i eksemplet her nedenfor.

Eksempel 9.26
Som tidligere beskrevet er den ℓ'te minimumsafstand givet ved forskellen mellem nulfølgen og det kodeord, der er tættest på denne, men forskellig fra nulfølgen i initialrammen (jævnfør definition 9.20). Eller nemmere endnu; det kodeord, der har mindst vægt og er forskellig fra 0-følgen i initialrammen, det vil sige kodeordet med ℓ'te minimumsvægt.

For foldningskoden beskrevet ved trellis-diagrammet i figur 9.2 kan vi for $\ell = 1, 2$ og 3 opskrive følgende:

$$\{00, \underline{11}\},$$
$$\{0000, 0011, \underline{1110}, \underline{1101}\},$$
$$\{000000, 000011, 001110, 001101, 111011, \underline{111000}, 110101, 110110\}.$$

Understregningerne angiver kodeordene med den ℓ'te minimumsvægt. Altså følger, at $d_1 = 2$, $d_2 = 3$ og $d_3 = 3$. Bemærk, at for $\ell = 2$ er kodeordet 0011 ikke kodeordet med den ℓ'te minumumsafstand, da dette kodeord er *lig* nul-kodeordet i den initiale ramme. Noget tilsvarende gør sig gældende for $\ell = 3$.

På lignende vis kan det aflæses af figur 9.25, at $d_4 = 4$ og godtgøres, at $d_i = 5$ for alle $i \geq 5$. Derfor haves at $d_{\text{fri}} = 5$. ◇

Trellis-tilgangen til foldningskoderne er, som vi skal se i kapitel 6 af HOVEDRAPPORTEN, særdeles anvendelig for afkodningen af disse.

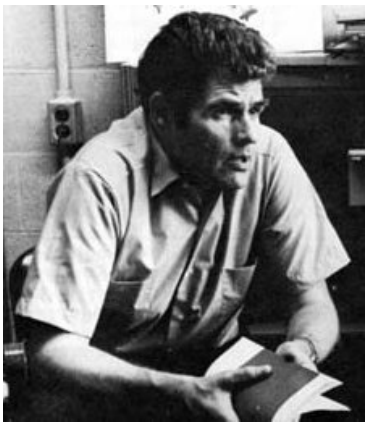
9.7 Afrunding

Foldningskoderne så første gang dagens lys i en artikel ved navn *Error-Free Coding* af Peter Elias fra 1954 (Elias; 1954).

I dette kapitel har vi indført foldningskoderne ved tre forskellige fremstillinger; en polynomial fremstilling, en matrixfremstilling og en trellisfremstilling.

Den første af disse fremstillingsformer er klart den mest matematisk orienterede. Vi har her indført foldningskoderne ved hjælp af deres generatorpolynomier og opstillet disse i en generatormatrix. Endvidere har vi opstillet en række kriterier, som foretrækkes at være opfyldt af generatormatricerne, idet de dermed bliver lettere at arbejde med. Trellisfremstillingen af foldningskoderne er, om man vil, en mere 'datalogisk' og implementeringsorienteret beskrivelse af koderne. Ikke desto mindre er den af relevans for afkodningen i MER-missionen, hvor der anvendes den såkaldte Viterbi-algoritme.

De i dag fortrinsvist anvendte foldningskoder er hovedsageligt fundet ved computersøgninger blandt andet af Odenwalder (Blahut; 2003, side 311-312).



Peter Elias
(1923-2001)

Elias blev født i Brunswick, New Jersey i 1923. Han var ved Swarthmore College, Pennsylvania, inden han i 1942 blev overflyttet til MIT, hvorfra han i 1944 modtog sin bachelorgrad i handel og teknisk kontrol. Fra 1944-46 tjente han i US Navy som radio-operatør. Herefter tog han kandidatgrad i engelsk og naturvidenskab samt en ph.d. fra Harvard University. I 1953 fik Elias en adjunkt-stilling ved MIT, i '56 blev han lektor og i '60 professor samme sted. Elias blev den yngste bestyrer af MIT Electrical Engineering and Computer Science (EECS), en stilling han holdt frem til 1966. Fra 1970-72 var Elias 'Cecil H. Green professor' i elektroteknik og i '74 blev han udpeget til 'Edwin S. Webster professor' i samme. Elias' forskningsområder var primært relaterede til informationsteori og dets anvendelser. I 1954 var han således den først til at introducere foldningskoderne. Elias var medlem af diverse komiteer og var editor på en række tidsskrifter. http://web.mit.edu/newsoffice/2001/elias.html

Litteratur

Adams, M. D. & Kossentini, F. (2000). Reversible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis, *IEEE Transactions on Image Processing* **9**: 1010–1024.

Berger, T. (1971). *Rate Distortion Theory – A Mathematical Basis for Data Compression*, Information and System Science Series, Prentice-Hall, Inc., Englewood, New Jersey.

Berlekamp, E. R. (1974). Key Papers in The Development of Coding Theory, IEEE Press, New York, pp. 1–6, 67–69, 107–108, 157–158, 233–237. Introductory Comments.

Biggs, N. L. (1989). *Discrete Mathematics*, Revised edn, Oxford Science Publications, Oxford.

Blahut, R. E. (2003). *Algebraic Codes for Data Transmission*, Cambridge University Press, Cambridge.

Booß-Bavnbek, B. (2005). Was bedeutet die Mathematisierung von Krieg?, *FIfF Kommunikation – Forum InformatikerInnen für Frieden und gesellschaftliche Verantwortung e.V.* **2**: 27–34.

Calderbank, A. R. (1998). The Art of Signaling: Fifty Years of Coding Theory, *IEEE Transactions on Information Theory* **44**(6): 2561–2595.

CCSDS (1997). Recommendation for Space Data System Standards – Lossless Data Compression.
URL: *http://public.ccsds.org/publications/archive/121x0b1.pdf*

CCSDS (2002). Recommendation for Space Data System Standards – Telemetry Channel Coding.
URL: *http://www.ccsds.org/documents/101x0b6.pdf*

Costello, Jr., D. J., Hagenauer, J., Imai, H. & Wicker, S. B. (1998). Applications of Error-Control Codes, *IEEE Transactions on Information Theory* **44**(6): 2531–2560.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*, Wiely Series in Telecommunications, John Wiley & Sons, Inc., New York.

Daubechies, I. (1992). *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series In Applied Mathematics, Capital City Press, Montpelier, Vermont.

Davis, P. J. (2004). A Letter to Christina of Denmark, *EMS* pp. 21–24.

Davis, P. J. (2005). Interview med professor emeritus Philip J. Davis den 6. marts 2005. Foretaget på Brown University, Providence.

Elias, P. (1954). Error-Free Coding, *IRE Transactions on Information Theory* **4**: 29–37.

Fernández-Cara, E. & Zuazua, E. (2003). Control Theory: History, Mathematical Achievements and Perspectives, *Boletin SEMA* **26**: 79–140.
URL: *http://me.wpi.edu/IEEETC_DPS/ARTICLES/DivSEMA.pdf*

Folkner, W. (2005a). Interview med doktor William Folkner den 16. marts 2005. Foretaget på JPL, Pasadena.

Folkner, W. (2005b). Interview med doktor William Folkner den 17. marts 2005. Foretaget på JPL, Pasadena.

Forchhammer, S. (2004). Interview med lektor Søren Forchhammer den 31. januar 2005. Foretaget på DTU, Lyngby.

Forney, Jr., G. D. (1966). Concatenated Codes, *MIT Press* .

Forney, Jr., G. D. (1970a). Coding and its application in space communication, *IEEE Spectrum* **7**: 47–58.

Forney, Jr., G. D. (1970b). Convolutional Codes I: Algebraic Structure, *IEEE Transactions on Information Theory* **IT-16**(6): 720–738.

Forney, Jr., G. D. (1974a). Convolutional Codes II: Maximum-Likelihood Decoding, *Information and Control* **25**: 222–266.

Forney, Jr., G. D. (1974b). Convolutional Codes III: Sequential Decoding, *Information and Control* **25**: 267–297.

Fraleigh, J. B. (1999). *A First Course in Abstract Algebra*, Sixth edn, Addison-Wesley, Reading, Massachusetts.

Frazier, M. W. (1999). *An Introduction to Wavelets through Linear Algebra*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, Berlin, Heidelberg.

Galar, R. (2005). *Adaptive vs Managerial Approach to S&T Policy*, Nowy Sacz.

Gallager, R. G. (1968). *Information Theory and Reliable Communication*, John Wiley & Sons, Inc., New York.

Gallager, R. G. (1978). Variations on a Theme by Huffman, *IEEE Transactions on Information Theory* **IT-24**(6): 668–674.

Gallager, R. G. & van Voorhis, D. C. (1978). Optimal Source Codes for Geometrically Distributed Integer Alphabets, *IEEE Transactions on Information Theory* **IT-21**(6): 228–230.

Golay, M. J. E. (1949). Notes on Digital Coding, *Proceedings of the IRE* **37**: 657.

Golomb, S. W. (1966). Run-Length Encodings, *IEEE Transactions on Information Theory* **IT-12**: 399–401.

Goodrich, M. T. & Tamassia, R. (2002). *Algorithm Design – Foundations, Analysis and Internet Examples*, John Wiley & Sons, Inc., New York.

Hamkins, J. (2005). Interview med doktor Jon Hamkins den 14. marts 2005. Foretaget på JPL, Pasadena.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes, *Bell System Technical Journal* **29**: 147–160.

Høholdt, T. (2004). Interview med docent Tom Høholdt den 26. august 2004. Foretaget på DTU, Lyngby.

Howard, P. G. (1998). Interleaving Entropy Codes, *Proceedings Compression and Complexity of Sequences 1997* pp. 45–55.

Howard, P. G. & Vitter, J. S. (1993). Fast and Efficient Lossless Image Compression, *Proceedings of the IEEE Computer Society/NASA/CESDIS Data Compression Conference* pp. 351–360.

Huffman, D. A. (1952). A Method for the Construction of Minimum-Redundancy Codes, *Proceedings of the IRE* **40**: 1098–1101.

Jacobs, I. M. (1974). Practical Applications of Coding, *IEEE Transactions on Information Theory* **IT-20**: 305–310.

Jensen, A. & la Cour-Harbo, A. (2001). *Ripples in Mathematics*, Springer-Verlag, New York, Berlin, Heidelberg.

JPL (1982). Reed-Solomon Encoders – Conventional vs Berlekamp Architecture.

JPL (2002). Mars Exploration Rover (MER) Project Mission Plan.

Justesen, J. & Høholdt, T. (2004a). *A Course In Error-Correcting Codes*, EMS Textbooks in Mathematics, European Mathematical Society, Zürich.

Justesen, J. & Høholdt, T. (2004b). Interview med professor Jørn Justesen og docent Tom Høholdt den 27. september 2004. Foretaget på DTU, Lyngby.

Kiely, A. (2005). Interview med doktor Aaron Kiely den 14. marts 2005. Foretaget på JPL, Pasadena.

Kiely, A. & Klimesh, M. (2001a). A New Entropy Coding Technique for Data Compression, *IPN Progress Report* (42-146): 1–48.
URL: *http://ipnpr.jpl.nasa.gov/progress_report/42-146/146G.pdf*

Kiely, A. & Klimesh, M. (2001b). Memory-Efficient Recursive Interleaved Entropy Coding, *IPN Progress Report* (42-146): 1–14.
URL: *http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-146/146J.pdf*

Kiely, A. & Klimesh, M. (2003). The ICER Progressive Wavelet Image Compressor, *IPN Progress Report* (42-155): 1–46.
URL: *http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-155/155J.pdf*

Kiely, A. & Klimesh, M. (2004). Preliminary Image Compression Results from the Mars Exploration Rovers, *IPN Progress Report* (42-156): 1–8.
URL: *http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-156/156I.pdf*

Kjeldsen, T. H., Pedersen, S. A. & Sonne-Hansen, L. M. (2004). *New Trends in the History and Philosophy of Mathematics*, Vol. 19 of *Studies in Philosophy*, University Press of Southern Denmark, Odense, pp. 11–27. Introduction.

Klimesh, M. (2005). Interview med doktor Matthew Klimesh den 14. marts 2005. Foretaget på JPL, Pasadena.

Kötter, R. (2005). Interview med doktor Ralph Kötter den 7. marts 2005. Foretaget på MIT, Cambridge.

Latour, B. (1987). *Science in Action – How to follow scientists and engineers through society*, Harvard University Press, Cambridge, Massachusetts.

Lin, S. & Costello, Jr., D. J. (1983). *Error Control Codes – Fundamentals and Applications*, Prentice-Hall, Inc., New Jersey.

MacKay, D. J. C. & Neal, R. M. (1996). Near Shannon Limit Performance of Low Density Parity Check Codes, *Electronics Letters* **32**: 1645–1646.
URL: *http://www.inference.phy.cam.ac.uk/mackay/mncEL.pdf*

Maimone, M. W. (2005). Interview med doktor Mark W. Maimone den 17. marts 2005. Foretaget på JPL, Pasadena.

Massey, J. L. (1975). Error Bounds for Tree Codes, Trellis Codes and Convolutional Codes with Encoding and Decoding Procedures, number 216 in *CISM Courses and Lectures*, Springer Verlag, Wien and New York, pp. 1–58.

Massey, J. L. (1992). Deep-Space Communication and Coding: A Marriage Made in Heaven, number 182 in *Lecture Notes in Control and Information Sciences*, Springer, Heidelberg and New York, pp. 1–17.

Matijevic, J. (2005). Interview med doktor Jacob Matijevic den 14. marts 2005. Foretaget på JPL, Pasadena.

McEliece, R. J. (1987). *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic Press, Boston, Dordrecht, Lancaster.

McEliece, R. J. (1998). *The Algebraic Theory of Convolutional Codes*, Elsevier Science B.V., Amsterdam, pp. 1065–1138. Kapitel 12 i Handbook of Coding Theory Volume I.

McEliece, R. J. (2002). *The Theory of Information and Coding*, Vol. 86 of *Encyclopedia of Mathematics and its Applications*, Second edn, Cambridge University Press, Cambridge, United Kingdom.

McEliece, R. J. (2004a). 2004 Shannon Lecture, *IEEE Information Theory Society Newsletter* **54**(4): 1–20.

McEliece, R. J. (2004b). Shannon Lecture XXVI – Are there Turbo Codes on Mars? Slides til foredrag.
URL: *http://www.ee.caltech.edu/EE/Faculty/rjm/papers/ShannonLecture.pdf*

McEliece, R. J. & Swanson, L. (1994). *Reed-Solomon Codes and the Exploration of the Solar System*, IEEE Press, New York, pp. 25–40. Kapitel 3 i Reed-Solomon Codes and Their Applications.

NASA (2003). Mars Exploration Rover Launches, *Press Kit* pp. 1–44.
URL: *http://www.jpl.nasa.gov/news/press_kits/merlaunch.pdf*

Niss, M. (1992). Noget om naturvidenskab, teknologi og samfund i den naturvidenskabelige revolutions periode (ca.: 1550-1750), number 228, IMFUFA, Roskilde, pp. 13–47.

Ono, F., Kino, S., Yoshida, M. & Kimura, T. (1989). Bi-Level Image Coding with MELCODE – Comparison of Block Type Code and Arithmetic Type Code, *Proceedings IEEE Global Telecommunications Conference* pp. 0255–0260.

Rabbani, M. & Jones, P. W. (1991). *Digital Image Compression Techniques*, Vol. 7 of *Tutorial Texts in Optical Engineering*, Spie Optical Engineering Press, Bellingham, Washington.

Reed, I. S. & Solomon, G. (1960). Polynomial Codes over Certain Finite Fields, *Journal of the Society for Industrial and Applied Mathematics* **8**: 300–304.

Reed, I. S. & Solomon, G. (1994). *Reed-Solomon Codes: A Historical Overview*, IEEE Press, New York, pp. 17–24. Kapitel 2 i Reed-Solomon Codes and Their Applications.

Roman, S. (1992). *Coding and Information Theory*, number 134 in *Gradute Texts in Mathematics*, Springer-Verlag, New York, Berlin, Heidelberg.

Roman, S. (1997). *Introduction to Information and Coding Theory*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, Berlin, Heidelberg.

Said, A. & Pearlman, W. A. (1993). Reversible Image Compression via Multiresolution Representation and Predictive Coding, *Proceedings of the SPIE Conference on Visual Communications and Image Processing 1993* **2094**: 664–674.

Said, A. & Pearlman, W. A. (1996). An Image Multiresolution Representation for Lossless and Lossy Compression, *Proceedings of the SPIE Conference on Visual Communications and Image Processing 1993* **9**(5): 1303–1310.

san Martin, M. & Folkner, W. (2005). Interview med doktor Miguel san Martin & doktor William Folkner den 14. marts 2005. Foretaget på JPL, Pasadena.

Sayood, K. (2000). *Introduction to Data Compression*, Second edn, Morgan Kaufmann Publishers, San Francisco, California.

Shannon, C. E. (1948). A Mathematical Theory of Communication I, II, Vol. 27, pp. 379–423; 623–656. Side 5-18 og 19-29 i Key Papers in The Development of Information Theory.

Shannon, C. E. (1949). Communication in the Presence of Noise, Vol. 37, pp. 10–21. Side 30-41 i Key Papers in The Development of Information Theory.

Shannon, C. E. (1959). Coding for a Discrete Source with a Fidelity Criterion, number 4, pp. 142–163. Side 245-266 i Key Papers in The Development of Information Theory.

Slepian, D. (1973). Key Papers in The Development of Information Theory, IEEE Press, New York, pp. 1–4. Introduction.

Solomon, D. (1998). *Data Compression – The Complete Reference*, Springer Verlag, New York.

Sontag, E. D. (1990). *Mathematical Control Theory – Deterministic Finite Dimensional Systems*, number 6 in *Texts in Applied Mathematics*, Springer Verlag, New York.

Strang, G. (2005). Interview med professor Gilbert Strang den 9. marts 2005. Foretaget på MIT, Cambridge.

Strang, G. & Nguyen, T. (1996). *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, Massachusetts.

Thompson, T. M. (1983). *From Error-Correcting Codes through Sphere Packings to Simple Groups*, number 21 in *The Carus Mathematical Monographs*, The Mathematical Association of America.

Vardy, A. (1998). *Trellis Structure of Codes*, Elsevier Science B.V., Amsterdam, pp. 1989–2117. Kapitel 24 i Handbook of Coding Theory Volume II.

Verdú, S. (1998). Fifty Years of Shannon Theory, *IEEE Transactions on Information Theory* **44**(6): 2057–2078.

Vitter, J. S. (1987). Design and Analysis of Dynamic Huffman Codes, *Journal of the Association for Computing Machinery* **34**(4): 825–845.

Walker, J. S. (1999). *A Primer on Wavelets and their Scientific Applications*, Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, Florida.

Weinberger, M. J., Seroussi, G. & Sapiro, G. (1996). LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm, *Proceedings of the IEEE Data Compression Conference* pp. 1–10.
URL: *http://www.hpl.hp.com/loco/dcc96copy.pdf*

Welch, G. & Bishop, G. (2004). An Introduction to the Kalman Filter, (TR-95-041): 1–16.
URL: *http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf*

Wicker, S. B. (1998). *Deep Space Applications*, Elsevier Science B.V., Amsterdam, pp. 2121–2169. Kapitel 25 i Handbook of Coding Theory Volume II.

Wicker, S. B. & Bhargava, V. K. (1994). *An Introduction to Reed-Solomon Codes*, IEEE Press, New York, pp. 1–16. Kapitel 1 i Reed-Solomon Codes and Their Applications.

A Fouriertransformationen på $\mathbb{Z}$

I dette appendiks vil vi udforske den diskrete Fouriertransformation, som udgør en del af grundlaget for at forstå forbehandlingen til den billedkomprimering, der er beskrevet i rapporten. Behandlingen af den diskrete Fouriertransformation vil i store træk bestå i at vise nogle af dens overordnede egenskaber. Fremstillingen er baseret på (Frazier; 1999, side 101-164).

Vi vil indledningsvis se på rummet $\mathbb{C}^N$ med indre produkt og tilhørende norm defineret ved

$$\begin{aligned} \langle z, w \rangle &= \sum_{j=0}^{N-1} z(j)\overline{w(j)}, \\ \|z\| &= \left(\sum_{j=0}^{N-1} |z(j)|^2 \right)^{1/2}. \end{aligned}$$

Når vi i det følgende taler om $\mathbb{C}^N$, vil vi underforstå det indre produktrum med ovenstående indre produkt.

Først udledes et par resultater vedrørende en særlig basis for rummet.

Sætning A.1
Lad $E_0, E_1, \ldots, E_{N-1} \in \mathbb{C}^N$ og lad E_m være givet ved

$$E_m(n) = \frac{1}{\sqrt{N}} e^{2\pi i m n/N}.$$

$\{E_m\}_{0 \leq m \leq N-1}$ udgør da en ortonormalbasis for $\mathbb{C}^N$.

Bevis

$$\langle E_j, E_k \rangle = \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i (j-k) n/N}.$$

For $k = j$ fås da

$$\langle E_j, E_k \rangle = \frac{1}{N} \sum_{n=0}^{N-1} 1 = 1.$$

For $k \neq j$ fås

$$\begin{aligned} \langle E_j, E_k \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i (j-k) n/N} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \left(e^{2\pi i (j-k)/N} \right)^n \end{aligned}$$

Dette er en potensrække med sum

$$\begin{aligned} & \frac{1}{N}\sum_{n=0}^{N-1}\frac{1-(e^{2\pi i(j-k)/N)^N}}{1-e^{2\pi i(j-k)/N}} \\ = \quad & \frac{1}{N}\sum_{n=0}^{N-1}\frac{1-e^{2\pi i(j-k)}}{1-e^{2\pi i(j-k)/N}} = 0. \end{aligned}$$

Ovenstående er en omskrivning, som vi skal benytte adskillige gange i begge appendikser. Da endvidere $\dim(\text{span}\{E_m\}_{m=0}^{N-1}) = N$ udgør $\{E_m\}$ en ortonormalbasis for $\mathbb{C}^N$. □

For $\mathbb{C}^N$ gælder altså, at

$$z = \sum_{i=0}^{N-1}\langle z, E_i\rangle E_i$$

og

$$\|z\|^2 = \sum_{i=0}^{N-1}|\langle z, E_i\rangle|^2.$$

Definition A.2
Lad $z \in \mathbb{C}^N$. For $m = 0, 1, \ldots, N-1$ defineres

$$\hat{z}(m) = \sum_{n=0}^{N-1} z(n)e^{-2\pi imn/N} = \sqrt{N}\langle z, E_m\rangle.$$

Så er

$$\hat{z} = \sqrt{N}\big(\langle z, E_0\rangle, \langle z, E_1\rangle, \ldots, \langle z, E_{N-1}\rangle\big).$$

Afbildningen $\hat{}: \mathbb{C}^N \to \mathbb{C}^N$, der afbilder z i $\hat{z}$ kaldes den diskrete Fouriertransformation*, og $\hat{z}$ kaldes* den Fouriertransformerede til z.

Vi kan benytte udtrykket for $\hat{z}(m)$ i definition A.2 til periodisk at udvide $\hat{z}(m)$ til alle $m \in \mathbb{Z}$.

$$\begin{aligned} \hat{z}(m+N) &= \sum_{j=0}^{N-1} z(j)e^{2\pi ij(m+N)/N} \\ &= \sum_{j=0}^{N-1} z(j)e^{2\pi ijm/N}e^{2\pi ij} \\ &= \sum_{j=0}^{N-1} z(j)e^{2\pi ijm/N} = \langle z, E_m\rangle = \hat{z}(m). \end{aligned}$$

Resultatet af udvidelsen er, at $\hat{z}$ bliver en N-periodisk funktion defineret på hele $\mathbb{Z}$. Vi vil derfor, når vi skriver, at $z \in \mathbb{C}^N$, underforstå, at z er en periodisk funktion med periode N.

Sætning A.3
Lad $z, w \in \mathbb{C}^N$. Så gælder

$$\begin{aligned}
\text{i} : z(n) &= \frac{1}{N}\sum_{j=0}^{N-1} \hat{z}(j) e^{2\pi i j n/N} \\
\text{ii} : \langle z, w\rangle &= \frac{1}{N}\sum_{j=0}^{N-1} \hat{z}(j)\overline{\hat{w}(j)} = \frac{1}{N}\langle \hat{z}, \hat{w}\rangle \\
\text{iii} : \|z\|^2 &= \frac{1}{N}\sum_{j=0}^{N-1} |\hat{z}(j)|^2 = \frac{1}{N}\|\hat{z}\|^2
\end{aligned}$$

Ovenstående sammenhænge kaldes henholdsvis *inversionsformlen, Parsevals ligning* og *Plancherels ligning.*

Bevis i-iii

$$\begin{aligned}
z(n) &= \sum_{j=0}^{N-1} \langle z, E_j\rangle E_j(n) = \sum_{j=0}^{N-1} \frac{1}{\sqrt{N}}\hat{z}(j)\frac{1}{\sqrt{N}} e^{2\pi i j n/N} \\
&= \frac{1}{N}\sum_{j=0}^{N-1} \hat{z}(j) e^{2\pi i j n/N} \qquad \text{og} \\
\langle z, w\rangle &= \sum_{j=0}^{N-1} \langle z, E_j\rangle\overline{\langle w, E_j\rangle} = \sum_{j=0}^{N-1} \frac{1}{\sqrt{N}}\hat{z}(j)\frac{1}{\sqrt{N}}\overline{\hat{w}(j)} \\
&= \frac{1}{N}\langle \hat{z}, \hat{w}\rangle \qquad \text{og} \\
\|z\|^2 &= \langle z, z\rangle = \frac{1}{N}\langle \hat{z}, \hat{z}\rangle = \frac{1}{N}\|\hat{z}\|^2.
\end{aligned}$$

□

Det viser sig, at basen $\{E_m\}$ på grund af kvadratroden ikke er så praktisk. Vi vil derfor indføre en anden basis.

Definition A.4
Definér for hvert $m = 0, 1, \ldots N-1$ vektoren F_m ved

$$F_m = \frac{1}{\sqrt{N}} E_m = \frac{1}{N} e^{2\pi i m n/N}.$$

Mængden $\{F_m\}_{m=0}^{N-1}$ kaldes da Fourierbasen for $\mathbb{C}^N$.

Fra definitionen er det klart, at Fourierbasen rent faktisk er en basis for $\mathbb{C}^N$, og at den er en ortogonalbasis. Benyttes Fourierbasen kan inversionsformlen udtrykkes således

$$z(n) = \sum_{j=0}^{N-1} \hat{z}(j) F_j(n),$$

så vektoren z kan skrives som

$$z = \sum_{j=0}^{N-1} \hat{z}(j) F_j.$$

Heraf ses, at afbildningen $\hat{}$ blot er et basisskift fra standardbasen til Fourierbasen. Fourierbasen har en interessant egenskab, som vi vil gøre opmærksom på her. Vi vil nøjes med at betragte realdelen af F_m. Denne er givet ved

$$\mathrm{Re}(F_m)(n) = \frac{1}{N} \cos\big(2\pi \frac{nm}{N}\big).$$

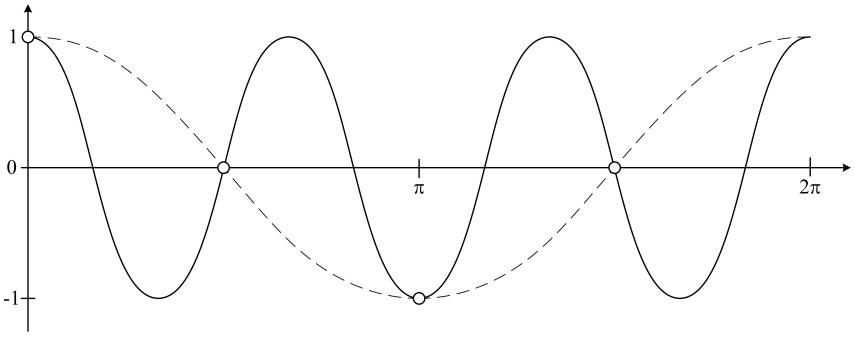
For $m = 0$ får vi bare den konstante funktion $1/N$. For $m = 1$ får vi funktionen

$$\frac{1}{N} \cos\big(2\pi \frac{n}{N}\big).$$

På intervallet $[0, N-1]$ foretager denne funktion én hel harmonisk svingning. På samme interval foretager $\mathrm{Re}(F_2)$ to hele svingninger og så videre. m synes altså umiddelbart at bestemme frekvensen for F_m. Her er dog en overraskelse. Lad $m > {}^N\!/_2$ og sæt $k = N - m$. Så er $k < {}^N\!/_2$ og

$$\begin{aligned}
\mathrm{Re}(F_m)(n) &= \frac{1}{N} \cos\big(2\pi \frac{nm}{N}\big) \\
&= \frac{1}{N} \cos\big(2\pi \frac{n(N-k)}{N}\big) \\
&= \frac{1}{N} \cos\big(2\pi n - 2\pi \frac{nk}{N}\big) \\
&= \frac{1}{N} \cos\big(2\pi \frac{nk}{N}\big) = F_k(n).
\end{aligned}$$

Ovenstående betyder, at sålænge $m \leq {}^N\!/_2$ giver større m større frekvens for $\mathrm{Re}(F_m)$. Når m bliver større end ${}^N\!/_2$, så falder frekvensen igen for $\mathrm{Re}(F_m)$. Denne egenskab skyldes, at vi kun ser på heltallige værdier for n. Svingningerne ligger med andre ord imellem de heltallige værdier. Dette er illustreret på figur A.1.



Figur A.1 $N = 4$. Funktionen $\cos(2\pi mn/4)$ er afbildet for $m = 1$ (den stiplede) og for $m = 3$ (den optrukne). $\circ$ angiver funktionerne evalueret i $n = 0, 1, 2, 3$. Bemærk, at man for disse n ikke kan skelne de to funktioner fra hinanden.

For en given vektor

$$z = \sum_{n=0}^{N-1} \hat{z}(n) F_n$$

kommer de højfrekvente bidrag altså fra n tæt på $N/2$, mens de lavfrekvente bidrag kommer fra n tæt på 0 eller N.

Vi vil her hurtigt indføre et begreb, som vi ikke vil behandle videre i appendiks, men som benyttes i selve rapporten til at definere Fouriertransformationen i en anden sammenhæng.

Definition A.5
Lad $\omega_N = e^{-2\pi i/N}$. Størrelsen ω_N kaldes da en N'te enhedsrod.

Navnet stammer fra den sidste af følgende egenskaber

$$\begin{aligned} \omega_N^j &= e^{-2\pi ij/N} \neq 1, \text{ for } j = 1, \ldots, N-1 \text{ mens} \\ \omega_N^N &= e^{-2\pi i} = 1. \end{aligned}$$

Vi kan benytte de N'te enhedsrødder til en anden formulering af den diskrete Fouriertransformation:

$$\hat{z}(m) = \sum_{n=0}^{N-1} z(n) e^{-2\pi imn/N} = \sum_{n=0}^{N-1} z(n) \omega_N^{mn}. \tag{A.1}$$

Sætning A.6
Fouriertransformationen $\hat{}$ er invertibel.

Bevis
Antag, at $\hat{w}, \hat{z} \in \mathbb{C}^N$, og at $\hat{w} = \hat{z}$. Så er $\hat{w}(n) = \hat{z}(n)$ for $n = 1, \ldots, N-1$ og

$$w = \frac{1}{N} \sum_{j=0}^{N-1} \hat{w}(j) F_j = \frac{1}{N} \sum_{j=0}^{N-1} \hat{z}(j) F_j = z.$$

Altså

$$\hat{w} = \hat{z} \quad \Rightarrow \quad w = z,$$

så afbildningen er injektiv. Surjektiviteten vises således: Lad $v \in \mathbb{C}^N$ og sæt

$$u(n) = \frac{1}{N} \sum_{j=0}^{N-1} v(j) e^{2\pi ijn/N}.$$

Så er

$$\begin{aligned} \hat{u}(m) &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} v(j) e^{2\pi ijn/N} e^{-2\pi imn/N} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} v(j) (e^{2\pi i(j-m)n/N})^n \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} v(j) \delta_{jm} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} v(m) \\ &= v(m). \end{aligned}$$

Ovenfor har vi igen benyttet resultatet for bestemmelse af en potensrækkes sum. Resultatet gælder for $m = 0, \ldots, N-1$ så $\hat{u} = v$. Surjektiviteten er således vist, og Fouriertransformationen er dermed bijektiv og invertibel. □

Definition A.7
Lad $z \in \mathbb{C}^N$ og sæt

$$\check{z} = \frac{1}{N} \sum_{n=0}^{N-1} z(n) e^{2\pi i m n/N} = \langle z, \overline{F_m} \rangle,$$

så kaldes afbildningen $\check{}$ den inverse Fouriertransformation.

For afbildningen gælder, at

$$(\hat{z})\check{}(m) = \frac{1}{N} \sum_{n=0}^{N-1} \hat{z}(n) e^{2\pi i m n/N} = z(m).$$

Vi kan benytte definition A.7 til at definere $\check{z}(m)$ for alle $m \in \mathbb{Z}$. Der gælder

$$\begin{aligned} \check{z}(m+N) &= \sum_{n=0}^{N-1} z(n) e^{-2\pi i n(m+N)/N} \\ &= \sum_{j=0}^{N-1} z(j) e^{2\pi i n m/N} = \check{z}(m). \end{aligned}$$

Resultatet af udvidelsen er altså ligesom for $\hat{z}$, at $\check{z}$ bliver en N-periodisk funktion defineret på hele $\mathbb{Z}$. Ved udregning ses, at følgende relation knytter den transformerede og den invers transformerede:

$$\check{z}(m) = \frac{1}{N} \hat{z}(-m) = \frac{1}{N} \hat{z}(N-m). \tag{A.2}$$

En stor del af de beviser, som gives i appendiks involverer ombytning af summationsrækkefølgen for forskellige summer. Formålet med udvidelserne af $\hat{z}$ og $\check{z}$ er i dette appendiks at gøre disse beviser mere elegante. For z's vedkommende har vi fra inversionformlen, at

$$\begin{aligned} z(m+N) &= \frac{1}{N} \sum_{n=0}^{N-1} \hat{z}(n) e^{2\pi i n(m+N)/N} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \hat{z}(n) e^{2\pi i n m/N} \\ &= z(m), \end{aligned}$$

så den N-periodiske udvidelse til $\mathbb{Z}$ giver altså også mening for z.

Definition A.8
Lad $z \in \mathbb{C}^N$ og lad $k \in \mathbb{Z}$. Definér

$$(R_k z)(m) = z(m-k).$$

Vektoren $R_k z$ kaldes translationen af z med k.

Eksempel A.9
Lad $z = \big(z(0), z(1), z(2), z(3)\big) = (1, 2, 3, 4)$. Så er

$$R_1 z = \big(z(-1), z(0), z(1), z(2)\big) = \big(z(3), z(0), z(1), z(2)\big) = (4, 1, 2, 3).$$

Af ovenstående ses, hvorfor R_k også kaldes en rotation. ◇

Sætning A.10
Lad $z \in \mathbb{C}^N$, og lad $k \in \mathbb{Z}$. Så er

$$(R_k z)\hat{}(m) = e^{-2\pi imk/N} \hat{z}(m).$$

Bevis

$$\begin{aligned}
(R_k z)\hat{}(m) &= \sum_{n=0}^{N-1} (R_k z)(n) e^{-2\pi imn/N} \\
&= \sum_{n=0}^{N-1} z(n-k) e^{-2\pi imn/N} \\
(j = n-k) &= \sum_{j=-k}^{-1} z(j) e^{-2\pi im(j+k)/N} + \sum_{j=0}^{N-1-k} z(j) e^{-2\pi im(j+k)/N} \\
(l = j+N) &= \sum_{l=N-k}^{N-1} z(l) e^{-2\pi im(l+k)/N} + \sum_{j=0}^{N-1-k} z(j) e^{-2\pi im(j+k)/N} \\
&= e^{-2\pi imk/N} \sum_{j=0}^{N-1} z(j) e^{-2\pi imj/N} \\
&= e^{-2\pi imk/N} \hat{z}(m).
\end{aligned}$$

□

Ovenstående kan også formuleres som

$$\hat{z}(m-k) = e^{-2\pi imk/N} \hat{z}(m). \tag{A.3}$$

Tilsvarende har vi

Sætning A.11
Lad $z \in \mathbb{C}^N$ og lad $k \in \mathbb{Z}$. Så er

$$(e^{-2\pi imk/N} z)\hat{}(m) = (\omega_N^{mk} z)\hat{}(m) = \hat{z}(m+k).$$

Bevis
Sæt $f(m) = \omega_N^{mk} z(m)$. Så er

$$\begin{aligned}
(\omega_N^{mk} z)\hat{}(m) &= \sum_{n=0}^{N-1} \omega_N^{nk} z(n) \omega_N^{mn} \\
&= \sum_{n=0}^{N-1} z(n) \omega_N^{n(m+k)} \\
&= \hat{z}(m+k).
\end{aligned}$$

□

Definition A.12
Lad $z, w \in \mathbb{C}^N$. Foldningen $z*w \in \mathbb{C}^N$ af z og w er vektoren, som opfylder

$$(z * w)(m) = \sum_{n=0}^{N-1} z(m-n)w(n), \quad \forall m \in \mathbb{Z}.$$

Eksempel A.13
Lad $z = (1, 2, 3, 4)$ og lad $w = (5, 6, 7, 8)$. Så er

$$\begin{aligned} z * w &= \big(z(0)w(0), z(3)w(1), z(2)w(2), z(1)w(3)\big) \\ &= (1 \cdot 5, 4 \cdot 6, 3 \cdot 7, 2 \cdot 8) = (5, 24, 21, 16). \end{aligned}$$

◇

Lemma A.14
Foldning er en kommutativ operation, det vil sige

$$(z * w)(m) = (w * z)(m).$$

Bevis

$$\begin{aligned} (z * w)(m) &= \sum_{n=0}^{N-1} z(m-n)w(n) \\ &= \sum_{j=m}^{j=m+N-1} z(j)w(m-j) \\ &= \sum_{j=0}^{j=N-1} w(m-j)z(j) \\ &= (w * z)(m), \end{aligned}$$

hvor vi har benyttet variabelskiftet $j = m - n$ i anden linie. □

Definition A.15
Lad $b \in \mathbb{C}^N$. Definér afbildningen $T_b : \mathbb{C}^N \to \mathbb{C}^N$ ved

$$T_b z = b * z.$$

T kaldes da en foldningsoperator.

Sætning A.16
Lad $z, w \in \mathbb{C}^N$. Så er

$$(z * w)\hat{}(m) = \hat{z}(m)\hat{w}(m), \quad \forall m \in \mathbb{Z}.$$

Bevis

$$\begin{aligned}
(z * w)\hat{}(m) &= \sum_{n=0}^{N-1} (z * w)(n) e^{-2\pi i m n/N} \\
&= \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} z(n-k) w(k) e^{-2\pi i m n/N} \\
&= \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} w(k) z(n-k) e^{-2\pi i m (n-k)/N} e^{-2\pi i m k/N} \\
&= \sum_{k=0}^{N-1} w(k) e^{-2\pi i m k/N} \sum_{n=0}^{N-1} z(n-k) e^{-2\pi i m (n-k)/N}.
\end{aligned}$$

Benytter vi, at z er N-periodisk, kan sidste sum i ovenstående skrives

$$\begin{aligned}
\sum_{n=0}^{N-1} z(n-k) e^{-2\pi i m (n-k)/N} &= \sum_{l=-k}^{N-1-k} z(l) e^{-2\pi i m l/N} \\
&= \sum_{l=0}^{N-1} z(l) e^{-2\pi i m (l)/N}.
\end{aligned}$$

Vi får samlet, at

$$\begin{aligned}
(z * w)\hat{}(m) &= \sum_{k=0}^{N-1} w(k) e^{-2\pi i m k/N} \sum_{l=0}^{N-1} z(l) e^{-2\pi i m l/N} \\
&= \hat{z}(m) \hat{w}(m).
\end{aligned}$$

□

Ved at gøre brug af Fouriertransformationen kan foldning altså foretages ved en simpel multiplikation. De resultater, som vi har vist for translation og foldning, vil vi gøre brug af i appendiks B.

B Wavelets

Dette appendiks fortsætter, hvor appendiks A slap. Teorien for wavelets bygger i høj grad på den diskrete Fouriertransformation, især hvis man ønsker at føre beviser. Det vil vi gøre her. I forhold til sidste appendiks er dette mere direkte koncentreret om beviser og indeholder ikke så meget forklarende tekst. Der gives beviser for sætning 3.7, 3.15, 3.19 og 3.21 samt for lemma 3.18. Alle er fra kapitel 3.

Appendikset bygger på (Frazier; 1999, side 165-263). Definitionerne samt enkelte af beviserne er taget herfra. De resterende beviser er foreslået som øvelser i samme bog.

Af hensyn til bevisernes overskuelighed har vi valgt at gengive de nødvendige definitioner i dette appendiks, selvom nogle af dem allerede står i kapitel 3.

B.1 Waveletbeviser

Definition B.1
Lad $z \in \mathbb{C}^N$. Den konjugerede refleksion *$\tilde{z}$ af z defineres som*

$$\tilde{z}(n) = \overline{z(N-n)} = \overline{z(-n)}$$

Lemma B.2
Den konjugerede refleksion opfylder

$$(\tilde{z})\hat{}(n) = \overline{\hat{z}(n)},$$

for alle n.

Bevis

$$\begin{aligned}
(\tilde{z})\hat{}(m) &= \sum_{n=0}^{N-1} \tilde{z}(n) e^{-2\pi imn/N} \\
&= \sum_{n=0}^{N-1} \overline{z(N-n)} e^{-2\pi imn/N} \\
&= \sum_{j=1}^{N} \overline{z(j)} e^{-2\pi im(N-j)/N} \\
&= \sum_{j=0}^{N-1} \overline{z(j)} e^{2\pi imj/N} \\
&= \overline{\hat{z}(m)}.
\end{aligned}$$

Fra anden til tredje linie i ovenstående er summationsrækkefølgen ombyttet, og der er foretaget variabelskiftet $j = N - n$. □

Lemma B.3
Lad $z, w \in \mathbb{C}^N$. Så er

$$(z * \tilde{w})(m) = \langle z, R_m w \rangle.$$

Bevis

$$\begin{aligned}
(z * \tilde{w})(m) &= (\tilde{w} * z)(m) \quad \text{ifølge (A.14)} \\
&= \sum_{n=0}^{N-1} \tilde{w}(m-n) z(n) \\
&= \sum_{n=0}^{N-1} \overline{w(n-m)} z(n) \\
&= \sum_{n=0}^{N-1} z(n) \overline{(R_m w)(n)} \\
&= \langle z, R_m w \rangle.
\end{aligned}$$

□

Lemma B.4
Foldning og konjugeret refleksion opfylder

$$(z * w)\tilde{}(n) = (\tilde{z} * \tilde{w})(n).$$

Bevis

$$\begin{aligned}
(z * w)\tilde{}(n) &= \left(\sum_{m=0}^{N-1} z(n-m) w(m) \right)\tilde{} \\
&= \sum_{m=0}^{N-1} \tilde{z}(n-m) \tilde{w}(m) \\
&= \sum_{m=0}^{N-1} \overline{z(m-n) w(-m)} \\
&= (\tilde{z} * \tilde{w})(n).
\end{aligned}$$

I ovenstående er benyttet, at den konjugerede refleksion af en linearkombination af vektorer er lig linearkombinationen af vektorernes konjugerede refleksioner. Dette følger af definition B.1. □

Definition B.5
Lad $z \in \mathbb{C}^N$. Definér z^* ved

$$z^*(n) = (-1)^n z(n)$$

for alle n.

Bemærk til en senere lejlighed, at

$$(z+z^*)(n) = z(n)(1+(-1)^n) = \begin{cases} 2z(n) & \text{for } n \text{ lige} \\ 0 & \text{for } n \text{ ulige} \end{cases} . \tag{B.1}$$

Lemma B.6
Lad $z \in C^N$. Så gælder

$$(z^*)\hat{}(m) = \hat{z}(n+M)$$

for alle n.

Bevis

$$\begin{aligned}
(z^*)\hat{}(m) &= \sum_{k=0}^{N-1} z^*(k) e^{-2\pi imk/N} \\
&= \sum_{k=0}^{N-1} (-1)^k z(k) e^{-2\pi imk/N} \\
&= \sum_{k=0}^{N-1} z(k) e^{-\pi ik} e^{-2\pi imk/N} \\
&= \sum_{k=0}^{N-1} z(k) e^{-2\pi ikM/N} e^{-2\pi imk/N} \\
&= \sum_{k=0}^{N-1} z(k) e^{-2\pi ik(m+M)/N} \\
&= \hat{z}(n+M).
\end{aligned}$$

□

Lemma B.7
Lad $N = 2M$, $M \in \mathbb{N}$ og lad $w \in \mathbb{C}^N$. Mængden $\{R_{2j}w\}_{j=0}^{M-1}$ med M elementer er da en ortonormal mængde, hvis og kun hvis

$$|\hat{w}(n)|^2 + |\hat{w}(n+M)|^2 = 2 \text{ for } n = 0, 1, \ldots, M-1.$$

Bevis

Vi starter beviset med at omskrive $(w, R_{2k}w)$.

$$\begin{aligned}
\langle w, R_{2k}w\rangle &= \frac{1}{N}\sum_{m=0}^{N-1}\hat{w}(m)\overline{\widehat{(R_{2k}w)}}(m) \quad \text{(Parsevals ligning (A.3))}\\
&= \frac{1}{N}\sum_{m=0}^{N-1}\hat{w}(m)\overline{\hat{w}(m)}e^{4\pi imk/N} \quad \text{(ifølge sætning A.10)}\\
&= \frac{1}{2M}\sum_{m=0}^{N-1}|\hat{w}(m)|^2e^{2\pi imk/M}\\
&= \frac{1}{2M}\left(\sum_{m=0}^{M-1}|\hat{w}(m)|^2e^{2\pi imk/M}+\sum_{m=M}^{N-1}|\hat{w}(m)|^2e^{2\pi imk/M}\right)\\
&= \frac{1}{2M}\sum_{m=0}^{M-1}|\hat{w}(m)|^2e^{2\pi imk/M}\\
&\quad+\frac{1}{2M}\sum_{j=0}^{M-1}|\hat{w}(j+M)|^2e^{2\pi i(j+M)k/M}\\
&= \frac{1}{2M}\sum_{m=0}^{M-1}|\hat{w}(m)|^2e^{2\pi imk/M}\\
&\quad+\frac{1}{2M}\sum_{j=0}^{M-1}|\hat{w}(j+M)|^2e^{2\pi ijk/M}\\
&= \frac{1}{2M}\sum_{m=0}^{M-1}\left(|\hat{w}(m)|^2+|\hat{w}(m+M)|^2\right)e^{2\pi imk/M}.
\end{aligned}$$

Fra linie fire til fem i ovenstående er der foretaget variabelskiftet $j = m - M$ for pladsholderen i den sidste sum. Ovenstående kan også skrives:

$$\frac{1}{2M}\sum_{m=0}^{M-1}W(m)e^{2\pi imk/M}. \tag{B.2}$$

Der skal altså vises, at $W(m) = 2$ for $m = 0, \ldots, M-1$ er ensbetydende med at ovenstående udtryk er 1 for $k = 0$ og 0 ellers. Antag det første. For k=0 kan udtrykket reduceres til

$$\langle w, w\rangle = \frac{1}{2M}\sum_{m=0}^{M-1}2 = 1.$$

For $k \neq 0$ fås

$$\begin{aligned}
\langle w, R_{2k}w\rangle &= \frac{1}{2M}\sum_{m=0}^{M-1}2e^{2\pi imk/M}\\
&= \frac{1}{2M}\sum_{m=0}^{M-1}2(e^{2\pi ik/M})^m\\
&= \frac{1}{2M}\sum_{m=0}^{M-1}2\frac{1-e^{2\pi ik}}{1-e^{2\pi ik/M}}\\
&= 0,
\end{aligned}$$

da $k \neq 0$. I ovenstående er udnyttet, at summen er en potensrække. Antag nu omvendt, at udtrykket i B.2 er lig 1, og at k er 0. Så har vi:

$$\frac{1}{2M}\sum_{m=0}^{M-1} W(m) = 1.$$

Heraf følger, at

$$\sum_{m=0}^{M-1} W(m) = 2M. \tag{B.3}$$

Lad nu $k \neq 0$. Så er

$$0 = \frac{1}{2M}\sum_{m=0}^{M-1} W(m)e^{2\pi imk/M} = \check{W}(k)$$

Den invers-transformerede opfylder altså

$$\check{W} = (\check{W}(0), 0, \ldots, 0).$$

Ifølge ligning (A.2) gælder derfor også

$$\hat{W} = (\hat{W}(0), 0, \ldots, 0).$$

Så har vi

$$\begin{aligned} W(m) &= \frac{1}{M}\sum_{n=0}^{M-1} \hat{W}(n)e^{2\pi imn/M} \\ &= \frac{1}{M}\hat{W}(0) \qquad \text{for } m = 0, \ldots, M-1. \end{aligned}$$

Funktionen $W(m)$ er altså en konstant funktion og fra ligning (B.3) sluttes, at $W(m) = 2$ for alle m. Dette færdiggør beviset. □

Lemma B.8
Lad $u, v \in \mathbb{C}^N$, igen med $N = 2M$. Så gælder

$$\{R_{2k}v\}_{k=0}^{M-1} \perp u \quad \Leftrightarrow \quad \{R_{2k}v\}_{k=0}^{M-1} \perp \{R_{2j}u\}_{j=0}^{M-1}. \tag{B.4}$$

Bevis
Pilen $\Leftarrow$ opnås umiddelbart ved at sætte $j = 0$. Den anden vej indses også let:

$$\begin{aligned} \langle R_{2j}u, R_{2k}v\rangle &= \sum_{n=0}^{N-1} u(n-2j)\overline{v(n-2k)} \\ &= \sum_{n=0}^{N-1} u(n)\overline{v(n+2j-2k)} \\ &= \langle u, R_{2(k-j)}v\rangle = 0. \end{aligned}$$

I ovenstående er udnyttet, at u og v er periodiske med periode N. □

Definition B.9
Lad $N = 2M$, $M \in \mathbb{N}$ og lad $u, v \in \mathbb{C}^N$. Definér for hvert $n \in \mathbb{Z}$ matricen $A(n)$ kaldet systemmatricen *for u og v ved*

$$A(n) = \frac{1}{\sqrt{2}} \begin{pmatrix} \hat{u}(n) & \hat{v}(n) \\ \hat{u}(n+M) & \hat{v}(n+M) \end{pmatrix}.$$

Sætning B.10
Lad $u, v \in \mathbb{C}^N$.

$$B = \{R_{2k}v\}_{k=0}^{M-1} \cup \{R_{2k}u\}_{k=0}^{M-1}$$

udgør da en førstetrins waveletbasis for $\mathbb{C}^N$, hvis og kun hvis systemmatricen $A(n)$ for u, v er unitær for $n = 0, \ldots, M-1$ eller tilsvarende

$$|\hat{u}(n)|^2 + |\hat{u}(n+M)|^2 = 2, \tag{B.5}$$
$$|\hat{v}(n)|^2 + |\hat{v}(n+M)|^2 = 2 \tag{B.6}$$

og

$$\hat{u}(n)\overline{\hat{v}(n)} + \hat{u}(n+M)\overline{\hat{v}(n+M)} = 0 \tag{B.7}$$

for hvert $n = 0, \ldots, M-1$.

Bevis
Ligningerne (B.5) og (B.6) giver ifølge lemma B.7, at mængderne $\{R_{2k}u\}_{k=0}^{M-1}$ og $\{R_{2k}v\}_{k=0}^{M-1}$ hver især er ortonormale. Vi viser først, at $\{R_{2k}v\}_{k=0}^{M-1} \perp u$ er ensbetydende med ligning (B.7).

Ifølge lemma B.3 er $(u, R_{2k}v) = 0$ ensbetydende med, at

$$(u * \tilde{v})(2k) = 0,$$

som igen ifølge bemærkningen til lemma B.1 er ensbetydende med

$$u * \tilde{v} + (u * \tilde{v})^* = 0,$$

fordi værdierne for ovenstående ulige indicer altid er 0. Ved invers Fouriertransformation fås, at ovenstående er ensbetydende med at

$$(u * \tilde{v})\hat{} + ((u * \tilde{v})^*)\hat{} = 0,$$

altså at

$$(u * \tilde{v})\hat{}(n) + ((u * \tilde{v})^*)\hat{}(n) = 0,$$

for alle $n \in \mathbb{Z}$. Kombineres dette med lemma B.6 fås, at

$$(u * \tilde{v})\hat{}(n) + (u * \tilde{v})\hat{}(n+M) = 0, \tag{B.8}$$

for alle $n \in \mathbb{Z}$. Fra lemma A.16 fås

$$(\hat{u}(\tilde{v})\hat{})(n) + (\hat{u}(\tilde{v})\hat{})(n+M) = 0,$$

som tilsidst sammen med lemma B.2 giver

$$\hat{u}(n)\overline{\hat{v}(n)} + \hat{u}(n+M)\overline{\hat{v}(n+M)} = 0.$$

Der gælder altså, at ovenstående er ensbetydende med, at $\{R_{2k}v\}_{k=0}^{M-1} \perp u$. Det sidste betyder imidlertid ifølge lemma B.8, at mængderne $\{R_{2k}u\}_{k=0}^{M-1}$ og $\{R_{2k}v\}_{k=0}^{M-1}$ er indbyrdes ortogonale og dermed ortonormale. Da antallet af elementer i de to mængder svarer til dimensionen af $\mathbb{C}^N$ udgør de to mængder en ortonormalbasis for $\mathbb{C}^N$. □

Definition B.11
Definér $D : \mathbb{C}^N \to \mathbb{C}^M$ *og* $U : \mathbb{C}^M \to \mathbb{C}^N$ *ved*

$$\begin{aligned} D(z)(n) &= z(2n), \\ U(z)(n) &= \begin{cases} z(n/2) & \text{for } n \text{ lige}, \\ 0 & \text{for } n \text{ ulige}. \end{cases} \end{aligned}$$

for $n = 0, \ldots, M-1$. D *kaldes* downsampling-operatoren. U *kaldes* upsampling-operatoren.

Lemma B.12
Lad N *være lige. Så gælder*

$$\begin{aligned} (D(z))\tilde{} &= D(\tilde{z}), \\ (U(z))\tilde{} &= U(\tilde{z}). \end{aligned}$$

Bevis
Fordi N er lige gælder, at hvis n er lige, så er også $-n$ lige i forhold til den periodiske udvidelse. Dermed er

$$\begin{aligned} (D(z))\tilde{}(n) &= \tilde{z}(2n) = \overline{z(-2n)} = D(\overline{z(-n)}) = D(\tilde{z})(n). \\ (U(z))\tilde{}(n) &= \left(\begin{cases} z(n/2) & \text{for } n \text{ lige} \\ 0 & \text{for } n \text{ ulige} \end{cases} \right)\tilde{} \\ &= \begin{cases} \overline{z(-n/2)} & \text{for } n \text{ lige} \\ 0 & \text{for } n \text{ ulige} \end{cases} = U(\tilde{z})(n). \end{aligned}$$

□

Lemma B.13
Lad $N = 2M$, $M \in \mathbb{N}$, *lad* $z \in \mathbb{C}^N$, *og lad* $x, y \in \mathbb{C}^M$. *Så gælder*

$$\begin{aligned} D(z) * x &= D(z * U(x)) \text{ og} \\ U(x) * U(y) &= U(x * y). \end{aligned}$$

Bevis
Bemærk først, at $U(x)(2n) = x(n)$. Derfor gælder

$$\begin{aligned} (D(z) * x)(n) &= \sum_{m=0}^{M-1} D(z)(n-m)x(m) \\ &= \sum_{m=0}^{M-1} z(2n-2m)U(x)(2m) \\ &= \sum_{k=0,\ k \text{ lige}}^{N-1} z(2n-k)U(x)(k) \\ &= \sum_{k=0}^{N-1} z(2n-k)U(x)(k) \\ &= (z * U(x))(2n) = D(z * U(x))(n). \end{aligned}$$

I ovenstående har vi benyttet, at $U(x)(k) = 0$ for k ulige og dermed, at

$$\sum_{k=0,\ k \text{ ulige}}^{N-1} z(2n-k)U(x)(k) = 0.$$

Dette viser første del.

$$\begin{aligned}
(U(x) * U(y))(n) &= \sum_{m=0}^{N-1} (U(x))(n-m)(U(y))(m) \\
&= \sum_{m=0,\ m \text{ lige}}^{N-1} (U(x))(n-m)(U(y))(m) \\
&= \sum_{k=0}^{M-1} (U(x))(n-2k)(U(y))(2k) \\
&= \sum_{k=0}^{M-1} (U(x))(n-2k)y(k).
\end{aligned}$$

Lad n være ulige. Så er $(U(x))(n-2k) = 0$ for alle k, så

$$(U(x) * U(y))(n) = 0 = U(x * y)(n).$$

Lad n være lige, det vil sige $n = 2j$. Så gælder

$$\begin{aligned}
(U(x) * U(y))(n) &= \sum_{k=0}^{M-1} (U(x))(2j-2k)y(k) \\
&= \sum_{k=0}^{M-1} x(l-k)y(k) \\
&= (x * y)(l) = (x * y)(2n) = (U(x * y))(n),
\end{aligned}$$

hvilket færdiggør beviset. □

Korollar B.14
Lad N være delelig med 2^p. Lad $z \in \mathbb{C}^N$, og lad $x, y \in \mathbb{C}^{N/(2^p)}$. Så gælder

$$\begin{aligned}
D^\ell(z) * x &= D^\ell(z * U^\ell(x)) \text{ og} \\
U^\ell(x * y) &= U^\ell(x) * U^\ell(y).
\end{aligned}$$

Bevis
Beviset føres ved induktion. For $\ell = 1$ følger ovenstående direkte af lemma B.13. Antag derfor,kapitel 3 at ovenstående gælder for ℓ. Så får vi:

$$\begin{aligned}
(D^{\ell+1}(z) * x)(n) &= (D(D^\ell(z)) * x)(n) \\
&= D(D^\ell(z) * U(x)) \qquad \text{(fra lemma B.13)} \\
&= D(D^\ell(z * U^{\ell+1}(x)))(n) \quad \text{(fra antagelsen)} \\
&= (D^{\ell+1}(z * U^{\ell+1}(x)))(n).
\end{aligned}$$

Dette viser første del.

$$\begin{aligned} U^{\ell+1}(x*y) &= U(U^{\ell}(x*y)) \\ &= U(U^{\ell}(x)*U^{\ell}(y)) \\ &= U(U^{\ell}(x))*U(U^{\ell}(y)) \quad \text{(fra lemma B.13)} \\ &= U^{\ell+1}(x)*U^{\ell+1}(y), \end{aligned}$$

hvilket færdiggør beviset. □

Sætning B.15 (Perfekt rekonstruktion)
Lad $u, v, s, t \in \mathbb{C}^N$. Lad $A(n)$ for hvert $n = 0, \ldots, N-1$ være systemmatricen for u og v. Så er der perfekt rekonstruktion, det vil sige

$$z = \tilde{s} * U(D(z*\tilde{u})) + \tilde{t} * U(D(z*\tilde{v})) \quad \forall z \in \mathbb{C}^N,$$

hvis og kun hvis

$$A(n)\begin{pmatrix} \hat{s}(n) \\ \hat{t}(n) \end{pmatrix} = \begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix} \tag{B.9}$$

for alle $n = 0, \ldots, N-1$. Hvis $A(n)$ er unitær, er kravet blot $t = \tilde{v}$ og $s = \tilde{u}$.

Bevis
Vi benytter os af, at

$$U(Dz) = \frac{1}{2}(z + z^*)$$

og dermed

$$U(D(z*\tilde{v})) = \frac{1}{2}((z*\tilde{v}) + (z*\tilde{v})^*).$$

For ovenstående foretager vi samme omskrivninger og slutninger som i beviset for sætning B.10 og får

$$\begin{aligned} & (\tilde{s}*U(D(z*\tilde{u})) + \tilde{t}*U(D(z*\tilde{v})))\hat{}(n) \\ = \;& \overline{\hat{t}(n)}\frac{1}{2}\left(\hat{z}(n)\overline{\hat{v}(n)} + \hat{z}(n+M)\overline{\hat{v}(n+M)}\right) \\ & + \overline{\hat{s}(n)}\frac{1}{2}\left(\hat{z}(n)\overline{\hat{u}(n)} + \hat{z}(n+M)\overline{\hat{u}(n+M)}\right) \\ = \;& \frac{1}{2}\left(\overline{\hat{t}(n)\hat{v}(n)} + \overline{\hat{s}(n)\hat{u}(n)}\right)\hat{z}(n) \\ & + \frac{1}{2}\left(\hat{t}(n+M)\hat{v}(n+M) + \hat{s}(n+M)\hat{u}(n+M)\right)\hat{z}(n+M) \end{aligned} \tag{B.10}$$

Hvis ovenstående er lig $\hat{z}(n)$ for alle n, så sikrer bijektiviteten af den inverse Fouriertransformation perfekt rekonstruktion.

Antag, at

$$\hat{t}(n)\hat{v}(n) + \hat{s}(n)\hat{u}(n) = 2 \tag{B.11}$$

$$\hat{t}(n+M)\hat{v}(n+M) + \hat{s}(n+M)\hat{u}(n+M) = 0 \tag{B.12}$$

for alle $n \in \mathbb{Z}$. Ved indsættelse i ligning B.10 ses, at vi så har perfekt rekonstruktion.

Antag omvendt perfekt rekonstruktion. Lad $\hat{z}(n) = 1$ og $\hat{z}(n+M) = 0$. Da vi har perfekt rekonstruktion må gælde, at

$$\hat{t}(n)\hat{v}(n) + \hat{s}(n)\hat{u}(n) = 2$$

for alle $n \in \mathbb{Z}$. Lad i stedet $\hat{z}(n) = 0$ og $\hat{z}(n+M) = 1$. På grund af perfekt rekonstruktion må gælde, at

$$\hat{t}(n+M)\hat{v}(n+M) + \hat{s}(n+M)\hat{u}(n+M) = 0.$$

Vi har altså samlet, at perfekt rekonstruktion er ensbetydende med at ligningerne (B.11) og (B.12) er opfyldt. Opskrives ligningerne på matrixform fås

$$A(n)\begin{pmatrix} \hat{s}(n) \\ \hat{t}(n) \end{pmatrix} = \begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix},$$

hvilket viser første del af sætningen.

Da $A(n)$ er unitær, gælder at $A(n)$ er invertibel med $A^{-1} = A^*$, hvor A^* er den adjungerede. Ved multiplikation med A^* på begge sider af ligning B.9 fås

$$\begin{aligned} \begin{pmatrix} \hat{s}(n) \\ \hat{t}(n) \end{pmatrix} &= A^*(n)\begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} \overline{\hat{u}(n)} & \overline{\hat{u}(n+M)} \\ \overline{\hat{v}(n)} & \overline{\hat{v}(n+M)} \end{pmatrix}\begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \overline{\hat{u}(n)} \\ \overline{\hat{v}(n)} \end{pmatrix} = \begin{pmatrix} (\tilde{u})\hat{}(n) \\ (\tilde{v})\hat{}(n) \end{pmatrix}. \end{aligned}$$

I ovenstående har vi brugt ligning B.2. Invers Fouriertransformation færdiggør beviset for sætningen. □

Definition B.16
Lad N være delelig med 2^p, $p \in \mathbb{N}$. En *p'te-trins waveletfilter-følge er en følge af vektorer $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ sådan, at der for hvert $\ell = 1, \ldots, p$ gælder*

$$u_\ell, v_\ell \in \mathbb{C}^{N/(2^{\ell-1})},$$

og at systemmatricen $A_\ell(n)$ defineret ved

$$A_\ell(n) = \frac{1}{\sqrt{2}}\begin{pmatrix} \hat{u}_\ell(n) & \hat{v}_\ell(n) \\ \hat{u}_\ell(n + N/(2^{\ell-1})) & \hat{v}_\ell(n + N/(2^{\ell-1})) \end{pmatrix}$$

er unitær for alle $n = 0, \ldots, N/(2^{\ell-1})$. Definér

$$\begin{aligned} x_1 &= D(z * \hat{v}_1) \in \mathbb{C}^{N/2}, \\ y_1 &= D(z * \hat{u}_1) \in \mathbb{C}^{N/2}. \end{aligned}$$

Definér endvidere ved induktion vektorerne

$$\begin{aligned} x_\ell &= D(y_{\ell-1} * \hat{v}_\ell) \in \mathbb{C}^{N/2^\ell}, \\ y_\ell &= D(y_{\ell-1} * \hat{u}_\ell) \in \mathbb{C}^{N/2^\ell}. \end{aligned}$$

for $\ell = 2, \ldots, p$.

Uddata fra en p'te-trins waveletfilter-følge er mængden af vektorer

$$\{x_1, x_2, \ldots, x_p, y_p\}.$$

Lemma B.17
Det samlede antal komponenter i

$$\{x_1, x_2, \ldots, x_p, y_p\}$$

er lig N.

Bevis
Vektoren x_ℓ har gennemgået ℓ downsamplinger og indeholder derfor $N/2^\ell$ komponenter. Det samlede antal komponenter i x_ℓ'erne og y_p er derfor lig

$$\begin{aligned}
\left(\sum_{n=1}^{p} \frac{N}{2^n}\right) + \frac{N}{2^p} &= N\left(\frac{1}{2^p} + \sum_{n=1}^{p} \frac{1}{2^n}\right) \\
&= \frac{N}{2}\left(\frac{1}{2^{p-1}} + \sum_{n=0}^{p-1} \frac{1}{2^n}\right) \\
&= \frac{N}{2}\left(\frac{1}{2^{p-1}} + \frac{1-\left(\frac{1}{2}\right)^p}{1-\frac{1}{2}}\right) \\
&= \frac{N}{2}\left(\frac{1}{2^{p-1}} + 2\left(1-\left(\frac{1}{2}\right)^p\right)\right) \\
&= \frac{N}{2}\left(\frac{1}{2^{p-1}} + 2 - \frac{1}{2^{p-1}}\right) \\
&= N.
\end{aligned}$$

□

Definition B.18
Lad $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ *opfylde*

$$u_\ell, v_\ell \in \mathbb{C}^{N/(2^{\ell-1})},$$

for $\ell = 1, \ldots, p$. *Lad* $f_1 = v_1$, $g_1 = u_1$ *og definér ved induktion* $f_\ell, g_\ell \in \mathbb{C}^N$ *ved*

$$\begin{aligned}
f_\ell &= g_{\ell-1} * U^{\ell-1}(v_\ell), \\
g_\ell &= g_{\ell-1} * U^{\ell-1}(u_\ell).
\end{aligned}$$

Lemma B.19

$$\begin{aligned}
\tilde{f}_{\ell+1} &= \tilde{g}_\ell * U^\ell(\tilde{v}_{\ell+1}), \\
\tilde{g}_{\ell+1} &= \tilde{g}_\ell * U^\ell(\tilde{u}_{\ell+1}).
\end{aligned}$$

Bevis
Følger direkte af lemma B.4 samt af lemma B.12. □

Lemma B.20
Lad $z \in \mathbb{C}^N$, og lad $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ være som i definition B.18. Definér $x_1, \ldots x_p, y_1, \ldots, y_p$ samt $f_1, \ldots f_p, g_1, \ldots, g_p$ som ovenfor. Så gælder for $\ell = 1, \ldots, p$, at

$$\begin{aligned} x_\ell &= D^\ell(z * \tilde{f}_\ell), \\ y_\ell &= D^\ell(z * \tilde{g}_\ell). \end{aligned}$$

Bevis
Der gives et induktionsbevis. $\ell = 1$:

$$\begin{aligned} x_1 &= D(z * \tilde{v}_1) = D(z * \tilde{f}_1), \\ y_1 &= D(z * \tilde{u}_1) = D(z * \tilde{g}_1). \end{aligned}$$

I ovenstående er definition B.16 benyttet. Antag nu, at sætningen er opfyldt for ℓ. Så er

$$\begin{aligned} x_{\ell+1} &= D(y_\ell * \tilde{v}_{\ell+1}) \\ &= D(D^\ell(z * \tilde{g}_\ell) * \tilde{v}_{\ell+1}) \\ &= D(D^\ell(z * \tilde{g}_\ell * U^\ell(\tilde{v}_{\ell+1}))) \\ &= D^{\ell+1}(z * \tilde{g}_\ell * U^\ell(\tilde{v}_{\ell+1})) \\ &= D^{\ell+1}(z * \tilde{f}_{\ell+1}). \end{aligned}$$

I udregningen har vi benyttet lemma B.19. Ovenstående er helt tilsvarende for y_ℓ. Dermed er sætningen vist. □

Korollar B.21
x_ℓ og y_ℓ opfylder

$$\begin{aligned} x_\ell(k) &= \langle z, R_{2^\ell k} f_\ell \rangle, \\ y_p(k) &= \langle z, R_{2^p k} g_p \rangle. \end{aligned}$$

Bevis
Vi benytter lemma B.20 og lemma B.3 til at skrive

$$x_\ell(k) = D^{\ell+1}(z * \tilde{f}_\ell) = (z * \tilde{f}_\ell)(2^\ell k) = \langle z, R_{2^\ell k} f_\ell \rangle$$

samt

$$y_p(k) = D^p(z * \tilde{g}_p) = (z * \tilde{g}_p)(2^p k) = \langle z, R_{2^p k} g_p \rangle.$$

□

Sætning B.22
Lad $u_1, v_1, \ldots, u_p, v_p$ for hvert $\ell = 1, \ldots, p$ opfylde, at

$$u_\ell, v_\ell \in \mathbb{C}^{N/(2^{(\ell-1)})}.$$

Definér f_ℓ og g_ℓ som i definition B.18. Hvis inddata til den ℓ'te gren i syntesen er x_ℓ, og alle andre inddata er 0, så er uddata fra syntesen lig

$$f_\ell * U^\ell(x_\ell).$$

Hvis inddata til den sidste gren er y_p, og alle andre inddata er 0, så er uddata fra syntesen lig

$$g_p * U^p(y_p).$$

Bevis
Lad

$$A_\ell(z) = (U(U(\ldots(U(U(z) * v_\ell) * u_{\ell-1})\ldots) * u_2) * u_1), \quad \text{(B.13)}$$
$$B_\ell(z) = (U(U(\ldots(U(U(z) * u_\ell) * u_{\ell-1})\ldots) * u_2) * u_1). \quad \text{(B.14)}$$

Vi viser først ved induktion, at

$$A_\ell(z) = U^\ell(z) * f_\ell, \quad \text{(B.15)}$$
$$B_\ell(z) = U^\ell(z) * g_\ell. \quad \text{(B.16)}$$

For $\ell = 1$ har vi

$$A_1(z) = U(z) * v_1 = U(z) * f_1,$$
$$B_1(z) = U(z) * u_1 = U(z) * g_1.$$

Antag nu, at ligning (B.15) og (B.16) gælder for ℓ. Så har vi

$$\begin{aligned} A_{\ell+1}(z) &= B_\ell(U(z) * v_{\ell+1}) \\ &= U^\ell(U(z) * v_{\ell+1}) * g_\ell \quad \text{(ifølge antagelsen)} \\ &= U^{\ell+1}(z) * U^\ell(v_{\ell+1}) * g_\ell \quad \text{(fra lemma B.14)} \\ &= U^{\ell+1}(z) * g_\ell * U^\ell(v_{\ell+1}) \\ &= U^{\ell+1}(z) * f_{\ell+1}. \end{aligned}$$

En tilsvarende udregning kan gennemføres for $B_{\ell+1}(z)$. Dette viser ligningerne (B.15) og (B.16).

Vi vender nu tilbage til sætningen. Lad alle inddata i rekonstruktion være 0 undtagen den ℓ'te gren hvis inddata er x_ℓ. Fra x_ℓ kan $y_{\ell-1}$ bestemmes (se eventuelt figur 3.4 i kapitel 3), fordi alle andre inddata er 0. $y_{\ell-1}$ bliver da

$$(U(x_\ell)) * v_\ell.$$

Nu kan $y_{\ell-2}$ bestemmes (igen fordi alle andre inddata er 0) til

$$(U(U(x_\ell)) * v_\ell) * u_{\ell-1}.$$

Hvis vi fortsætter denne upsampling og foldning, får vi tilsidst uddata fra rekonstruktionen. Uddata er givet ved

$$(U(U(\ldots(U(U(z) * v_\ell) * u_{\ell-1})\ldots) * u_2) * u_1),$$

som ifølge ligning (B.13) er lig $A_\ell(x_\ell)$. Fra ligning (B.15) har vi så, at uddata fra analysefasen er givet ved

$$U^\ell(x_\ell) * f_\ell,$$

hvis x_ℓ er eneste inddata forskellig fra 0. Resten af beviset kan gennemføres helt tilsvarende for y_p. □

Definition B.23
Lad $f_1, \ldots, f_p, g_p \in \mathbb{C}^N$, og lad B være på formen

$$\{R_{2k} f_1\}_{k=0}^{N/2-1} \cup \{R_{4k} f_2\}_{k=0}^{N/4-1} \cup \ldots \cup \{R_{2^p k} f_p\}_{k=0}^{N/2^p-1} \cup \{R_{2^p k} g_p\}_{k=0}^{N/2^p-1}.$$

Hvis B udgør en ortonormalbasis for $\mathbb{C}^N$, kaldes B en p'te-trins wavelet-basis. *$f_1, \ldots, f_p, g_p \in \mathbb{C}^N$ siges at generere B.*

Lemma B.24
Lad $g_{\ell-1} \in \mathbb{C}^N$ og lad mængden

$$\{R_{2^{\ell-1}k} g_{\ell-1}\}_{k=0}^{(N/2^{\ell-1})-1}$$

være ortonormal og indeholde $N/2^{\ell-1}$ elementer. Lad $u_\ell, v_\ell \in \mathbb{C}^{N/2^\ell-1}$ og lad systemmatricen $A_\ell(n)$ være unitær for hvert $n = 0, \ldots, N/2^\ell - 1$. Definér

$$\begin{aligned} f_\ell &= g_{\ell-1} * U^{\ell-1}(v_\ell), \\ g_\ell &= g_{\ell-1} * U^{\ell-1}(u_\ell). \end{aligned}$$

Så udgør

$$\{R_{2^\ell k} f_\ell\}_{k=0}^{(N/2^\ell)-1} \cup \{R_{2^\ell k} g_\ell\}_{k=0}^{(N/2^\ell)-1}$$

en ortonormal mængde med $N/2^{\ell-1}$ elementer.

Bevis
Ifølge lemma B.3 gælder, at

$$(g_{\ell-1} * \tilde{g}_{\ell-1})(2^{\ell-1}k) = \langle g_{\ell-1}, R_{2^{\ell-1}k} g_{\ell-1} \rangle = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k = 1, \ldots, {}^{N}/_{2^{\ell-1}}-1 \end{cases}$$

Da $A_\ell(n)$ er unitær for alle n gælder, at

$$\{R_{2k} v_\ell\}_{k=0}^{(N/2^\ell)-1} \cup \{R_{2k} u_\ell\}_{k=0}^{(N/2^\ell)-1}$$

udgør en ortonormalbasis for $\mathbb{C}^{N/2^{\ell-1}}$. Altså gælder

$$\begin{aligned} (v_\ell * \tilde{v}_\ell)(2k) &= \langle v_\ell, R_{2k} v_\ell \rangle = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k = 1, \ldots, N/2^{\ell-1} - 1, \end{cases} \\ (u_\ell * \tilde{u}_\ell)(2k) &= \langle u_\ell, R_{2k} u_\ell \rangle = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k = 1, \ldots, N/2^{\ell-1} - 1, \end{cases} \\ v_\ell * \tilde{u}_\ell(2k) &= \langle v_\ell, R_{2k} u_\ell \rangle = 0 \quad \forall k. \end{aligned}$$

Om f_ℓ gælder:

$$\begin{aligned} \langle f_\ell, R_{2^\ell k} f_\ell \rangle &= f_\ell * \tilde{f}_\ell(2^\ell k) \\ &= (g_{\ell-1} * U^{\ell-1}(v_\ell) * \tilde{g}_{\ell-1} * U^{\ell-1}(\tilde{v}_\ell))(2^\ell k) \\ &= ((g_{\ell-1} * \tilde{g}_{\ell-1}) * U^{\ell-1}(v_\ell * \tilde{v}_\ell))(2^\ell k) \\ &= \sum_{m=0}^{N-1} (g_{\ell-1} * \tilde{g}_{\ell-1})(2^\ell k - m)(U^{\ell-1}(v_\ell * \tilde{v}_\ell))(m). \end{aligned}$$

I ovenstående har vi benyttet lemma B.4 og korollar B.14 samt at foldning er kommutativ. Den sidste faktor i summen antager på grund af upsamplingen kun værdier forskellige fra 0 for $m = 2^{\ell-1}j$. Vi foretager derfor variabelskiftet $j = m/2^{\ell-1}$ og får:

$$\begin{aligned}\langle f_\ell, R_{2^\ell k} f_\ell \rangle &= \sum_{j=0}^{N/(2^{\ell-1})-1} (g_{\ell-1} * \tilde{g}_{\ell-1})(2^\ell k - 2^{\ell-1}j)(U^{\ell-1}(v_\ell * \tilde{v}_\ell))(2^{\ell-1}j) \\ &= \sum_{j=0}^{N/(2^{\ell-1})-1} (g_{\ell-1} * \tilde{g}_{\ell-1})(2^\ell k - 2^{\ell-1}j)(v_\ell * \tilde{v}_\ell)(j). \qquad \text{(B.17)}\end{aligned}$$

For den første faktor i summen gælder, at

$$\begin{aligned}(g_{\ell-1} * \tilde{g}_{\ell-1})(2^\ell k - 2^{\ell-1}j) &= \langle g_{\ell-1}, R_{2^\ell k - 2^{\ell-1}j} g_{\ell-1} \rangle \\ &= \begin{cases} 1 & \text{for } j = 2k, \\ 0 & \text{for } j \neq 2k \end{cases}.\end{aligned}$$

Indsættes dette i ligning (B.17) fås

$$\begin{aligned}\langle f_\ell, R_{2^\ell k} f_\ell \rangle &= (v_\ell * \tilde{v}_\ell)(2k), \\ &= (v_\ell, R_{2k} v_\ell) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k = 1, \ldots, N/2^\ell - 1 \end{cases}.\end{aligned}$$

hvilket viser, at

$$\{R_{2^\ell k} f_\ell\}_{k=0}^{(N/2^\ell)-1}$$

er ortonormal. På helt tilsvarende vis fås, at

$$\{R_{2^\ell k} g_\ell\}_{k=0}^{(N/2^\ell)-1}$$

er ortonormal. Den indbyrdes ortonormalitet vises ved en lignende udledning, som giver

$$\langle f_\ell, R_{2^\ell k} g_\ell \rangle = \sum_{m=0}^{N-1} (g_{\ell-1} * \tilde{g}_{\ell-1})(2^\ell k - m)(U^{\ell-1}(v_\ell * \tilde{u}_\ell))(m).$$

Om den sidste faktor gælder for alle n, at

$$(v_\ell * \tilde{u}_\ell)(n) = \langle v_\ell, R_{2n} u_\ell \rangle = 0.$$

Hele summen giver altså 0. Sammen med lemma B.8 færdiggør dette beviset. □

Lemma B.25
Lad mængden

$$\{R_{2^{\ell-1}k} g_{\ell-1}\}_{k=0}^{(N/2^\ell)-1}$$

være ortonormal og indeholde $N/2^{\ell-1}$ elementer. Lad $u_\ell, v_\ell \in \mathbb{C}^{N/2^{\ell-1}}$, og lad systemmatricen $A_\ell(n)$ være unitær for hvert $n = 0, \ldots, N/2^\ell - 1$. Definér

$$\begin{aligned} f_\ell &= g_{\ell-1} * U^{\ell-1}(v_\ell), \\ g_\ell &= g_{\ell-1} * U^{\ell-1}(u_\ell). \end{aligned}$$

Definér endvidere underrummene

$$\begin{aligned} V_{-\ell+1} &= \text{span}\{R_{2^{\ell-1}} g_{\ell-1}\}_{k=0}^{N/(2^{\ell-1})-1}, \\ W_{-\ell} &= \text{span}\{R_{2^\ell} f_\ell\}_{k=0}^{N/2^\ell-1}, \\ V_{-\ell} &= \text{span}\{R_{2^\ell} g_\ell\}_{k=0}^{N/2^\ell-1}. \end{aligned}$$

Så gælder, at

$$V_{-\ell+1} = V_{-\ell} \oplus W_{-\ell}.$$

Bevis
Fra lemma B.24 og linearitet har vi, at

$$V_{-\ell} \perp W_{-\ell}.$$

Vi viser nu, at $V_{-\ell} \subset V_{-\ell+1}$, og at $W_{-\ell} \subset V_{-\ell+1}$.
For $k = 0, \ldots, N/2^\ell - 1$ gælder

$$\begin{aligned} R_{2^\ell k} g_\ell(n) &= g_\ell(n - 2^\ell k) \\ &= g_{\ell-1} * (U^{\ell-1}(u_\ell))(n - 2^\ell k) \\ &= \sum_{m=0}^{N-1} g_{\ell-1}(n - 2^\ell k - m)(U^{\ell-1}(u_\ell))(m). \end{aligned}$$

Udtrykket $(U^{\ell-1}(u_\ell))(m)$ antager på grund af upsamplingen kun værdier forskellige fra 0 for $m = 2^{\ell-1} j$. Vi foretager derfor variabelskiftet $j = m/2^{\ell-1}$ og får derved fjernet upsamplingen:

$$\begin{aligned} R_{2^\ell k} g_\ell(n) &= \sum_{j=0}^{N/(2^{\ell-1})-1} g_{\ell-1}(n - 2^\ell k - 2^{\ell-1} j) u_\ell(j) \\ &= \sum_{j=0}^{N/(2^{\ell-1})-1} u_\ell(j) R_{2^{\ell-1}(j+2k)} g_{\ell-1}(n). \end{aligned}$$

Da ovenstående gælder for alle n, har vi, at

$$R_{2^\ell k} g_\ell = \sum_{j=0}^{N/(2^{\ell-1})-1} u_\ell(j) R_{2^{\ell-1}(j+2k)} g_{\ell-1}.$$

En vilkårlig af de udspændende vektorer for $V_{-\ell}$ kan altså skrives som en linearkombination af vektorer fra $V_{-\ell+1}$. Det samme gælder derfor for et vilkårligt element i $V_{-\ell}$. Vi har derfor, at $V_{-\ell} \subset V_{-\ell+1}$. Udregningen og slutningen kan gennemføres helt analogt for $W_{-\ell}$. Desuden gælder. at

$$\dim(V_{-\ell}) + \dim(W_{-\ell}) = \dim(V_{-\ell+1}).$$

Dette færdiggør beviset. □

Sætning B.26
Lad N være delelig med 2^p og lad $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ udgøre en p'te-trins waveletfilter-følge. Definer $f_1, \ldots, f_p, g_p$ som i definition B.18. Så genererer $f_1, \ldots, f_p, g_p$ en p'te-trins waveletbasis for $\mathbb{C}^N$.

Bevis
Ifølge definition B.16 er den til u_ℓ, v_ℓ hørende systemmatrix $A_\ell(n)$ unitær for $\ell = 1, \ldots, p$ og $n = 0, \ldots, N/2^\ell - 1$. Fra sætning B.10 har vi, at

$$\{R_{2k} f_1\}_{k=0}^{N/2-1} \cup \{R_{2k} g_1\}_{k=0}^{N/2-1}$$

udgør en ortonormal mængde. Fra lemma B.24 har vi så, at

$$\{R_{4k} f_2\}_{k=0}^{N/4-1} \cup \{R_{4k} g_2\}_{k=0}^{N/4-1}$$

ligeledes udgør en ortonormal mængde. Ved induktion og lemma B.24 får vi, at

$$\{R_{2^\ell k} f_\ell\}_{k=0}^{N/2^\ell-1} \cup \{R_{2^\ell k} g_\ell\}_{k=0}^{N/2^\ell-1},$$

udgør en ortonormal mængde.

Vi mangler at vise, at

$$\{R_{2^\ell k} f_\ell\}_{k=0}^{N/2^\ell-1} \perp \{R_{2^j n} f_j\}_{n=0}^{N/2^j-1} \tag{B.18}$$

og at

$$\{R_{2^\ell k} f_\ell\}_{k=0}^{N/2^\ell-1} \perp \{R_{2^p n} g_p\}_{n=0}^{N/2^p-1}. \tag{B.19}$$

Vi begynder med ligning (B.18). Antag for nemheds skyld, at $\ell < j$. Så gælder, at

$$R_{2^j n} f_j \in W_{-j} \subset V_{-j+1} \subset \ldots \subset V_{-\ell+1}.$$

Desuden gælder, at

$$R_{2^\ell k} f_\ell \in W_{-\ell} \perp V_{-\ell+1}.$$

Så

$$R_{2^j n} f_j \perp R_{2^\ell k} f_\ell.$$

Dette viser B.18. Ligning (B.19) vises tilsvarende.

Vi har altså samlet, at alle underrummene er ortonormale, at de er underrum af $\mathbb{C}^N$, og at summen af deres dimensioner er N. Dette godtgør, at rummenes baser samlet udgør en ortonormalbasis for $\mathbb{C}^N$. □

Vi kan tænke på konstruktionen af rummene således: Vi starter med at dele $\mathbb{C}^N$ i to underrum (V_{-1}, W_{-1}) med dimension $N/2$. Det ene underrum (W_{-1}) beholder vi uden videre opdeling. Det andet underrum (V_{-1}) deler vi igen i to underrum (V_{-2}, W_{-2}). Vi beholder W_{-2} og fortsætter med at dele V_{-2}. Denne deling gentages p gange.

Definition B.27
Lad $u_1, v_1, u_2, v_2, \ldots, u_p, v_p$ udgøre en p'te-trins waveletfilter-følge. Definer $f_1, \ldots, f_p, g_p$ som i definition B.18 og definér

$$\begin{aligned} \psi_{-j,k} &= R_{2^j k} f_j, \\ \phi_{-j,k} &= R_{2^j k} g_j. \end{aligned}$$

Den føromtalte ortonormalbasis for $\mathbb{C}^N$ har da formen

$$\{\psi_{-1,k}\}_{k=0}^{N/2-1} \cup \{\psi_{-2,k}\}_{k=0}^{N/4-1} \cup \ldots \cup \{\psi_{-p,k}\}_{k=0}^{N/2^p-1} \cup \{\phi_{-p,k}\}_{k=0}^{N/2^p-1}.$$

Elementerne i denne basis kaldes wavelets *på $\mathbb{C}^N$.*

Med ovenstående notation kan vi skrive

$$\begin{aligned} V_{-\ell} &= \text{span}\{\phi_{-\ell,k}\}_{k=0}^{N/2^\ell-1}, \\ W_{-\ell} &= \text{span}\{\psi_{-\ell,k}\}_{k=0}^{N/2^\ell-1}. \end{aligned}$$

Definition B.28
Lad

$$\{\psi_{-1,k}\}_{k=0}^{N/2-1} \cup \{\psi_{-2,k}\}_{k=0}^{N/4-1} \cup \ldots \cup \{\psi_{-p,k}\}_{k=0}^{N/2^p-1} \cup \{\phi_{-p,k}\}_{k=0}^{N/2^p-1}$$

udgøre en waveletbasis for $\mathbb{C}^N$*. Definér for* $z \in \mathbb{C}^N$ *den* partielle rekonstruktion $P_{-\ell}(z)$ *af* z *på niveau* $-\ell$ *ved* z*'s projektion på* $V_{-\ell}$ *som defineret ovenfor. Dette kan også formuleres*

$$P_{-\ell}(z) = \sum_{k=0}^{N/2^\ell-1} \langle z, \phi_{-\ell,k}\rangle \phi_{-\ell,k}.$$

Tilsvarende defineres $Q_{-\ell}(z)$ som z's projektion på $W_{-\ell}$, det vil sige

$$Q_{-\ell}(z) = \sum_{k=0}^{N/2^\ell-1} \langle z, \psi_{-\ell,k}\rangle \psi_{-\ell,k}.$$

Ifølge det foregående gælder så, at

$$P_{-\ell+1} = P_{-\ell} + Q_{-\ell}$$

og

$$z = P_0(z) = P_{-1} + Q_{-1}.$$

B.2 Resultater for Haar-basen

Lad u_ℓ, v_ℓ være defineret ved

$$\begin{aligned} u_\ell &= \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right), \\ v_\ell &= \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right) \end{aligned}$$

for $\ell = 1, \ldots, p$. Definér f_ℓ, g_ℓ som i definition B.18. Vi viser ved induktion, at

$$g_\ell(n) = \begin{cases} 2^{-\ell/2} & n = 0, 1, \ldots, 2^\ell - 1 \\ 0 & n = 2^\ell, 2^\ell + 1, \ldots, N-1 \end{cases}, \tag{B.20}$$

$$f_\ell(n) = \begin{cases} 2^{-\ell/2} & n = 0, 1, \ldots, 2^{\ell-1} - 1 \\ -2^{-\ell/2} & n = 2^{\ell-1}, 2^{\ell-1} + 1, \ldots, 2^\ell - 1 \\ 0 & n = 2^\ell, 2^\ell + 1, \ldots, N-1 \end{cases}. \tag{B.21}$$

For $\ell = 1$ følger ovenstående direkte fra definitionen af f_1, g_1. Vi viser først ligning (B.20). Antag, at ligningen gælder for ℓ. Så gælder, at

$$\begin{aligned}
g_{\ell+1}(n) &= (g_\ell * U^\ell(u_{\ell+1}))(n) \\
&= \sum_{m=0}^{N-1} g_\ell(n-m)(U^\ell(u_{\ell+1}))(m) \\
&= g_\ell(n)(U^\ell(u_{\ell+1}))(0) + g_\ell(n-2^\ell)(U^\ell(u_{\ell+1}))(2^\ell) \\
&= g_\ell(n)u_{\ell+1}(0) + g_\ell(n-2^\ell)u_{\ell+1}(2^\ell) \\
&= \frac{1}{\sqrt{2}}(g_\ell(n) + g_\ell(n-2^\ell)) \\
&= \begin{cases} \frac{1}{\sqrt{2}}g_\ell(n) & n = 0,1,\ldots,2^\ell-1 \\ \frac{1}{\sqrt{2}}g_\ell(n-2^\ell) & n = 2^\ell, 2^\ell+1, \ldots, 2^{\ell+1}-1 \\ 0 & n = 2^{\ell+1}, 2^{\ell+1}+1, \ldots, N-1 \end{cases}, \\
&= \begin{cases} 2^{-(\ell+1)/2} & n = 0,1,\ldots,2^{\ell+1}-1 \\ 0 & n = 2^{\ell+1}, 2^{\ell+1}+1, \ldots, N-1 \end{cases}.
\end{aligned}$$

På helt tilsvarende vis får vi, at

$$\begin{aligned}
f_{\ell+1}(n) &= \frac{1}{\sqrt{2}}(g_\ell(n) - g_\ell(n-2^\ell)) \\
&= \begin{cases} \frac{1}{\sqrt{2}}g_\ell(n) & n = 0,1,\ldots,2^\ell-1 \\ -\frac{1}{\sqrt{2}}g_\ell(n-2^\ell) & n = 2^\ell, 2^\ell+1, \ldots, 2^{\ell+1}-1 \\ 0 & n = 2^{\ell+1}, 2^{\ell+1}+1, \ldots, N-1 \end{cases}, \\
&= \begin{cases} 2^{-(\ell+1)/2} & n = 0,1,\ldots,2^\ell-1 \\ -2^{-(\ell+1)/2} & n = 2^\ell, 2^\ell+1, \ldots, 2^{\ell+1}-1 \\ 0 & n = 2^{\ell+1}, 2^{\ell+1}+1, \ldots, N-1 \end{cases}.
\end{aligned}$$

Nu er ligningerne (B.20) og (B.21) vist, og $\psi_{-\ell,k}$, $\phi_{-\ell,k}$ er lig $R_{2^\ell k}f_\ell$ og $R_{2^\ell k}g_\ell$ henholdsvis, det vil sige

$$\begin{aligned}
\psi_{-\ell,k}(n) &= \begin{cases} 2^{-\ell/2}, & n = 2^\ell k, 2^\ell k+1, \ldots, 2^\ell k + 2^{\ell-1} - 1 \\ -2^{-\ell/2}, & n = 2^\ell k + 2^{\ell-1}, \ldots, 2^\ell k + 2^\ell - 1 \\ 0, & n = 0,1,\ldots,2^\ell k - 1, 2^\ell k + 2^\ell, \ldots, N-1 \end{cases}, \\
\phi_{-\ell,k}(n) &= \begin{cases} 2^{-\ell/2}, & n = 2^\ell k, 2^\ell k+1, \ldots, 2^\ell k + 2^\ell - 1 \\ 0, & n = 0,1,\ldots,2^\ell k - 1, 2^\ell k + 2^\ell, \ldots, N-1 \end{cases}.
\end{aligned}$$

Sætning B.29
For Haar-transformationen opfylder den partielle rekonstruktion for $2^\ell k \leq n \leq 2^\ell k + 2^\ell - 1$

$$P_{-\ell}(z)(n) = \frac{1}{2^\ell}(z(2^\ell k) + z(2^\ell k + 1) + \ldots + z(2^\ell k + 2^\ell - 1)).$$

Bevis
Lad $2^\ell k \leq n \leq 2^\ell k + 2^\ell - 1$. Så er

$$P_{-\ell}(z)(n) = \sum_{m=0}^{N/2^\ell - 1} \langle z, \phi_{-\ell,m}\rangle \phi_{-\ell,m}(n).$$

Da $2^\ell k \leq n \leq 2^\ell k + 2^\ell - 1$ er $\phi_{-\ell,k}$ eneste basisvektor, som antager en værdi forskellig fra 0. Det vil sige

$$\begin{aligned} P_{-\ell}(z)(n) &= (z, \phi_{-\ell,k})\phi_{-\ell,k}(n) \\ &= \left(\sum_{i=0}^{N-1} z(i)\overline{\phi_{-\ell,k}(i)} \right) 2^{-\ell/2} \\ &= \left(\sum_{i=0}^{N-1} z(i)\phi_{-\ell,k}(i) \right) \frac{1}{\left(\sqrt{2}\right)^\ell}. \end{aligned}$$

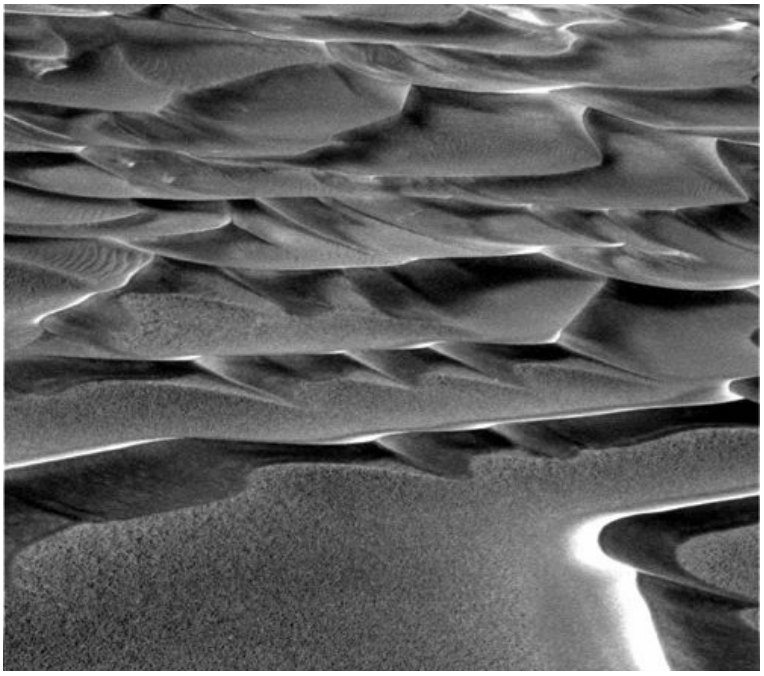
Basisvektoren $\phi_{-\ell,k}$ antager kun værdier forskellige fra nul for $2^\ell k \leq i \leq 2^\ell k + 2^\ell - 1$, så ovenstående bliver

$$\begin{aligned} P_{-\ell}(z)(n) &= (z(2^\ell k) + z(2^\ell k + 1) + \ldots + z(2^\ell k + 2^\ell - 1)) \frac{1}{\left(\sqrt{2}\right)^\ell} \frac{1}{\left(\sqrt{2}\right)^\ell} \\ &= \frac{1}{2^\ell}(z(2^\ell k) + z(2^\ell k + 1) + \ldots + z(2^\ell k + 2^\ell - 1)). \end{aligned}$$

□

Matematikken bag Mars-missionen

Transskriberede interviews fra
DTU, Brown University, MIT og JPL



Matematikspeciale af

Uffe Thomas Jankvist & Bjørn Toldbod

Oktober 2005 ∘ IMFUFA ∘ Roskilde Universitetscenter

I would imagine that at a place like JPL practically every branch of mathematics comes into play in some way. But I would also imagine that if you talk to a specific person the range of mathematics that that person is using is rather limited.

Philip J. Davis, Brown University, 2005

Forord

Denne del af specialet indeholder transskriberinger af de interviews, vi har foretaget i forbindelse med udarbejdelsen af specialet. †

De første interviews er foretaget ved Danmarks Tekniske Universitet, Lyngby i august og september 2004 samt januar 2005. Dernæst følger interviews foretaget under vores studierejse til USA i marts 2005: Brown University, Providence; Massachusetts Institute of Technology, Cambridge; Jet Propulsion Laboratory, Pasadena. Vi har valgt at bringe enkelte af de billeder, vi selv har taget under opholdene ved de tre amerikanske institutioner.

Uffe Thomas Jankvist & Bjørn Toldbod
Institut for Matematik og Fysik
Roskilde Universitetscenter
Februar 2006

† Billederne på rapportens omslag er taget af Opportunity. Forsidebilledet kan findes på http://marsrovers.jpl.nasa.gov/gallery/all/. På bagsiden ses hjulspor efter kørsel i svært fremkommeligt terræn. http://marsrovers.jpl.nasa.gov/gallery/press/opportunity/20040827a/merb_sol205_ncam_slippage-B210R1_br.jpg

Figur 0.1 Bjørn og Uffe på besøg hos Union Station Brewery i Providence, Rhode Island under opholdet ved Brown University i marts måned, 2005.

Indhold

1 Interviews fra Danmarks Tekniske Universitet **1**
1.1 Docent Tom Høholdt 26/8 2004 1
1.2 Professor Jørn Justesen & docent Tom Høholdt 27/9 2004 3
1.3 Lektor Søren Forchhammer 31/1 2005 6

2 Interview fra Brown University **13**
2.1 Professor emeritus Philip J. Davis 6/3 2005 13

3 Interviews fra Massachusetts Institute of Technology **25**
3.1 Doktor Ralph Kötter 7/3 2005 25
3.2 Professor Gilbert Strang 9/3 2005 25

4 Interviews fra Jet Propulsion Laboratory **33**
4.1 Doktor Aaron Kiely 14/3 2005 33
4.2 Doktor Matthew Klimesh 14/3 2005 44
4.3 Doktor Jon Hamkins 14/3 2005 47
4.4 Doktor Jacob Matijevic 14/3 2005 50
4.5 Doktor Miguel san Martin & doktor William Folkner 14/3 2005 54
4.6 Doktor William Folkner 16/3 2005 64
4.7 Doktor William Folkner 17/3 2005 70
4.8 Doktor Mark W. Maimone 17/3 2005 78

1 Interviews fra Danmarks Tekniske Universitet

Følgende interviews er foretaget ved Danmarks Tekniske Universitet (DTU) i Lyngby i henholdsvis august og september 2004 og januar 2005.

1.1 Docent Tom Høholdt 26/8 2004

Vi har håbet på, at man måske kan give specialet en historisk dimension, så det bliver en historie om anvendt matematik...

– Ja og der kan man sige, at hvis man tager fat i lige netop kodningsdelen af det, så er det overkommeligt fordi kodningsteori kun er halvtreds år gammel, og det sjove ved det er jo, at meget af det er udviklet langt inden teknologien var på plads. Det er jo udviklet som ren matematik og spekulation det her. Det er først i slutningen af tresserne, der kommer noget frem, som anvender kodningen, så på den måde er det jo skægt...

Ved du hvilke koder, der anvendes i dag?

– Lige præcis den sidste mission kan jeg ikke huske, men man bruger sådan et kombineret system... fordi der er ligesom to typer af fejl, der opstår; nogle hvor fejlene kommer lige efter hinanden i en stor blok, burst errors, og så er der enkeltfejl. Så man bruger et kombineret system, hvor man har to koder i sving af to forskellige typer. Den ene er en algebraisk kode, som hedder en Reed-Solomon kode, og den anden kode er en foldningskode, som opererer på en helt anden måde... Golay-koder bruger man ikke.

Er de nye koder perfekte?

– Nej, dem er der stort set ingen af... der er Hamming-koder og så er der Golay-koder. De er gode til nogen ting, men de har den svaghed, at hvis der kommer for mange fejl, så gør man med garanti noget forkert... så retter man til noget forkert. Det, der er det charmerende ved nogle af de andre er, at selvom man designer dem til at rette så så mange fejl, så i langt størstedelen af tilfældene, selvom der kommer stort set dobbelt så mange fejl, som koden garanterer at kunne rette, så kan de rette dem.

Hvad er det for en type koder. Er det lineære koder?

– Ja, ja og foldningskoderne er også lineære, men den præcise specifikation af kodningssystemet, det kan man finde på deres [JPLs] hjemmeside.

Har du været tilknyttet JPL?

– Nej, men Jørn Justesen har lavet noget for ESA, blandt andet den afkoder, der var oppe... jeg var mest med på den teoretiske side...

Hvornår stammer Reed-Solomon koderne fra?

– Helt tilbage fra halvtredserne. I en vis forstand er det de mest anvendte. Det er dem man bruger i en CD, og en CD-afspiller står jo i stort set hvert hjem.

Hvad bruger man til lagring på en harddisk?

– Der bruger man nogen gange Reed-Solomon koder og nogen gange bruger man faktisk Hammingkoder, som jo er meget simple.

Er der overhovedet sket noget nyt siden Shannons artikel i 1948?

– Det er jo et eksistensudsagn, som siger, at hvis man vil signalere med en bestemt hastighed og hvis den hastighed er under det der kaldes kanal-kapaciteten, så kan man gøre det med så lidt fejlsandsynlighed som muligt. Så er der sådan et sideresultat, at hvis man tager en tilfældig kode, så er den god. Og det er jo

meget sjovt, at hvis man tager en tilfældig kode, så er det faktisk svært at finde en, der er dårlig. Men når man rent faktisk skal lave dem, så er det ret svært... Jo der er sket to ting: Den ene er at i det der Mars-kodningssystem bruger de det der hedder en turbokode som ydre kode, og det er en bestemt type foldningskode fra slutningen af firserne. Den er opfundet af nogle franskmænd som ikke anede noget om kodningsteori sådan 'nu prøver vi det her' og det virkede. Hvorfor det virkede så godt er der ikke rigtigt nogen der forstår i dag. Man forstår lidt af det... men godt nok til at de kom ind i standarden. Det er den ene ting. Der foregår også stadig udvikling af de algebraiske koder, og så er man ved at vende tilbage til ting fra 60'erne opfundet af Gallager, som hedder LDPC-koder. Det viser sig, at de også virker fremragende, men det er så blokkoder i modsætning til foldningskoderne... og det forstår man heller ikke helt. Man regner med at næste generation af kommunikationssystemer kan bruge dem. Det er ikke kun satellitter, men også modemmer og mobiltelefoner med mere.

– [...] ... men sandheden er jo, at normalt hænger den egentlige anvendelse noget bagefter den teoretiske udvikling. Og i tidernes morgen langt bagefter. Men nu går det forholdsvis hurtigt, så tre-fire år efter at turbokoderne er opfundet, så kommer de ind i standarderne. Det kan man så nogen gange synes er lige hurtigt nok, fordi man ikke forstår dem godt nok, men det er sådan et teori-matematik-synspunkt. Men synspunktet er et andet en gang i mellem for hvis det virker, så skal man da bruge det.

Justesen-koder, hvad er det?

– Det er teori. Hvis man har forstået hvad en Reed-Solomon kode er, så er det let at forstå. Men det er teoretisk resultat. Dem er der ingen, der bruger til noget som helst. Men det er et meget stort teoretisk resultat. Man har nogle grænser, som kommer efter Shannon, for hvor gode koder man kan lave...

...som for eksempel Gilbert-Varshamov?

– Ja, men den er ikke konstruktiv... og det eneste man i virkeligheden ved om det, altså binære koder, det er Justesen-koder... men det er et asymptotik-resultat – når koderne bliver meget lange – det er et teori-resultat. Men koder med bloklængde op mod en million, det bruger man da.

Det er et gammelt resultat?

– Ja det er fra '72. Så er der folk, der har forbedret det en lille bitte smule for nogle hastigheder, men den oprindelige idé er helt den samme. Det der skete i 80'erne var, at for ikke-binære koder, for koder over alfabeter af størrelse 49, der kan man faktisk lave noget, der er bedre end Gilbert, og det startede med et teoriresultat igen. Det er det, der hedder algebraiske geometrikoder... så der sker stadigvæk noget, men det er mere eller mindre overtaget af matematikerne. De var meget optaget af det, ingeniørerne, i en overgang af de der algebraiske geometrikoder, til afløsning af Reed-Solomon koder som standard, men det er dødt... desværre... der blev lavet én chip.

Er der nogen grund til, i forhold til Mars-missionen, at interessere sig for andet end binære koder?

– Nej. Reed-Solomon koderne er født ikke-binære, men når man bruger dem så... ja man har et endeligt legeme med to i ottende elementer, så er hver byte repræsenteret ved otte bit...

Hvad med ikke-lineære koder?

– Nej. No way. Det er teori. Men en charmerende teori, for nogen gange kan man lave ikke-lineære koder med dobbelt så mange kodeord som en lineær kode. Men der er ikke nogen, der bruger dem til noget, det er alt for kompliceret.

I Hammings bog nævner han anvendelse af ternære-koder?

– Ja af ternære koder, men det er ikke til kommunikation, det er til at lave tips-systemer. Der er en ternær Golay-kode, der er perfekt, som hvis man har en tipskupon med elleve rækker kan garantere ni rigtige, hvis du tipper 729 rækker. Så du skal bare finde to sikre og ofre 729 kroner, så har du den... hvis man har en tipskupon med tolv kampe og spørger: 'Hvor få rækker skal du tippe for at garantere en elleve'r, se det er der ingen der ved.

Har du et lignende tip til Lotto?

– Nej nej. Der er ikke noget at gøre.

Det var en skam. Ellers kunne vi måske få finansieret vores rejse...

1.2 Professor Jørn Justesen & docent Tom Høholdt 27/9 2004

Med hensyn til kodningsteorien er vores mål blandt andet at finde ud af hvilke koder, der rent faktisk anvendes, hvordan de anvendes og hvorfor man anvender lige præcis de koder...

JJ Til min overraskelse var McEliece's påstand den, at det var den store foldnings-kode, der nu var gravet op igen... og Reed-Solomon-koder.

Er der nogen særlig grund til at man anvender lige præcis de koder, man gør? Man kunne godt forestille sig, at når man skulle kommunikere med en robot, der kører rundt på Mars, at der så var nogen andre forhold, der gjorde sig gældende der, end hvis bare skulle kommunikere på Jorden, og at man skulle tage højde for dét, når man valgte hvilken kode, man ville bruge.

JJ Jeg vil sige, at sådan bør det ikke være. Man skal ligesom bestemme sig for, hvilket miljø man er i, når man laver kommunikationssystemer, og det er som regel noget med hvor stor pålidelighed man vil have, og hvor stor forsinkelse man kan tillade sig. Så hvis man snakker om mobiltelefoner, for eksempel, så befinder man sig helt ude ved den yderlighed, hvor man er ligeglad med kvaliteten stort set, så længe man kan høre hvad der bliver sagt, men forsinkelser er meget generende – det er svært at tale sammen, hvis forsinkelsen er et sekund. Den modsatte yderlighed, er de ting, der vedrører styring af, hvad de forskellige mekanismer på Mars laver. Hvis man sender en besked, der bliver fejlfortolket, så går det måske helt galt, så der går man meget op i at de ting, der kommer igennem er korrekte, og der er jo i forvejen en stor forsinkelse, så det er ligesom ikke det, der er det mest kritiske.

Hvor stor er forsinkelsen cirka?

JJ Omkring otte minutter, så det at det også tager et sekund at regne ud, det gør hverken fra eller til. Så der kan man sige, at hvis man snakker om fjerne rumsonder, der er det endnu mere ekstremt, at man ikke sender noget begge veje, og at man stort set er ligeglad med hvor lang tid man skal bruge på at afkode. Så i den situation vil man gerne have noget, der er meget beskyttet, og ligeglad med, hvor meget man skal regne på det. Så er der også en afvejning, der er meget mere teknisk og lidt sværere at forklare, men jo bedre man vil udnytte sin kommunikationsflade, jo flere data man vil have igennem, jo hårdere skal man arbejde på afkodningen. Så der er en afvejning mellem hvor dyrt det er at skaffe sig energi, og hvor meget man vil presse kommunikationen. Hvis tingene begynder at gå for dårligt, så bliver man nødt til at gå ned i hastighed, men det afhænger jo af hvor meget energi, man er i stand til at skaffe sig, og der er meget lidt energi på Mars, der er næsten ingenting. Det vil også sige, at man kan tillade sig en meget heftig kode. På den anden side laver man ikke koderne om i tide og utide. Så grundlæggende bruges en standard, som blev grundlagt i begyndelsen af 70erne, og som ikke har ændret sig dramatisk siden. Det var et godt valg man lavede, og det er også meget svært at få lavet om. Det er meget konservativt.

Er det simpelthen fordi den teknologi man har, har de her koder indbygget i sig i forvejen, og man så er nødt til at lave ny teknologi... altså er det et pengespørgsmål?

JJ Ja, det er det, at der findes det udstyr på alle Jordens stationer, både NASA og ESA og visse andre, og de er enige om at bruge det her. Hvis man ændrer det, er det i kraft af, at man siger: 'Nu indfører vi et alternativ.' Så vil man i en lang periode have to muligheder for at kommunikere og se om det nye er bedre, og så vil man eventuelt om ti år skifte, men det går meget stille for sig. Og det er klart, at hver gang man afprøver noget, så er man nødt til at gå gennem store anstrengelser for at sikre, at udstyret er korrekt. Det at teste, at man koder og afkoder korrekt er et stort problem. En gang imellem er der nogen, der sender noget op, der ikke virker, og det er ikke specielt morsomt. Man kommer på forsiden hver gang. Det er de helt små ting, som gør at det hele ødelægges, så hele proceduren omkring at sikre at konstruktionerne er korrekte og afprøvede

i alle ender og kanter, de er meget grundige. Og man kan sige, at sådan en ting som afkodere er særligt slemme, for der kan jo være en fejl i udstyret, og det kan være kamufleret. De rigtig slemme fejl opdager man først, når den er i gang. Så gør den noget skørt.

Når vi har kigget på JPLs hjemmeside, og også NASAs, så skriver de meget om turbokoder. Bruger de dem?

JJ Det tror jeg ikke. Turbokoder er stadigvæk noget, der har status som alternativ til det etablerede system, og det findes på Casssini-Hyugens som alternativ kommunikation. Det etablerede system består af Reed-Solomon koder (255,223) og foldningskoder, og det har været standard siden 1970'erne. Det skøre tilsyneladende ved Mars-missionen er, at de har gravet en meget stor foldningskode frem, som har været prøvet engang som et eksperiment, og som efter alle andres mening er en sindssyg konstruktion. Prøv at finde ud af hvorfor de synes, at det er en god ide – andet end at de har investeret ufatteligt mange penge i at lave afkoderen.

Hvad bruger man på ESA?

JJ ESA bruger den mindre foldningskode, som har været brugt siden begyndelsen af 70erne med turbokoder som et alternativ, man nu begynder at implementere. I en vis forstand er Mars så tæt på, at der er masser af signal, og det er derfor ikke så kritisk at have en fejlkorrigerende kode. Det der foregår på den anden side af Saturn, det er noget helt andet, fordi kun en lille smule signal vil komme igennem, på grund af afstanden.

Hvad hedder foldningskoden?

JJ Den har en hastighed som er en halv, og det man plejer at bruge er en hvor hukommelsen er seks, men det de bruger nu er én hvor den er fjorten, tror jeg. Og det betyder, at antallet af tilstande, som man skal opbygge er to i fjortende i stedet for to i sjette. Vi andre mener, at det skyldes, at der var en eller anden skør ph.d.-studerende, som fik for meget frit løb, og lavede det der. Sammenlignet med forskellige alternativer, som er meget enklere at implementere, så ser det mærkeligt ud, at det der Storm P.-apparat står der og overhovedet nogensinde har fået lov til at blive lavet. Så, ja, spørgsmålet om hvorfor det lige er netop den kode der er valgt, det er til dels et godt spørgsmål. Der er to mærkelige svar: Det ene er, at det stammer fra tidernes morgen, hvor der stadig væk ikke eksisterede nogen teknik; og det andet er, at det skyldes en eller anden skør studerende.

Er der nogen ekstreme forhold på Mars, som gør sig gældende?

JJ Ingen energi.

Ved I noget om datakomprimeringen, komprimerer man alle data?

TH Jo, men der er jo nogen af dem, man ikke kan komprimere; billeder er oplagt, men andre typer af data kan ikke komprimeres. Hvis man for eksempel måler en temperatur er der ikke meget at komme efter.

JJ Billeder er mest oplagt, for de fylder kolossalt, og der er ikke meget information i dem. Der arbejder man mest med at komprimere, så de får en stor faktor. Det er noget med at klippe billedet i små stykker og transformere dem.

Er det sådan noget fraktalt noget?

JJ Nej, det kan ikke betale sig i den sammenhæng, men det er formentlig mest noget med Fouriertransformationer. Det ved jeg ikke noget om præcist.

Det lyder lidt fantastisk, at der ikke foregår nogen form for kryptering af det, der sendes fra Mars. Man kunne godt forestille sig, at NASA gerne ville hemmeligholde et eventuelt billede af en Mars-mand...

JJ Det er jo altså ikke så nemt at modtage signalet, man skal have en kæmpestor antenne, og hvor mange andre af dem findes der udover NASAs? Derudover er det meste data af hensyn til kommunikationssystemet randomiseret, og det gør det besværligt at afkode. Derudover skal man finde ud af hvor tingene står henne, og alt skal pakkes ud og stykkes sammen. Det er forholdsvis svært, når man ikke kender detaljerne, så jeg tvivler på, at der krypteres. Man skal altså kende hele systemopsætningen for at finde de data, man er interesseret i.

Når man snakker om Mars-missionen er der vel ingen tvivl om, at teknologien bliver presset til sit yderste?

JJ Noget af teknologien, men på en meget forsigtig måde. Det bliver ikke presset på den måde, at man laver noget, der er ved at falde fra hinanden.

Men tingene skal vel være små, når de skal sendes så langt?

JJ Nu ved jeg ikke om I har kigget på hvad for nogen integrerede kredse, der anvendes, men det har været én af diskussionerne: Hvad tør man sende derop og stole på virker. Det skal op igennem et højradioaktivt bælte omkring Jorden, og Gud ved hvad det bliver udsat for i rummet. I gamle dage fandtes der krav til det der kaldes space hardening, nogen chips, der var lavet af noget specielt materiale, hvor der var specielt langt mellem de forskellige gates, og der skulle en stor ladning til at ændre dem, så de kunne tåle et høj-radioaktivt miljø. Efterhånden er man gået væk fra det og bruger i stedet militære specifikationer, der er lidt strammere end de almindelige, men de er så gode nok. Hvis ikke folk synes, de er gode nok, kan de pakke et stykke folie om det, det hjælper jo altid. Det er for dyrt at lave specielle kredsløb. For eksempel det at sende mikroprocessorer op har været et kæmpe problem, fordi de er dyre at lave, og de der var godkendt har altid været meget langt bagefter. Okay, der findes mikroprocessorer på Mars, men det er altså ikke Pentium, det er nogen, som man bestemt ikke ville acceptere i sin hjemmecomputer. Der en skæg historie om den dér skildpadde på Mars, som gik fuldstændigt i baglås og måtte rebootes. Så viser det sig, at det er en fejl i DOS, som man aldrig har fået rettet, og det synes jeg da var helt sødt. Jeg troede sgu da ikke, at der fandtes nogen DOS-systemer mere, men det viser lidt om, at teknologien ikke går så hurtigt. Der er ikke Windows eller Linux eller noget andet, der er DOS. Og en 8086-processor, som man troede var på museum, men som er deroppe i en eller anden særlig hård udgave.

Okay, så teknologien bliver altså ikke presset til sit yderste...

JJ Jo, men du skal forstå 'presset' på dén måde. Det er et helt andet pres.

Kan man så snakke om, at der er noget matematik, der bliver presset?

JJ De har i hvert fald nogen skidegode matematikere ansat.

TH Ja, men det der med at ramme og finde ud af hvor ting er og sådan noget, det er jo stort set også standardiseret. Kompressionen ved man også, hvordan man foretager og kodningssystemet er velkendt.

JJ Men altså i kodningsteori er der jo masser af pres forstået på den måde, at vi på fem minutter kan stille jer et problem, som det ville være godt at kunne løse, som ville få tingene til at gå lettere. Men det er jo ikke nok at sige, at man gerne vil, hvis der ikke er nogen, der har en god ide. Der sidder jo folk og tænker grundigt over de der ting, men det er jo ikke altid nok. Fremskridtene kommer jo i den rækkefølge, de kommer.

TH Jo, men der går også et stykke tid fra noget er opdaget på teorisiden til det bliver implementeret i en eller anden mission.

JJ Ja, der går lang tid. Det går meget hurtigere med at få en ny smart ide ud i hjemmecomputere og modemmer og sådan noget. Der kommer det fra det ene år til det andet, og tit viser det sig nu om stunder, at hjemmecomputere er meget foran industrielle anvendelser, fordi det ingen rolle spiller, om de virker eller ej. Folk vil gerne have det nyeste, og billederne skal være flotte i spillene, så det er ligesom der, det nye får lov til at blive prøvet af. Men der er da masser af uløste problemer i kodningsteori, som folk gerne vil kunne løse. For er par år siden var der nogen som sagde: 'Tag de kodningssystemer, som man har med Reed-Solomon koderne, så kan man faktisk rette halvanden gang så mange fejl, som man plejer'. Og det er da ikke uinteressant. Det er ikke engang svært. Så det er da muligt, at man gerne vil gøre det. Det skal bare sælges til ESA og udvikles.

TH Men det laver ikke om på selve kodningssystemet. Det er det samme system man bruger, men det er en anden afkodningsmetode. Det betyder altså, at det samme udstyr er meget mere robust end man troede.

JJ Ja, og hvis man havde kunnet det for fem år siden, så kan det da godt være at den der dims, der modtager ting fra Mars, ikke ville være som den er nu, for det er da en gevinst. Det er jeg sikker på, at man gerne ville have opdaget for ti år siden.

Vi troede, at fejlretningen var en egenskab ved koden og ikke ved afkodningsalgoritmen?

TH Det er til dels rigtigt.

Så egenskaben ved koden er altså en slags garanteret fejlretning?

TH Præcis. Så længe du ikke laver mange fejl, så garanterer koden retningen. Der hvor det bliver interessant er, når du laver flere fejl. Hvad sker der så med din afkoder? Hvis den retter i halvfems eller nioghalvfems af tilfældene, så er det jo fint. Så selvom teorien siger, at garantien er noget andet, så kan man ved at bruge den her afkoder typisk rette flere fejl. Så det er ikke kun en egenskab ved koden. Ikke-perfekte koder kan meget mere end teorien siger de kan, med den rette afkoder.

JJ For at udnytte kommunikationen bedst muligt, skal man faktisk længst ned i hastighed, have flest mulige paritetssymboler i forhold til informationen for at få størst effektivitet. Man kunne tro, at det var omvendt, men det er det ikke. Og jo længere man kommer ned i hastighed, jo mere ekstremt er det forhold mellem hvor mange fejl, man garanteret kan rette, og dem man kan rette i gennemsnit, så det er dernede man gerne vil være.

TH Så i de nye afkodere er der bestemt noget ny matematik, som må betegnes som frontforskning. Det er ikke noget man kan se i en bog.

JJ Afkodningen foregår i to trin: Man afkoder først foldningskoden, som arbejder på de enkelte bit, og den har altid været sådan, at man afkoder fuldstændigt til nærmeste kodeord. Næste skridt er så at bruge Reed-Solomon koden til at rette de blokfejl, der opstår.

Du (JJ) har været involveret i ESA?

JJ Jeg har været i den gruppe, som laver modtagere, afkodere til Jordstationer. Nu om stunder bliver afkodere lavet i det man kalder programmérbare gates. Det vil sige, at man køber en stor blok af logik, som ikke er dedikeret til noget som helst, og så er det simpelthen et program, man lægger ind, som forbinder ledninger. Så efterhånden når man kommer på bedre tanker, kan man opdatere dem. Det er altså lækker teknologi. Det kaldes field-programmable, fordi det kan opdateres efter at systemet er taget i drift. Derfor er man bedre stillet overfor at rette de fejl, som værkstedet kan lave i chip'ene. Det er noget, man kan sætte sig ned ved computeren og lave. Det er den foretrukne måde at lave afkodere på. Det er noget, som vi kan lave, og så sender vi det videre til et firma, som rent faktisk laver implementeringen og tester det.

Hvordan foregår modulationen?

JJ Det foregår i flere trin. Du tager dit digitale signal, som det kommer ud af kredsløbene – det vil sige nuller og et-taler, altså to spændingsniveauer – og det skal igennem et modulationssystem og ende med at være noget i gigahertz-området, som kan sendes ud. Som regel er der imellem de trin en hel række pakketrin, hvor man gradvist modulerer signalerne op og pakker signalerne sammen i en bunke. Tilsidst ender det med at være én stor samlet sag, som bliver sendt af sted. Så man tager ikke de enkelte bit og laver til en bølge. Der foregår en hel masse digital pakning.

1.3 Lektor Søren Forchhammer 31/1 2005

Hvad er det du pt. er involveret i med ESA?

– Det er et ESA-projekt, hvor vi skal have et apparat med seks videokameraer op med en Columbus-rumfærge. Det er sådant set nogle af de samme tanker, der ligger bag, som ved Mars-missionen.

Har du været tilknyttet rumforskning før?

– Nej, det er mere perifert. Vi regner med at få denne kontrakt igennem, men det er så den første.

Er det sådan det fungerer, at ESA kontakter folk på universiteterne?

– Nej, det er lige så meget den anden vej rundt. I det her tilfælde er det en gruppe inden for det der hedder Dansk Rumcenter, som har taget initiativet. De vil godt have lavet det apparat. Det de vil filme er lyn eller fænomener oven over lyn, det vil de gerne filme og måle både med røntgen og gamma. Så det er dem, der er udadfarende, men måden at få det i værk på, det er ved at få ESA-penge til det.

Hvad baserer ESA sin kompression på, ved du det?

– Nej, men det er noget af det vil skal have checket op på. Jeg er ikke up-to-date.

Det er altså ikke noget, der er fuldstændig standardiseret, man ku' jo ellers godt forestille sig, at man bare slog op i den store ESA-bog under 'datakompression'?

– Nej, ikke 100 procent. En af grundene er, at det her er lidt specielt, ved at man kun vil måle og filme, når der er lyn fordi man hverken har lager eller transmissionskapacitet til at filme hele tiden. Så man må være meget selektiv. Man må finde ud af hvornår der er lyn eller hvornår, der er de bestemte fænomener, og så kun sende de relevante data ned. Så i det her tilfælde skal vi undersøge om datakompressionen skal kobles sammen med en event-detektor. Men vi skal selvfølgelig se hvad ESA har på hylderne.

Findes der ikke nogen 'rumstandarder', ligesom der gør det indenfor kodningsteori?

– Jo, der er nogen standarder. Nu er det vel relativt nyt, at man sender kameraer op. Hvis I tager det, I sendte mig, så har de ligesom lavet deres egen variant, men de har jo taget udgangspunkt i de standardmetoder, man bruger. Det bliver lidt det samme med ESA-projektet, nu skal vi selvfølgelig se hvad de har.

Altså JPEG-baseret kompression?

– Ja, JPEG dækker jo i dag de forskellige teknikker der er. LOCO til tabsfri er den, der ligger i JPEG-LS. Det er en prædiktiv kode, hvor man i et billeddomæne prøver at forudsige hvad den næste pixelværdi er, og så sender man bare differensen i stedet for at sende pixelværdien. Så er der jo den gamle JPEG, som er DCT-baseret [Diskret Cosinus-Transformation], i modsætning til den nye JPEG 2000, der er wavelet-baseret. Det er vel de dominerende teknikker.

LOCO bruger Huffman-kodning?

– Ja, man kan sige, at billedkodning foregår i to eller tre trin. Hvis det er tabsfrit vil man typisk lave en eller anden forbehandling, og så vil man anvende en såkaldt entropikode, hvor LOCO anvender Huffman. Der er også noget, der hedder aritmetisk kodning. Hvis man koder med tab har man mellem forbehandlingen og entropikodningen et præciseringstrin, der kontrollerer hvor stor præcision, man arbejder med. På den måde taber man noget, men man kan så også opnå højere kompression. Hvis vi tager den gamle JPEG, så har den DCT-kodning eller DCT-transformation i første trin, præcisering og så Huffman-kodning. Så det er jo i familie med wavelets, det er noget hvor man går over i et frekvensdomæne. Hvis vi tager JPEG 2000, så er det wavelet-kvantisering og så aritmetisk kodning. Og det skal jo så siges, at både til JPEG-LS og til den gamle JPEG kan man sådant set også anvende aritmetisk kodning og således få en lidt bedre kompression, men det er mere komplekst og der er også nogle patenter ...

Hvad går man mest efter i ESA-projektet, vil man have lav kompleksitet eller vil man have høj kompression?

– Ja, det er jo noget af det vil skal se... Man kan sige, at generelt til ESA er der tale om en ret kritisk applikation, det er jo dyrt at sende udstyret op, så man vil gerne gøre noget, der er meget avanceret, men der ligger jo så også nogle begrænsninger i; hvor meget processorkraft man har, og hvor meget power der er. Hvis vi nu tager det jeg er i gang med, så ville jeg jo gerne bruge wavelets og aritmetisk kodning fordi det giver den bedste kompression. Så ligger der en vurdering af hvorvidt der er processorkraft til det og om processerne simpelthen koster for meget power.

Det gør de normalt?

– Ja, altså JPEG 2000 med både wavelets og aritmetisk kodning er mere krævende end den gamle JPEG med DCT og Huffman. Og i virkeligheden er det entropikodningen, der er den tungeste i JPEG 2000 i dag, hvor man ved aritmetisk kodning først benytter noget, der hedder kontekstmodellering. Det går ud på at man med aritmetiske koder anvender sandsynlighedsestimater for hver eneste bit, så det vil sige, at der er ligeså mange flere symboler, man skal kode, end hvis man tager det hele med, som man gør ved Huffman-kodning. Så for hver eneste bit vil man beregne et sandsynlighedsestimat ved at kigge på en omegn af tidligere bits og så ud fra denne omegn opsætte en betinget sandsynlighed. Det er altså en betinget sandsynlighed man bruger. Denne kontekstmodellering er det tungeste JPEG 2000 beregner.

Vores to cases er datakompression og kodning. Vi synes de hænger godt sammen...

– Jeg ville sige, at begge to har overskriften kodning eller kodningsteori og så ville jeg splitte det i to. Nu går jeg tilbage til det klassiske – Shannon – så datakompression hedder kildekodning (source coding) og fejlkorrigerende kodning, det hedder channel coding. Et af hans (Shannons) klassiske resultater er at hvis vi ser på det asymptotisk, så kan man uden tab adskille de to. Det er selvfølgelig meget ingeniørmæssigt...

Det gør jo at der bliver tale om to teorier...

– Ja, ja, og det har jo også kørt meget adskilt. Så kan der være nogle praktiske situationer, hvor man kan sige, at forudsætningerne ikke holdt, men langt hen af vejen kører man efter det gamle paradigme, der siger, at man tager dem hver for sig. Altså først laver vi datakompressionen og så transmitterer vi. I det paper I gav fra Mars kobler de det jo lidt sammen, idet de siger, at der er en meget lang responstid, når der sendes ned, så fejl kan ikke undgås. De ser så på hvad der skal gøres på billedkodningssiden i forhold til de her fejl, så de bygger noget fejltolerance ind i deres billedkodning og har sådan set koblet det lidt sammen. Det er jo så begrundet med nogle praktiske forhold i opsendelsen.

Som vi har forstået det, bliver koder som Reed-Solomons først relevante på rummissionerne i forbindelse med billedkompression. Kompressionsalgoritmerne er sensitive overfor fejl, de kan kun acceptere op til en eller anden grænse. På den måde synes vi jo også at vores cases er stærkt forbundne.

– Ja, specielt til rumanvendelserne er det oplagt også at se lidt på samspillet imellem de to.

Kan man sige, at datakompressionen og kodningsteori i en vis forstand lever i teknologien?

– Jo, nu ved jeg ikke lige hvordan man tolker det, men meget af det der sker i dag på den praktiske side af udviklingen, det er jo at man får mere og mere computerkraft så man kan bruge mere komplicerede, avancerede metoder. Det er jo selvfølgelig klart at der kan være nogen, der har lavet det før rent teoretisk, men der kommer jo også man-power og store virksomheder bag...

Benytter man datakompression nogen steder uden for en computer?

– Det er jo en definitionssag. Man kan sige, at morsealfabetet har indbygget datakompression. Det er ligesom det klassiske historiske eksempel. ... Går man tilbage til Bell Labs var det vel kommunikation og den anvendelse, så er det jo en definitionssag om man har computere eller ej når man laver kodning, men man har et eller andet processorkraft.

Ja, det ville være kedeligt at gøre i hånden.

– Men det er jo klart det bliver en definitionssag, ligeledes i de tidlige fax-dage. Det var jo også i en vis forstand en beregningskraft.

Hvad er det for noget matematik man skal kigge på når man snakker wavelets, er det kun diskrete wavelets?

– Hvis vi tager JPEG 2000, og det man bruger til billedkodning, så bruger man såkaldte biortogonale wavelets. I matematikken er det inden for det man kalder en Riesz-basis. Grunden til at man bruger det frem for de oprindelige wavelets som Daubechies lavede, er at man i praktiske anvendelser vil arbejde med endelige diskrete wavelets. For at få nogle, der er symmetriske og af ulige længde, kan man ikke klare sig med en ortogonal wavelet-basis, men man må over i en biortogonal basis. Man kan diskutere om det er kontinuert eller ej... Den måde det er defineret på, det er at når man har et billede, så vil man starte med et diskret billede, som man koder. Hvis man skal koble det sammen med wavelet-teorien – sådan som jeg ser det – så hopper man ind i den opløsning, og så implicit, underforstår man nogle kontinuerte basisfunktioner nedenunder. Så når man har valgt sin wavelet-basis, så vil man i virkeligheden sige: 'Jeg tager mit udgangsbillede og hopper ind på det her opløsningsniveau og så kan jeg med de basisfunktioner, jeg har, lave et kontinuert billede, fordi de diskrete wavelets der arbejdes med er udledt udfra kontinuerte funktioner.' Men som ingeniør kan man jo glemme alt om matematikken og bare tage dem som diskrete.

Er wavelets opfundet i forbindelse med datakompression eller kommer de et andet sted fra?

– Det klassiske eksempel er Haar-wavelets som går tilbage til omkring århundredeskiftet, og det er mig bekendt ren matematik. Der er selvfølgelig adskillige der

har arbejdet med det, men det man fra den praktiske side har været interesseret i er en samtidig beskrivelse i frekvens og tid. Hvordan gør man det? Vi kan godt tage musik som et klassisk eksempel. Der er jo frekvenser, men det giver ikke så meget mening at tale om dem til uendelig tid. Ingrid Daubechies var også på Bell Labs da hun lavede det, men hun er matematiker. Jeg tror mere det var den her grundlæggende ide om den samtidige tid- og frekvensanalyse, der var hendes indgang. Så hun er matematiker, men hun havde også en relation til det praktiske, og det har hun også flot dyrket siden. Hun var med til at lave det med de biortogonale wavelets. Den sidste ting jeg måske vil nævne er det, der hedder lifting. Altså, det er vist, at hvis man har en biortogonal wavelet, så kan man dekomprimere i en række simple trin. Man kan lave effektiv tabsfri kompression med wavelets i JPEG 2000. Med den gamle DCT-baserede kan man ikke. I praksis står man med en 8-bit repræsentation af sine data og dem vil man gerne have rekonstrueret perfekt. Hvis man så går over i frekvensdomænet, så er det svært her at styre, hvor stor nøjagtighed man skal have for at stå med den rigtige 8-bit repræsentation, når man går tilbage til det oprindelige domæne. Det kan man ved hjælp af lifting. Så det er sådan set en generalisering. Det er vist, at man kan tage enhver biortogonal wavelet, dekomprimere den, og så kan man lave den ulineær – det er det der skal til – og nu kan man så i de her lifting steps kontrollere hvor stor præcision man vil have. Det er matematikken i JPEG 2000.

– Man har sit signal og så tager man et højpasfilter og et lavpasfilter, der giver henholdsvis scaling og approximation, så nedsampler man dem, smider hver anden værdi væk. Man har samme wavelet-repræsentation, hvis det er en ortogonal og en biortogonal wavelet, forskellen er blot, at hvis det er biortogonalt så er der to filtre – der udledes det ene ikke ud af det andet, de udledes ligesom sammen.

Er det ikke forholdsvist nyt, at man er begyndt at bruge wavelets så meget?

– Jo, det som Ingrid Daubechies lavede var tilbage i slutningen af 1980'erne, så det er derfra det har taget fart. Den artikel, hvor de laver billedkodning med wavelets, den kom i 1992. Billedkodning er jo en af de store anvendelser af wavelets, men der er også noget signalanalyse etc.

Med hensyn til tabsfri komprimering og komprimering med tab, kan man i rummissioner sige at tabsfri går på kontroldata, hvor man med billeder for eksempel godt kan acceptere et eventuelt tab?

– Generelt vil kontrolinformationer være uden tab, man skal så måske i praksis eller hvis det er kritisk alligevel være i stand til at håndtere de situationer, hvor der er tab. På billedsiden til rummissioner, er udgangspunktet at de helst vil have de her meget dyre billeder ned tabsfri, men der kan komme et trade off, hvor man ser på hvor meget man kan få ned. Jeg tror rumindustrien har været lidt konservativ, hvor man i starten har sagt at det skulle være tabsfrit, men det er så senere blevet demonstreret, at man kan få mere billedmateriale ned, hvis man var villig til at acceptere tab.

Men det er små tab, eller hvad?

– Det kan man jo så generelt styre selv. Der er et 'trade off' mellem hvor mange bit per pixel man skal bruge og hvor stort tabet er.

Wavelets er simpelthen bedre end Fourier?

– Ja. Der er sådan set en svaghed både for DCT og wavelets. Hvis man har en kant i billedet skal man bruge nogle overtoner til at få det hele til at passe. Men der er wavelets så bedre end den DCT-baserede.

Hvor foregår forskningen i datakompression?

– Ja, historisk eller... Går vi tilbage til Shannon er det jo Bell Labs og MIT. Billedkodning har de også arbejdet med der. Hvis vi tager wavelets – DCT'en ligger lidt før min tid – det var igen Bell Labs med Daubechies. Men efter det er kommet frem har det været mange forskellige steder.

Men er det koncentreret i universitetsmiljøet eller også i private forskningsinstitutioner?

– Det er begge dele. Der er store erhvervsvirksomheder med i JPEG 2000: Kodak, IBM, de japanske; Mitsubishi, europæiske; Siemens...

... og det betyder at der er patenter involveret?

– Ja!

Er det derfor NASA udvikler egne kompressionsalgoritmer, i stedet for at bruge for eksempel JPEG 2000?

– Grunden til at gøre det hos NASA og ESA vil være at de applikationer er så specielle, at selv om de måske bruger de samme grundlæggende teknikker, så kan der være grunde til at man i forhold til rumapplikationer gerne vil gøre det meget specifikt. Som jeg husker det I sendte fra NASA med hensyn til wavelets, så gjorde de noget for at gøre det simplere end JPEG 2000. Noget af det, der ligger i JPEG 2000, er at det kan bruges til både naturlige billeder og computergrafik eller blandinger, scannede dokumenter eller fingeraftryk. Så med JPEG 2000 sigter man væsentligt bredere. Men hos NASA er det naturlige billeder man tager, og derfor kan man få den samme kompression, men med lidt simplere teknikker.

Ifølge Høholdt og Justesen anvender man til kanalkodningen velafprøvede koder, som man ved er pålidelige. Er det på samme måde med kompressionen? Det virker som om ICER er noget helt nyt, altså at der her er større mulighed for at eksperimentere med nogle nye ting?

– Hvis man sammenligner fejlrettende koder og datakompression så er man inden for datakompression tættere på de data, man koder. Når man laver billedkodning, forsøger man at arbejde med de billeder, man har, mens man med fejlkorrigerende koder anvender en mere ren matematisk opsætning. Så det er i hvert fald en forskel og det slår vel også igennem her med Mars-missionen...

– Jeg vil ikke sige, at ICER er så ny. De grundlæggende teknikker er de samme. Det er det, der foreligger op til og med JPEG 2000. Og noget af det de gør, er i virkeligheden at gå lidt længere tilbage, de skræller noget af det nyere i JPEG 2000 omkring de her kontekstmodeller og aritmetisk kodning af. Så det er relativt gennemprøvet.

I nogen af de rapporter vi har set skriver de at det meste af datakompression foregår i software. Det undrer os, fordi kanalkodningen ofte foregår i hardware-kodere. Er det fordi hastigheden ikke betyder lige så meget ved datakompressionen?

– Med hensyn til det ESA forslag vi arbejder med, der gælder at der kommer meget store datamængder ind. Når der så er nogle billeddata man gerne vil gemme, så går det så hurtigt med de data, der kommer fra kameraerne, at der skal laves en hardware-løsning der. Så bliver det lagt ned i et lager, hvorefter der vil være en processor, der står for yderligere kompression i software. Man kan sige, at grunden til at vi gør det sådan, er at det i første runde går så hurtigt at vi er pisket til at anvende en hardware-løsning. I anden runde er vi nede i datahastigheder, hvor man godt kan klare det og så er det lettere at lave mere avancerede ting på en generel processor. Det er et stort arbejde at lave de tilsvarende løsninger i hardware. Så hvis vi skal tage det som eksempel, så vil jeg sige, at hvis man har en processor af en given styrke, så vil man udnytte den så vidt den kan klare det.

Bruger man software betyder det vel også at det er nemmere at skifte ud, for eksempel igen i modsætning til de fejlrettende koder, der anvender hardware og derfor må følge de her fastlagte standarder.

– Det er nok også lettere på billedkodningssiden, for hvis vi ser på kommunikationssystemet – hvis vi kun tager kodningen – der har man en kilde, så har man datakompressionen, så har man kanalkodningen og så pakker man det ud igen, så man får sine data. Det vil sige i vores tilfælde, og det gælder sådan set også Mars-ekspeditionen, vil den billedkompression, man laver, være knyttet sammen med de kameraer man har sendt op, så det er jo ikke sådan at man vil låse sig fast til altid at sende samme kameraer op. Man er ligesom tættere på missionen i en vis forstand og det vil jo også gøre det lettere at lave om på, hvis vi går tilbage til ESA eksemplet, så skal de jo alligevel lave noget software...

  [Midlertidigt afbrudt.]

– Rent historisk må man vel sige at de fejlkorrigerende koder var på plads tidligere. De har jo været grundlæggende i kommunikationssystemet; vi må have de fejlkorrigerende koder ellers kommer vi ingen vegne. Og det er så først noget senere at det bliver relevant at lave billedkompression.

Men billedkompressionen påvirker vel så igen de fejlrettende koder sådan at de nu skal være bedre.

– Jo, men noget vil handle om at man har en hel klasse af standard koder, og så kan man vælge ud fra hvilke hastigheder vi kører med. Men det er jo rigtigt på den måde, at når vi laver datakompressionen eller at desto hårdere vi komprimerer, desto værre bliver fejlene så desto mere vil vi fejlbeskytte. Hvis man gør det med aritmetisk kodning og der opstår bare en enkelt bitfejl, så er resten mere eller mindre tabt, hvor med Huffman-kodning, lidt afhængig af hvordan det er, så kan man godt komme tilbage på sporet.

Vi har læst at der ikke går mere end 10-12 år fra udviklingen af wavelets til de finder deres kommercielle anvendelser. Kan man sige at vejen fra matematisk udvikling til anvendelse indenfor datakompressionen er kortere end indenfor mange andre områder?

– Ja, det er hurtigt. Men jeg ved ikke om det er så entydigt, nu tager I selvfølgelig wavelets som er en del af det her... Der er jo også nogen tilsvarende, der har lavet subband-kodning uden at de egentligt er wavelets, men har den samme struktur. Så jeg vil måske være fræk at sige, at man kan lave noget lige så godt, næsten lige så godt, uden wavelets. Wavelets viser sig nu om dage at være de bedste til billedkompression, men ingeniørerne skulle nok have skruet et eller andet sammen der måske var lige så godt. Det er den ene side af det, den anden side af det er, at Huffman, han lavede sin kode i 1950erne på MIT Elias, der også var på MIT lavede også aritmetisk kodning – men bare i en form, der ikke kunne anvendes. Så det er jo ikke fordi det er så entydigt. Der har ligget mange af de andre elementer, og man har arbejdet med DCT og alle de her ting. Så waveletten kom ikke ud af det blå. Men det den tilbyder, er det der nok viser sig at være det bedste. Men der var altså nogen der tidligere havde lavet subband-kodning. Tager vi grundideen med at vi har et høj- og et lavpasfilter, det kan man finde masser af, det ligger også i et stereoanlæg med at man har bassen og diskanten. Men altså også til subband-kodning, det er samme blokdiagram, hvor man har et høj- og et lavpasfilter, men som ikke matematisk set har været wavelets. Så er det at wavelet-teorien faktisk giver bedre filtre.

– Så jo, jeg vil sige, det er hurtigt fra den grundlæggende matematik er lavet og så til standarden ligger der.

Kan du eksempler på påvirkninger mellem ren og anvendt matematik, eller anvendt matematik og teknologi i den her forbindelse eller inden for rumforskningen?

– Et eksempel kan være lifting, eller måske også subband-kodning. Der blev jo lavet noget tabsfri wavelet-kodning, med noget der var lifting uden at der var en teori bag det. Og det tror jeg da har været et input til dem, der siden har lavet liftingarbejdet. Daubechies var med til det og Calderbank også. Så det tror jeg godt man kan sige. Og det er måske også i virkeligheden spørgsmålet om subband... jeg vil tro, at Daubechies har været bekendt med subband-teknik. Men i begge tilfælde kan sige, at man har stået med noget, som har en anvendelse og er interessant, men som mangler en matematisk teori. Og i og med at de går ind og udvikler eller arbejder med den teori, så står de lige pludselig med noget der er et lidt mere generelt værktøj, end det man startede med. Så det tror jeg godt man kan sige på den måde.

– Med hensyn til rumforskningen. Der er jo et eller andet sammenspil. Wavelet har jo fået enorm interesse. En af wavelets styrker har vel også været at det både har været interessant teoretisk og praktisk. Inden for billedkodning til rumforskning har JPL været meget aktive, for eksempel med Rice-kodning. Jeg tror også, at det at JPL har været der, og at der er en anvendelse, har en funktion i forskningsverdenen, altså det at der er nogen der arbejder med det. Der ligger noget omkring det der med at kombinere fejlkorrigerende koder og datakompression som ikke er helt på plads endnu.

Hvordan mener du?

– Vi mangler vel lidt teori. Man har ligesom den teori, der siger at man kan skille tingene ad – det er jo så, hvis man ser på det asymtotisk. Men vi mangler måske lidt en teori, hvor man kan opsætte nogle begrænsninger eller nogle forudsætninger og så sige, at under de her forudsætninger så skal vi dekomprimere og lave den fejlkorrigerende kode på den og den måde.

Er det mest eksistensresultater ligesom Shannons man har, eller har man også mere konstruktive resultater som for eksempel Justesens?

– Begge dele. ... Jørn Justesen vil jo helst have konstruktive beviser, og det tror jeg, der er mange der vil foretrække. Inden for datakompression kan man komme med eksemplet om LZ-kodning. Der er deres to teknikker eller artikler fra 1977 og 1978. Jeg ville kalde disse rene teori-artikler, men der lå i begge to noget af en generel datakompression. Der skulle for så vidt tages et skridt fra deres beskrivelse til hvad der er effektivt eller praktisk. Det de går ud fra rent teoretisk, er at sige, at nu kender vi ikke den model der er, som dataene er genereret ud fra, vi antager bare at ordnen af den skal være endelig. Og under de forudsætninger viser de at den er asymtotisk og at datakompressionen konvergerer mod entropien. Når man laver overgangen til uendelig så vil det antal bit man har brugt per kildesymbol konvergere mod entropien. Men den gren kommer ikke for alvor ind i billedkodning, vil jeg sige.

LZ har ikke været anvendt til billedkomprimering?

– Det kan godt være det har været anvendt på billeddata, men det kører jo bare på en en-dimensional streng. GIF baserer sig på en LZ-variant, men er ikke ligeså effektiv som egentligt billedekompression og grunden til det er altså at den tager det som en en-dimensional streng, så man mister den relation der er i den anden dimension...

– Grundideen med både den gamle JPEG og JPEG 2000 er, at man kan lave visuel tabsfri kodning. Det man smider væk er noget øjet ikke kan se. En grund til at kodning i frekvensdomænet også er interessant, det er at vi ikke kan se de høje frekvenser lige så godt som de lave frekvenser, synet er mindre følsomt overfor de høje frekvenser. Det bliver så udnyttet i den kvantisering der foreligger, specielt i det gamle JPEG. ... Så er det jo også noget andet med rumdata, hvis man bruger det til videnskabelige beregninger.

Grunden til at datakompression bliver så interessant, er det at man får større computerkraft og dermed øgede muligheder for lagring og lignende?

– Nej, lige så meget fra kommunikationssiden. Videotelefoni for eksempel. Jeg vil nok sige, hvis man går længere tilbage, så er det nok kommunikationssiden. Man kan godt sige, at det først har taget fart med computeren....

– Videotelefoni er et eksempel på en teknik, man i lang tid har snakket om. Det kan godt være at det først er ved at komme nu. En af de første anvendelser med hensyn til billedkompression var faxen.

Ja, men det var vist det...

– Ja, men som sagt, hvis vi kan udveksle noget omkring generel billedkodning til rumanvendelser. Hvis I får nogle standarder, eller bare navnene på nogle standarder. Vi skal også på et tidspunkt til at finde ud af hvad ESA har, men der er vist ikke noget offentligt sted, hvor man bare kan gå på nettet og søge.

2 Interview fra Brown University

Nedenstående samtale med Professor Emeritus Philip J. Davis er foretaget på Brown University i Providence, Rhode Island den 6. marts 2005.

Figur 2.1 Til venstre: Professor emeritus Philip J. Davis. Til højre: Brown University Department of Mathematics i forgrunden og Division of Applied Mathematics i baggrunden.

2.1 Professor emeritus Philip J. Davis 6/3 2005

– [...] Now you are going up to MIT to interview Gilbert Strang and then you are going to the JPL to interview a number of people. I think that there are various aspects that you can put in. I don't think that that many pages you have to devote entirely to mathematics. I think you want to go also for what you might call the human side of the story. When you interview these people try to get something about their background, what were their experiences before they came to the JPL, what their training is, whether it's in computer science, mathematics, physics, whatever. Try to find out how they operate within the group, probably you'll find that each one is part of a group, how the group operates, how it connects or meshes with the other groups in the JPL and so on. Try to find out what each one does, how much independence they have, how much opportunity is there for them to develop new ideas or new things whether it's in mathematics or whether it's in writing algorithms, software and that kind of thing. I think you want to make a story here of what it means to work in a space program as a mathematician or a computer scientist or whatever these people are. Then of cause you do have the opportunity to explain a little bit about the mathematics, I mean if you are talking about data compression and so on, maybe a little bit about the history of data compression. Maybe you want to find out whether there are new developments in this that are found to be more efficient or maybe new computers are coming in with more potentiality and so on and how this connects to the communication from the satellites that are going around and sending data back. I would say, try to be a little bit like journalists rather than teachers of abstracts mathematics. Look at it this way: when you get through with all the pages there may be ten pages, rearranged and so on, that you could even send to the newspapers for general interest – I think you could make an interesting story there. One of the things you might find out from these people – this might be difficult though – how old they are

and how they see their professional future; is their profession always at JPL until they retire, do they go into teaching of some sort, do they want to have a start-up company of their own – it's quite possible – selling some sort of software packages... Anyway that is my idea of what your thesis might look like.

– I had one student about 5 or 6 years ago, very talented young women who had been in elementary teaching, teaching young kids, you know, 12, 13, 14 years old or something like that. And then she wanted to become a professor of mathematical education, she didn't want to become a mathematician, but mathematical education. I said: 'One of the things that are missing in the books that I have read is... actually, when you get into a place where mathematics is used, precisely what mathematics is used, how much does the people...' She said: 'I get the idea.' She was studying out in California, in Stanford. She went to an engineering company that was doing both architectural work and contracting work and so on and so on. She got onto a young man there who was engaged in building some sort of large garage that had certain functions to it... I don't know what they are. And she followed him for a number of weeks as to find out what mathematics he was doing; was it advanced mathematics, was it simple mathematics you know, and what aids he had in doing this, was he doing computing, was he using reference books, where was his information coming from, how was he interacting with his colleges who were not doing the same kind of thing? Anyway she put all of this together; it was a long thesis of almost day to day experiences in this engineering company. Eventually she got her degree and she is now an assisting professor of mathematical education somewhere in California. Her job is teaching teachers of mathematics who are going into elementary school, going into high school, into gymnasium or something like that.

– Now you'll be out there for what... a week or so? Maybe there will be an opportunity for you to go around with some of these people that you are interviewing, it's possible, if they will allow it, I don't know. If they will allow it that will be very nice and then you'll see exactly what it is that they are doing, all of these other dimensions, these features, what mathematics it is and in what form it comes, are they using packages or are they developing their own stuff, how is this going to fit in to future projects that NASA has? I think it is quite a challenge for you, but it should be very interesting.

Besides our two primary cases; data compression and channel coding, we are also looking to do a more overall and general survey of what mathematics is involved in the recent Mars shot and perhaps some of the history of the involved mathematics. You mention in your article A Letter to Christina of Denmark *...*

– Oh, you read that. Those are actual questions; I didn't make this up, you know. She asked me those questions. [...]

You say: 'Consider the recent Mars shot' and then mention how much mathematics is involved in that and that it would take a very knowledgeable historian of mathematics to make a survey of this math and the history of it.

– Yeah, because all of these things go back, they all have roots many years ago. So for example it may be that in data compression the Fast Fourier Transform is employed. That has a history that goes back two hundred years... The digital age gives it a shot in the arm so even the things that Fourier couldn't have done... things too long, too tedious, too boring... now you do it in micro seconds. Also wavelets and so...

One example is the Reed-Solomon codes used in channel coding. They are based on the theory of Galois which goes back to the 1830s.

– Yes the theory of Galois and the algebra in it goes back to the 1830s. There is group theory. Is it all these abelian groups or is it noncommutative groups also, do you know? Okay, you could give some background there. I think that the thesis has to have a balance of different kinds of material; personal material, biographical material...

– I've never been to the JPL. I don't have any students that are working there or colleagues or anything. So it would be interesting for you to give me a feeling of what the atmosphere is like. I used to work for NASA after I got my undergraduate degree. This is many years ago; sixty years ago [laughs]. So I have a feeling of what it was like to work in the aviation group many, many years ago.

It was before the space program. Space wasn't even invented. The center was an outcome of World War II with the missiles and so forth. So you can trace that program right through. The weapons that was developed in World War II; V1, V2... that came down on London and so on.

So in the two cases there are lots of mathematics; algebra, wavelets, Fourier transforms and you can look at this from an historic point of view. In the general survey there's the trajectories and...

– ... dynamical systems, planetary dynamics, Newton's laws and so on. I would imagine that at a place like JPL practically every branch of mathematics comes into play in some way. But I would also imagine that if you talk to a specific person the range of mathematics that that person is using is rather limited, much more limited... I would expect everything; partial differential equations, control theory, feedback and control mechanisms...

What we were hoping for you to help us with was to set the scene for such an investigation. You mentioned the human aspects...

– The human aspects is one. Maybe a little bit of the philosophical aspects of the thing; why do they want to go to the moon, why do they want to go to Mars, why do they want to send up the Hubble Telescope and find more and more phenomena, the cosmos... It seems to me endless and these programs are not cheap, it costs lots and lots of money. What do they expect to get out of it? And of course the scientists say 'well, we are getting more information about the universe', but it seems that it never comes to an end [laughs]. There is always some new thing that is discovered; some new particle, some new configuration, new theories of 17 dimensional [laughs] worlds or something like that. So you could say 'how does this come back to the general public?' The general public is paying for it in some sense. There is another human aspect of this, and this is one that NASA often makes when they go to Washington and ask for money; what are the spin-offs, for example the digital camera, is that a spin-off from the space program. In some way it might very well be. The executives of NASA will argue for more billions of dollars and say 'we have a list of 30 or 40 spin-offs, that you wouldn't have unless...' And then of course there is the defense thing that is important and the offense thing, I mean the spy satellites... they can look down and see what we are doing here, that kind of thing.

Another possibility is to let the overall frame of the thesis be the 'stagnation of science' in the last couple of decades. For example in biology there might not have been a really large breakthrough since the discovery of the double helix, and in physics there hasn't been a big breakthrough since quantum theory.

– Quantum theory is eighty years old and the helix theory is like fifty years old or something like that, so in a certain sense that is correct. Yes well, one of the things you might find out is that any fundamental development in biology or in physics is a result of the space program. But I don't see it yet, I don't see it...

No, but state-of-the-art science could be in the space programs so our job could be to discover where the development is going.

– Precisely yeah, I'm sure you'll find state-of-the-art stuff there. You might also find secret stuff there that they won't tell you about.

But do you think we'll find any new mathematics or will it just be...

– That they are not making public. I don't think so. I think you'll find maybe packages, computer packages, computer algorithms that they haven't released to the public. I would think so, you might hear about them, but of course they won't give you a copy of them – particularly those things that relate to military matters.

Regarding the spin-offs there seem to also be some kind of spin-on from everyday technology.

– Sure, they're going to take everything and use it. Well to describe the spin-ons would be good... I don't think that the development of the digital computer was independently a result of the space program. It was a result of many demands, many military demands actually; naval, army plus civilian demands. The mechanical computers were first used for problems within insurance... for keeping track of insurance records and stuff like that. It had very little to do with the military.

– You describe this general framework of the thesis, I think that is good; the historical aspects, the human aspects... Given these demands and things we don't like, particularly things that leads to trouble. You take a piece of modern equipment like these laptops for example, the insurgents in their opt can use them too, they can use the Internet for whatever purposes they have. This is the downside, you might say, the downside of new technology. You can't always anticipate what the future holds for a piece of technology in terms of human values. [...]

Regarding the stagnation: In your lifetime you have experienced certain new events like being able to travel by airplane, the invention of the computer and television and so on and for our great grandparents; the telephone. But what has happened in our lifetime except for maybe the remote control and variations of this?

– [laughs] Well you could also ask what innovations, in your lifetime, came up in mathematics. And there are a number of things that came up... some of them had great publicity at the time but they haven't turned out to be much. One is fractals; I don't think that has turned out to be of much use or anything. Another one is chaos theory, which is part of, I guess, differential geometry, I don't think this has proved particular useful. But things like wavelets... The whole development in mathematics in accumulation of little things like computer aided geometrical designs; it involves this little idea, that little idea, this piece of software, that piece of software, this application to automobiles, this applications to ships and to airplane surfaces and so on and so on and the development of standard packages to these things. Now that's all mathematics that certainly wasn't around when I was a student and most of it was developed during your lifetime. Now there is nothing here that is tremendously spectacular in sense of writing a big newspaper article; front page, new breakthrough and so on and so on. The newspapers like these very dramatic things... not a slow piling up of information, ideas, techniques, computer applications and so on. No particular step is spectacular in a way but it all adds up to a new way of looking at things, a new way of doing things...

– [...] Some of the things that are proposed in for example cosmology, like the string theory, I don't think it has proved out. At the moment it is not possible to verify it by experimentation and as I understand it, I think that it has given rise to lot of mathematics. Whether it is new mathematics I don't know... there may be some new ideas in mathematics there, it's possible...

Your description of mathematics as a slow accumulation of new knowledge... Is mathematics different from the other natural sciences in that way?

– I think it is hard to point to things that you can call really breakthroughs in the sense of biology or in the sense of physics. A breakthrough is really a journalistic kind of concept. A journalist might hear that Fermat's last theorem had been proved and he might write a big article on the front page of the newspaper in Copenhagen: 'New Breakthrough in Mathematics'. Well in a certain sense, yeah; a very old problem is now solved, but what does it mean to be solved and so on... What effects will it have on the lives of the inhabitants of Copenhagen? Practically none! The food will be the same, the message of distribution, the automobiles will be the same and so on and so. You can say going back now seventy or eighty years, Gödel's theorem in mathematical logic; that it is impossible to prove certain things. Now that's very important in logic, but you can ask the same question: What effect did it have on our daily lives? None! So, on the other hand things like DNA... or going back even further the discovery of penicillin and antibiotics, they have a tremendous effect on our lives. And pieces of technology... the transistor, it has a tremendous effect on our lives... that's a piece of technology the transistor... a chip. My personal philosophy of mathematics is that I am very hesitant to talk about breakthroughs. ... I'm not saying that there aren't brilliant mathematicians that do brilliant things. That's true; there are brilliant mathematicians that do brilliant things. Gödel is one and Fermat is one... These are brilliant people. But how it is affecting our lives isn't clear.

– I'll tell you this: There's pure mathematics, there's applied mathematics, there's logic, statistics, computer science and so on. Pure mathematics wants support from the central government just like all these other fields wants support. The support in pure mathematics comes largely to universities. The support for

applied mathematics comes in part to universities, but it might come to outfits like NASA or outfits that do research in warfare strategy or something like that. When the Congress asks the pure mathematicians why they should support them, they say 'well, pure mathematics has the potential of being applied though I can't say that that this piece of pure mathematics is going to be applied in our lifetime, but the potential is always there' and that is true of course. The potential is always there, but the probability for a particular piece of pure mathematics that is supported by grants is rather low. Of course they are not going to say this, that the probability is low, they say that the potential is there and therefore it is worth taking a chance. And then they'll give cases; 'well number theory, number theory never had any applications and now it is important in cryptography and you know cryptography is important because it keeps everybody's things secret' [laughs out loud]. Everything about me I'm sure is known despite cryptography; all my finances, all my love affairs, everything [laughs]... So [laughs]...

– It's a wonderful subject mathematics of course and the interesting thing is that it is coming into our lives more and more. The age is the mathematical age. Most of the mathematics is hidden, it's invisible to people because it's in programs, it's in chips, it's in laws... So you don't see it and if you don't see it you don't think it is there.

That is also one of the things we are hoping for in our thesis; to make the mathematics in the Mars shot more visible.

– More visible, that's right. So you'll do a thesis, but that is why I say that there is a potential in this thesis for articles, for publicity and making mathematics more visible.

When people in applied mathematics want more money what do they say?

– Well they look at the accomplishments. They say 'look, there is a mathematical aspect of tomography – tomography, you know, where you go into the brain, sections of the brain – there is a mathematical basis for that kind of thing, if it wasn't for that tomography wasn't around'. They are even talking about computer controlled aspect of complicated operations inside the body to aid the surgeons. Well everybody is interested in help, the Congress is interested in help, so they are putting more money into biomathematics or medical mathematics or so forth. And then maybe some mathematical theories for locating oil, some are good and some are better and so on, so we need more money to locate oil so we can all buy SUV colors [laughs] ... increase the consumption of oil... So they go for very specific accomplishments and predictions about accomplishments in very specific areas. There's an area in mathematical pharmacology – the production of drugs – and they talk now of the possibility of what they call designer drugs. And they say that there is a mathematical basis – not very good – for designer drugs and you can make a plea that you ought to give the pharmaceutical companies a lot of money for experimentation and for development and so on. I have some students that are in that field, working for drug companies. So I think that the problems for getting money for pure math and applied math are very different. And then way down on the list is getting money for something like the history of mathematics. They are going to say 'who needs this?' The answer is that we need it because we need more publicity about the way that mathematics affects our lives.

We have been reading something by Bruno Latour who is doing what he calls 'technoscience'. He says that if you as a researcher can't fit your research under either the warfare program or the health program you'll have huge difficulties in getting your funding.

– I would say that those two programs have the most money and give out the most money. But it's not all that there is, there are other things also, for example studies of various social arrangements, economics, there are money given to economics. Take the problem of traffic on the roads. Well the flow of traffic is kind of a fluid problem... particularly in California you'll see the fluid [laughs]...

– I have to tell you in advance since you are going to rent a car in the LA area that commutation times, which are like 7.30 to 9.00 in the morning and in the evening perhaps 4.30 to 6.00 when people are going to work and back, that traffic on the freeways is enormous and very slow. And the people out there somehow is used to it, they don't think of it, they don't think 'we ought to do something'.

I noticed something out there, that we don't have here in the east where the traffic is much less severe. There's a freeway and it may be 6 lanes this way and 6 lanes this way or maybe even more and there are entrances to this. Now there is a light, a stop light in front of the entrance and it stops people from going into the freeway immediately and it waits until a certain phenomenon happens. There must be a chip in there that is executing some sort of an algorithm for when you allow the next two cars to go in. I've seen this because I was driven around and I said: 'My God, you got this thing here, it must be severe, we don't have this'. The entrances and the exits are all okay, sometimes there is a pile up but it doesn't often happen. Well I don't know what places would have a project in that, but traffic control would be part of applied mathematics. Or they are even thinking of putting chips in automobile with censors to regulate the speed and distance and so on. I imagine that a certain amount of government money has gone into that kind of thing.

– So what Bruno Latour is saying is essentially correct; military and health, those are the big ones. But there are also many, many small ones. For example the weather predictions; hurricane predictions. That is a big thing because the hurricane down in the south and in Florida is severe. A number of people in this department are working with problems in turbulence. Maybe what they are doing... solving some of the partial differential equations of fluid flow maybe that will help the prediction of hurricanes. Maybe even tsunamis, I mean this tsunami thing is frightful out there in the Pacific. So these people are getting supported by federal funds. The funds usually comes in two forms; there are funds for the support of students and for the support of the principal investigator and that is usually what is called a sum of salary; two months of sum of salary. The students may get more and more support in terms of the fees and so on. And then there is support for equipment. That in applied mathematics is not so much, we don't have tremendous laboratories. We make use of maybe hooking into some computer, the state of the art computer maybe out somewhere else and we can hook into that. Yeah, he is essentially right but there is enough other things also to keep this department in business [laughs].

Another thing concerning the relationships between pure science, applied science, especially pure and applied mathematics, and technology. Is it correct to say that things begins in pure math, goes into applied and then technology or is there a feedback mechanism involved also?

– Yes there is a feedback mechanism, it goes both ways. The demand or the need often creates some application of mathematics that's already around and it often creates new mathematics although the basis for the new mathematics may be already there. The coding theory was already there before anyone used groups because it was rather simple and you could figure it out without groups. Then as things became more sophisticated somebody said: 'This is really an application of group theory.' So now we investigate what kind of groups we can use to produce codes which will have better features than the old ones. So it goes back, there is a lot of feedback in every direction. Then it is also true that the existence of a piece of technology creates demands. The most spectacular thing of this sort that I think I've have seen lately are the cell phones. Before we had cell phones people had telephones, the telegraph and we even had email and this seemed to be enough. Then the cell phone comes out: My brother! Everybody wants one, everybody wants to talk on the street to someone, talk while they are driving – we never talked while we were driving, we talked to someone right here, but never on a cell phone... The existence of the technology created a demand. But it can't be predicted what pieces of technology are going to create a demand and which ones are going to die. The possibility of a television telephone was around when I was a child but first of all the technology wasn't there and the demand wasn't there. Now you have it. Whether people like it I don't know but it's there. I know if I'm talking to someone I don't particular want to see what they look like; I know what they look like... There are a lot of people in the world that love gadgets, when a new gadget comes out they are the first to go a get it, use it and enjoy it... Well it's part of enjoyment. So I see many aspects of which you can fill all your pages. [...]

– We don't have anything like your projects in this department. We had something like it when I was a Ph.D. many, many years ago. This was at Harvard where I did my thesis. We had a big thesis of course which was original material but

we also had something called a minor thesis and the minor thesis was this: We went into the office of the chairman of the department and he had a hat with little bits of slips in it and we fished in and pulled out a slip. It had a couple of words on it; 3, 4, 5 words on it and he said: 'Have you heard of this topic? Write on this topic, you have a month to do it. You don't have to be original, if you want to produce some new stuff that's fine, but you don't have to'. So I fished into the hat, pulled out a slip, it said: 'The 3 body problem'. I had a month to put something together and the amount of information was enormous so I had to narrow it down otherwise I'd never get through. It was a challenge and it was useful. And there were people that couldn't get through that and they had to do it several times. They failed in one month or they didn't do it right, then they would go back and forth... slip again, get another topic. [...]

– Did I tell you why I wanted to know whether the Vikings used any mathematics in making their beautiful boats?

Not other than you were doing an article on some American boat race.

– That's right. I never did the article. I still have it in mind to do. It's called The America's Cup and it's a boat race. It costs a lot of money to compete in this, you have to be a billionaire or a zillionaire to compete in this because the boats are made new every time and the costs of the boats are enormous. For many years The America's Cup was run off the ocean here at Rhode Island. The race was in a triangle, it was marked up here. America lost it's preeminence in this, it's possible that the millionaires in America were not so interested in this kind of competition and they wanted some other thing I don't know what... what they spent their money on. But the America's Cup boats were made in Bristol which is about 20 miles south of here. Then ten or fifteen years ago the Australians won the thing and then the race was run down there in Australia. And then interestingly a millionaire or zillionaire, I think partly Italian and partly Swiss got interested in it and the Swiss now hold the Cup. The Swiss [laughs] ... I don't know where they run their races [laughs] ... In Switzerland? [laughs]. It's a funny thing. But anyway I met some Italian and Swiss mathematicians who got interested in the modeling of America's Cup yachts and the millionaire or the billionaire gave this crew a lot of money to work through models and to help the improvements of the design. And I got to know the chief investigator, he was here for a conference and I said: 'This is marvelous, do you think that the information you feed back into your models running and so on... do you think that this helped in winning the race?' He was a little bit shaky on this but he said: 'Well we won the race didn't we' [laughs]. But you know it is more than just the design of the boat it is also the skill of the captain and the crew, what they do with the sails and so on. So it is a mixture of the boat design and so on. But the boat design is exceedingly complicated because it involves not only the hydrodynamics of the boat in the water but it involves also the oceanographic design of the particularly course and the metrological design of the winds out there. And also it is an elastic problem because the boat has a main mast and the mast is elastic, it yields like this and so on depending on the stresses going through the sail. If you do a certain kind of maneuver it may overstress the mast so that the mast breaks. So you have to avoid this... the race is lost if the mast breaks. So there is an elastic problem that combines with the wind velocity and so on. It is an extremely complex problem with mathematical problems. [...] So he and his students have been on to this for a number of years and he sent me some reprints and I wanted to write a popular article about this going back to old boats. Of course I saw the museum there in Roskilde and I saw the new ones they are building and they seemed so beautiful; the new ones they are putting together to kind of duplicate the old ones. I wondered whether the boat designers were using any kind of mathematics. Maybe it is elementary, just to lay out the pieces or something like that or if it is more sophisticated than that, more complicated than that. So that is why I wrote for some information.

– So you would say that this is applied mathematics, not for war, not for medicine but for sport and for amusement and for pleasure of the people that are involved in the thing. Whether this kind of modeling might have some spin-offs, well it might have. Spin-offs in aerodynamics or airplane design or even in boat design – we are still using boats. It might have some spin-offs but the immediate thing is to win a race, design a better boat than the next person.

It is somewhat related to the space missions with prestige and...

- Yes there would be a lot of prestige in this thing. [...] [Alfio] Quarteroni is the leader of this group. [...] The money for this came from the millionaire who gets the Cup. I asked him how much money he got to do this and it was a number of millions of dollars. That's just the simulation. The actual construction of the boat is another question. [...] There is a lot of programming involved in this. I wrote them because I wanted to find out to what extend they are using packages that are already there, commercial packages.
- That is something that you might even found out there at the JPL; to what extend are they using commercial packages and to what extend do they have to develop their own things. [...] It would be interesting to know the policy of computation out there; are they using personal computers are they using these huge things and so on, super computers. The manner or working! [...]

We've been meaning also to ask you about your views regarding pure mathematics vs. applied mathematics – some of the views that you pose forward in The Mathematical Experience.

- My views are somewhat sociological and psychological maybe. My background is mixed. My doctoral thesis was in pure mathematics but prior to that I had had experience at NASA which was applied mathematics. Shortly after I got my doctoral thesis I worked in Washington in the National Bureau of Standards, and again that was applied mathematics. Then in 1963 I came up to this job here in the department of applied mathematics so my teaching has been teaching courses in applied mathematics. I've done some research in applied mathematics, particular in numerical analysis. I've written a lot of papers in numerical methods. So my experience is in both. On the sociological and psychological level there seem to be a fair amount of hostility between the pure and applied mathematicians. The pure mathematicians like to think that they are at the top of the heap and 'the applied mathematicians are just people down there that are just taking the stuff that we, the pure people, develop and...' [laughs]. Then the hostility has another aspect, they say that the applied mathematicians are getting all their funding; the pure mathematicians are getting some funding but it is not like the funding of the applied mathematicians. I think that in terms of the progress of the science of mathematics it goes both ways: The applied problems suggest pure mathematics and pure mathematics is useful in the real world. Once you find it is useful in the real world it becomes applied mathematics. And as I said before there is an awful lot of pure mathematics that is never going to be applied. It's there; you can find it in books and so on... Pure mathematics seems to put stresses on theorems. If you do a thesis in pure mathematics you are expected to produce new theorems and the number of theorems that are produced is enormous. ... Most of them are useless in a sense. So the criterion of acceptance in pure mathematics is first of all theorems and proofs of theorems particularly. Pure mathematics puts a lot of stress on proofs. Applied mathematics doesn't care too much about proofs... yeah, okay, it puts stress on the question of whether the models you develop are faithful to what you observe in nature or in society. ... You take this thing here [the article about The America's Cup], I mean they are using the Navier-Stokes equations. The Navier-Stokes equations come into play here in a very important way; the Navier-Stokes equations in the fluid theory. And there are so many unsolved pure mathematical problems about the Navier-Stokes equations, you see. And I asked Quarteroni about that and he said: 'Well it's irrelevant to us, we just make our models and we try to understand the physical situation. We try to make good models, we run them on the computer and then we test them out. Sometimes we test them out in the laboratory and sometimes we test them out in the ocean'. It's a different frame of mind, different goals, different senses of accomplishment.

So would you say they are definitely two different things or...?

- No they feed into one another. This takes from here and this takes from there and so on. They are complementary in a sense

So there are things going from applied mathematics to pure mathematics?

- Oh, I would think so... for example all these things with cryptography. The number theory is there. In fact the first application of number theory was very elementary number theory. Number theory was around since Euler or maybe

even Fermat [laughs]. They hooked into this kind of thing. Fourier series... all kinds of things they hooked into. What I'm saying is that the number theorists developed more number theory to suit the demands in cryptography. So it's a two way street. At the personal level it can be: 'We are better!' 'No, we are better!' and so on. 'We do real things and you do imaginary things' [laughs]. ... There was a very good applied mathematician that used to work years ago for the Bell Telephone Laboratories: Hamming, he was the one with Hamming codes. I knew him. He was a very clever fellow and he had a considerable disgust for some parts of pure mathematics. He said: 'If I knew that the safety of an airplane depended upon the Lebesgue integral, I would never fly in it' [laughs]. What he meant by that was that the theories of measure and the Lebesgue integration and so on was perfectly irrelevant to the sort of problems he was interested in. ... His head was screwed on right too [laughs].

Is it true that he always wore tweed coats?

– Yes he was quite stylish, quite dapper we would say. Very nice man, very nice fellow... very sharp tongue; if what he saw was stupid, he would say... I wasn't the one to know, maybe I shouldn't say it, I might loose my job if I say it [laughs].

Where did you meet Hamming?

– Well he worked for the Bell Telephone Laboratories in the days when I was working for the National Bureau of Standards in Washington. We had something in common – I forget what it was – many years ago. I met him a number of times. I liked him. I think he liked my philosophy also.

He was on the Los Alamos project, I think, before he went to Bell's.

– He was? Well that doesn't surprise me. That project had all the brightest people in the world there. Ulam was there, Metropolis, von Neumann... quite a bunch of mathematical stars.

You didn't have anything to do with it?

– No nothing at all. My war thing had to do with NASA and with airplanes. ... My first printed paper was from NASA, I can show it to you. It's the first one! I was pretty young; 22 or 23. I didn't have a Ph.D. in mathematics, I just had the undergraduate courses in mathematics. Of course there were no digital computers in those days, they had slide rules and they had these electrical things that did multiplication and addition but they couldn't see ... the memory was very limited, they could keep one thing in memory [laughs]. They were very limited, extremely slow...

Did you have punch cards?

– No they didn't have punch cards. Punch cards were around but for business applications. They were not used yet for scientific applications. There were some analog devices that were around but we didn't use them, we used these desk calculators. Anyway there was a new flying boat that was being tested out, flight tested, and during the test flight they were doing this maneuver and that maneuver and so on and go through various maneuvers. And they did one maneuver and the rudder broke. So they interrogated the pilot – nobody was killed or anything because it was pretty much on the water or near the water – they interrogated him 'what did you do' and so on and he said 'well he flipped the rudder back and forth like this a couple of times'. So my boss said: 'Well you want to analyze this thing here because I think what the pilot was doing was oscillating this thing in synchronization with the master frequency of the elastic system' – because the airplane in flight is essentially an elastic system. [...] My boss said: 'I think that's the kind of thing you want to investigate.' 'How can I investigate it', I said. He said: 'There are three ways an airplane can fly; it can do this, that's called pitching, it can do this, that's called rolling and it can do this, that's called yawing'. He said: 'What you want to do, since the rudder controls mostly yawing, and the ailerons controls rolling and so on and the elevators controls pitching, there seems to be a problem with the rudder. So just take a second order linear differential equation with a forcing function on the right hand side and just work it out and see what happens when you do this.' And of course... yeah, it was a linear differential equation with a right hand side but the coefficients of the differential equations were parameters that came from wind tunnels. So there were wind tunnel tests involved in this thing

here and also the way that the rudder connects up with the aerodynamic forces also came from wind tunnel tests. So 'let's see what you get'. Well that was quite a trick [laughs] to solve on one of these old fashion electrical/mechanical adding machines... multiplying machines. Most of the machines could only add and subtract but there were some more complicated ones that could multiply and divide [laughs] ... and you had to use those. Anyway, the investigation showed that it is possible; the forces on the rudder would build up to the point where it might not be sustainable. That was my first paper. I'll show it to you... Somebody pulled it out and said: 'My God you wrote this' you know, it's even on the web. It is so elementary I am embarrassed, I mean in mathematical terms it's so elementary that I am embarrassed but still; it's applied mathematics. The funny thing is that a couple of years ago the same phenomenon happened, tragically, because a commercial airplane got into the wake of another airplane's jet and did the same business with the rudder in trying to escape somehow the turbulence. That was tragic. And I wonder whether the commercial pilots are informed about this business about oscillating the rudder. [...]

Did you ever get to meet Claude Shannon?

– Yes I met him once. After I got my degree I had some interviews for jobs and one of the places I was interviewed was at the Bell Labs. I had a fellow student who had taken a job there and he was enthusiastic about the work that was being done there at the time. I came down for an interview and I was being interviewed by two people; one of them was Claude Shannon. For some reason I decided against it. I decided to take the job in Washington. If I remember correctly I'm not even sure that they offered me the job because I think that I told them, after the interview, that I really didn't want to work there with this kind of material. They had a certain kind of material that I would be working with and so on. In those days this was the foremost mathematical laboratory in the country. But I said no, I'd rather go back to Washington... do other things.

What kind of work did you do at the Bureau of Standard in Washington?

– Well this was at the first generation of computers and in the first generation of computers we didn't really know of the best way of solving problems. We would have different ideas, different strategies to solve a problem... numerical... and we would try them out. I was working in elliptic partial differential equations for a while and trying out methods that involved functions of a complex variable. I spent a couple of years with this, programming it and so on and so on but it didn't work out. There were a lot of things that didn't work out so it was experimental in that sense. And then finally after a number of years went by people in different parts of the world... they were going to use finite difference methods or something much better to do it. So... there were a time even in elliptic partial differential equations they were talking about probabilistic methods, Monte Carlo methods for solving these things which are exceedingly slow, and people wanted to try it out. So in the first generation of computers there was a lot of just experimenting, what would work and what wouldn't work, what was too complicated, what was too time-consuming, too much memory space involved. So I got involved in that kind of work.

Memory space was given another consideration...

– In those days it was important, that's right. I have to tell you that the first computer they used had a 128... slots. 128 words or bits or bytes maybe and then the programming were very primitive. We programmed in what was called the 4-address system, I mean this is ancient history, the 4-address system. What was the 4-address system? All of the slots were numbered; 1 up to a 128. The 4-address system was like this: Take a number in cell 63, combine it with the number that you'll find in cell 82, combine it according to the instruction; add, subtract, multiply, divide, that you'll find in cell 10, put the answer in cell 106. And then 'goto' such and such and such, the next instruction. So this was very primitive...

When was this?

– Between 1952 and 1957 or something like that. But you know, they are still talking about how you solve linear systems of equations, the best way of solving linear systems of equations. Well what are the dimensions, maybe a million by a million and depending upon what the matrix is they want to use this idea or they want to use something else. Is the matrix symmetric, okay then you

use this. Is the matrix sparse, being mostly zeros, there are a few terms on the diagonal or off the diagonal then you use such and such. What is the condition of the matrix, if the condition of the matrix is too high then you want to do this thing and so on. I think this discussion is still going on. Even though you have now commercial packages that will solve linear systems of algebraic equations investigating the best way of doing it is still a matter of some research. ...

– Well we talked for almost two hours...

Figur 2.2 Til venstre: Bygning husene Division of Applied Mathematics, Brown University (den lyse pixel under hvælvingen er Uffes ansigt). Til højre: Bjørn ved samme bygning.

3 Interviews fra Massachusetts Institute of Technology

Følgende interviews er foretaget ved Massachusetts Institute of Technology (MIT) i Cambridge, Massachusetts den 7. til den 9. marts 2005.

Figur 3.1 Til venstre: Ray and Maria Stata Center på MIT. Til højre: Indgang til LIDS – Laboratory for Information and Decision Systems, hvor doktor Ralph Kötter havde kontor.

3.1 Doktor Ralph Kötter 7/3 2005

– [About LDPC codes]... and then the entire community took off into a different direction following convolutional codes and sequential decoding that was supposed to be the savior and make information theory practical. So why coding theory took off for roughly thirty years and completely used up the whole field, concatenated codes mainly ... I've asked him [Gallager] about it, because he never had any idea that they might be useful, he invented them to show theoretical properties about error correcting coding... Like I said sequential decoding was supposed to be the savoir, turned out it wasn't, things became to complex, you never really got close to the capacity you wanted. Still up till around 1993 everybody was sort of following the same path; concatenation, convolutional codes, a good decoder for the inner code, algebraic decoder for the outer code. There were good progress being made on the algebraic side of coding theory, but not really on the engineering side, that hardly changed. What happened next was the invention of turbo codes. What is interesting is that it is very close to Shannon's ideas. In some sense it's much closer to Shannon's idea than all that happened afterwards; all the algebraic coding theory, the statistical description of signals. So they could demonstrate enormous gains over the other schemes, very close to capacity. People at first didn't believe it... then everybody proved for themselves that it was true. Then it set off a fury of research into understanding what was really going on...

They are still not understood are they?

– No they are not understood. The interesting part is that coding theory had developed into a very clean subject, a mathematical subject, at least the theory

part of coding... packings in metric spaces, very clean, the dominating parameter was minimum distance, the questions being asked were extremely elegant, elegant solutions, which really is the thing, I believe, that seduced the entire community to stay on that path and never look anywhere else. With the turbo codes people sort of just tried; 'what could we do?' There was one guy in France, Batai, he had been on that path for a long time. He had been on the path of trying to define codes in the manual structure with a good decoding algorithm for years. And I think Berrou was following that... he was their spiritual advisor more or less. Like I said, people tried to prove for themselves that it was really true. And it set off a whole revolution, which now leaves algebraic coding theory somewhere in the... It's not really a mainstream engineering problem anymore.

Would you call turbo codes a major breakthrough in coding theory or…

– Oh definitely. It is a major breakthrough. Interestingly, like I said, it's still not understood how they work. It's funny... in the beginning of turbo codes there were a bunch of papers which had flashy titles like 'Elucidating the performance of turbo codes' or 'Turbo codes; why they work' or so [laughs]. So there still is no real mathematical explanation. Well, last year, or the last two years maybe, there have been some things popping up that comes close to a real understanding. Connections to convex optimization... convex optimization problem in programming... primary doing algorithms for LPs [Linear Programming problems]. Most people had already given up on ever understanding what is going on, what's really, really going on. At some point of time the community was sick and tired of this whole thing; that nobody could make progress. There was a paper which was a breakthrough in the analysis and which at least provided a very nice tool to understand how the codes would behave if they were very long. And I think to some extend the community declared victory; 'now we understand them', and it's true in that specific case, they are understood, but the finite case still isn't. But besides this paper there was not much until maybe three years ago, where this new and pretty hefty theory popped up with connections to optimization theory.

Do you think that there are any new developments in coding theory that are due to the use of coding theory in space missions?

– I don't think so. I mean space missions, they are a great test case because the channel is very clean and because of the whole set up of the AWGN [Additive White Gaussian Noise] channel. So this is the only case where this really makes sense. All the other applications of coding theory or most of the applications of coding theory are governed by the properties of the channel. If you don't achieve the performance you want then it is not really the codes problem...

So the space programs were taking advantage of what was already there and had been developed in other aspects of coding theory?

– The space programs didn't think different than other coding theorists. They also had the concatenated coding schemes consisting of an algebraic code and a convolutional code.

The fact that the space channel is closely related to the AWGN channel did that make it any easier for the coding theorists involved in space communication in the early days?

– I don't think so. If Gallager had invented his codes perhaps not so early in coding theory... I think the problem was what to do with scomputers. Gallager's codes had the same potential as turbo codes; he could have produced the same curves. I don't think he was guided by anything in space missions. He had coding schemes and they were coding schemes. Maybe if he had the computers to evaluate them... he could have produced the same curves and coding theory would have taken quite a different path, it would never have gone down that byroute as much as it did. I don't think that the space missions have been the driving factor here. And I doubt very much that the French guys had the space missions in mind when they invented the turbo codes.

Do you know what institutions have driven the development of coding theory?

– MIT, I think, had something to say...

What about the Jet Propulsion Lab?

– I think they are more in the applied science, their problems were their missions... Well, many places I guess. To be honest more things outside coding theory has been the driving difference...

Are you thinking about CDs or...

– I think that the real driving forces in coding theory has been the making of recording disc drives. CDs and modem technology was a huge one and far more important than space technology... a computer modem over the telephone line, so the capacity of that line is roughly 50 kb/s. That we can achieve the capacity of the telephone line is a technological masterpiece, even if it is only 50 kb/s. For the channel, you see, it is a major achievement... That has driven coding theory far more, I think.

Earlier you mentioned sphere packing conditions. Do you think there has been any 'back and forth' between the coding theory in applied mathematics and the coding theory in pure mathematics?

– Coding theory in pure mathematics?

Geometrical codes, algebraic codes; codes not very useful in practice and maybe not even binary.

– Yes that was part of what my thesis was about [laughs].

But wouldn't you say that sphere packing bounds and such were more pure mathematics?

– No I think, that's still applied mathematics.

But some of the packings are in 24 dimensions...

– It's still applied. 24 dimensions is just 24 dimensions; a vector of length 24. I would still call it very much applied.

What about perfect codes?

– For example the Golay code. The classification of simple groups. That is a real contribution to mathematics, it has nothing to do with engineering, nobody in engineering cared about that. The Golay code played a major role in that.

– So there is stuff going back and forth between pure and applied mathematics. The algebraic junction codes cause some curves... and the developments about counting points on curves. They are definitely in the contribution to pure math. That is not applied math.

What do you do within coding theory?

– Oh, I do a little bit of everything. I do whatever comes my way. I do algebraic coding theory and etc. etc.

Where is coding theory going now and where will it be in ten years?

– That's a very good question... I think it is scattering a little bit. There was a unifying goal in coding theory and now that this is achieved it is not entirely clear what is left. So the people who know coding theory they take their knowledge about this and apply it to different areas. That's the natural way things go. However as they apply it to different areas, some to fading channels, some to other channels... there is no centralizing anymore. So people spread out, they take knowledge of coding theory with them and try it in many different fields now. So it is not entirely clear what the future of coding theory will be – in those new fields or problems that will come up it will become much more diverse and distributed.

Is coding theory 'dying' a little bit?

– I don't think 'dying' is the right word at all. Some people go to biology with coding theory and try to apply coding theoretical understanding to biological problems. It is an up-and-coming field, a lot of people go into it and there are very good problems and many good applications for coding theory there. Other people focus much more on practical channels, some on fading channels, so there is a whole branch of coding theory that moves into this direction and really try to make progress in this area. If these two directions will still be talking to each other in twenty years, I don't know. I really don't know, maybe they are, and maybe they aren't. Without having achieved capacity on the AWGN channel I would have said yes, because that was a focusing problem that everybody could relate back to, but that focusing problem is somewhat gone now...

Do you think there is going to be any other breakthroughs like turbo codes for example or do you think that is it?

– Well, there was the old problem about achieving capacity on channels, the AWGN channel and the binary selective channel – the two old big ones. But all this is done. It is just simply done. The best codes provably achieve up to 0.00-something. For all practical purposes it's done. So there's not even room left for any breakthroughs, that is with respect to this old problem. I'm sure that coding theory can make breakthrough contributions to other areas in it's branching out process.

You don't think there is a chance of any new problems popping up?

– Centralizing problems? I don't know. It is not clear what it would be. Maybe there are some information theoretic questions which are being addressed right now. That has created some general interest among coding theorists. Something will happen though, but I don't know. The old coding theory focusing on achieving capacity on these two channels has served it's purpose and achieved its goal.

Professor Gallager doesn't do any coding theory anymore?

– No, he does networking. [...]

We went to the Technical University of Denmark to talk to...

– Jørn Justesen?

Yes.

– He is a real engineer [laughs]. You spoke to Tom Høholdt as well?

Yes. We also spoke to Søren Forchhammer. He is in data compression, you know him?

– Sure. Coding theory is nice in that way. You know everybody.

Justesen – and Forchhammer also for that matter – are involved in some of the ESA missions. Justesen said that some of the codes are implemented in hardware which will make it hard to begin using new codes – they are not using the turbo codes or anything.

– No but they will. Next time they will. Someone told me so.

But the fact that the codes in use have to be very reliable and well tested must mean something?

– Sure. You don't want to be the guy who has to explain why it doesn't work while things are flying in outer space. It is the same in all companies. Disc drives aren't going to be using turbo codes within a long period of time. For them [the CD companies] part of the reason is that nobody really knows what is going on [with turbo codes]. And as I said, nobody wants to be the guy who explains why the line of product doesn't work.

There are still Hamming codes in use in some computer hardware isn't there?

– Oh yeah. There are the most insane codes out there.

– I think, coding theory is to some extend overrated. I mean the gains we get out of the codes. Nowadays we fight over half a dB or so. You should have paid the antenna guy a little more, or the battery guy, then you easily could get a dB there. If you are a cell phone company like Motorola, you can achieve performance on many different fronts. You can make the antenna a little bit larger, you can make the battery a little bit larger, you can make the signal processing unit a little bit larger so you can afford more coding theory. And in the balance of all those different things it is not clear that coding theory is always the thing you want to invest in. So if a simple Hamming code gets you seventy percent of the gain, then for the last thirty percent you would have to work harder and harder and harder... Okay, outer space is special since every kilogram you send out there is very expensive. In comparison you can set up a lot of hardware here, you can fill up rooms and rooms of hardware here which still makes it cheaper to... the end coding is never the problem, the decoding is the problem, but the decoder down here can be large an expensive for a space mission, which is very special. But if you talk cell phones, then every cell phone needs a decoder.

What codes are in use in cell phones?

– Also convolutional codes and the new standard on turbo codes. And again there are also a lot of problems there... they probably got bigger problems than coding theory! You can easily loose a lot of dBs... so from an engineering point of view it is a very interesting optimization problem where you want to put your resources. And it is not clear that it should be within coding theory with more powerful and more complicated codes.

Regarding cell phones, how much coding would you put in hardware and how much in software?

– As much in software as possible. You can change it easily.

We understand, that in the space programs they usually have big hardware decoders on ground, another reason why they can't easily change their coding standards.

– Yes, they have BIG decoders. They do a 14th state convolutional code. You would never be able to do anything like that in other applications.

Do you know what they are using exactly? We heard they are using the big convolutional code and not the small one?

– Oh, they screwed up! There was some problem... they didn't get enough energy or enough power from some of the solar panels, I think.

From where do you know that?

– From where do I know that? It seems that everybody knows that. They screwed up; they actually changed the code in flight. They first had a low memory convolutional code, but that code didn't give them the gains they needed, so they changed the code in flight and made up for it.

So they screwed up in the way that they couldn't make it work or...

– Yes sort of, at least that's the story that I know.

– Sorry guys, it's four o'clock, I have to go now.

Figur 3.2 Til venstre: Bygning hvor LIDS har til huse. Til højre: Kort over MIT.

3.2 Professor Gilbert Strang 9/3 2005

... The thesis has an overall frame that has to do with the stagnation of science and we hope you have some views on that concerning data compression. When we say stagnation we mean that not much has really happened in our lifetime. In our parents and grandparents lifetimes things like cars, aeroplanes and computers have been invented, but what is new in our lifetime - except perhaps the remote control? So this is the technological side. We are trying to find out if this trend applies to science as well by looking at the Mars shot an asking the question: 'Is there anything new in the science of the Mars shot or is it just old stuff that is being twisted a bit?' We have talked to Ralph Kötter about the same and when it comes to coding theory he sort of agreed with the idea of stagnation. The problems within coding theory have more or less been solved.

– Well, I suppose that any Biologist would say 'our science is the one moving fast now'. I have to tell you that my personal feeling is that NASA gets way too much money – for what? You are probably excited about going to the moon and stuff like that. For me that is a little bit old. I don't think the US government should be... I mean NASA has a budget three times the National Science Foundation and for me that's not reasonable. I think NASA is one of those things that when it started was exciting, but for me the excitement is not there. I unwisely said this at a faculty table with aeronautic engineers around it that didn't agree. I guess I'm just expressing and ordinary persons opinion. I wouldn't be surprised if some of their problems are solved so to speak. It becomes engineering. Of course something like GPS comes a long and by a whole new method get orders of magnitudes better accuracy on the positioning, so news things do happen. Now for image processing, that's really your question...

Do you think that the new features in data compression are in personal computers or in the Space Program?

– I'll make any comment, but again remember that you are talking to a mathematician guy, and we don't have to worry about these things, I mean we have our own problems, but these are problems we create, so to speak. When I'm acting as an applied mathematician I have to work with a problem that the world creates. So what can I say about that one problem of data compression... people are still thinking hard about that. Whether the result will be an order of magnitude improvement in data compression I think is to early to say. We have basic ideas already in data compression. My overall picture of that process is that you start with a signal and somehow you have to represent it as some combination of functions like sines and cosines, recently as wavelets and other new ideas. Those are chosen, and the coefficients in the combinations are what you save. Of course you don't save everything because we want to do compression, so we better do compression between the representation and the reconstruction. So we start with the real signal, we represent it as a combination of our good functions, we look at the coefficients and the small ones we throw away or we use only a smaller number of bits and we reconstruct with the new coefficients. I would say, from what I know, that wavelets have brought some new ideas to this technology. So the JPEG standard for this got changed a few years ago from using sines and cosines to using wavelets. So there is change. It may not be the order of magnitude changes we are seeing in biology where we understand things we never had the slightest ideas about before, or GPS was my example of technology where we get orders of magnitudes better answers. So 'stagnation' is too strong, 'slowing down' would be a better word, it happens after all. I think it happens in big labs too, without knowing JPL at all, I don't know if I've been inside the door. I think it's true about big labs with a special effort like 'get to the moon'. They get good people, they do it, but then ten or twenty years later it's hard to do something so great every year and that may be what you are saying. But each effort probably peaks [draws on blackboard] as it solves a problem and then the excitement goes, the rate of invention drops, the quality of the science flattens out. I think you could have an interesting thesis if you could make any of that quantitated. Because it's probably true. So biology or brain science is on the fast curve upwards, but the Mars shot, I don't know where to place it.

But would you say that the development in mathematics is slower than in biology? Off course things are happening in biology, but the breakthrough was the discovery of the double helix, wasn't it?

– Right, but using that in coding the genome, and knowing all the genes is a fantastic event, which MIT was actually highly involved with. You remember the gene was decoded by two independent groups, there was a public group, in which MIT and probably CalTech had a place, and then there was this private company, and that was a big deal. They started with being able to decode a couple of genes a day and they got to do tens of thousands so that was clearly on the way up. Now we know what the genes are, it's like knowing what all the pieces of a car are, but how it works we don't know, so that's extremely exciting. And math has a place in this luckily. Maybe not such an obvious place as in the old days with physics. Mathematics was clearly connected to physics and engineering through partial differential equations and so on. Math had a natural application there. Now these are new applications, they are more discrete, not differential

equations so much as combinatorial and statistical. So new mathematicians are in business, new mathematics is created. We are probably a little behind the biologists who are in the lab making the incredible stuff happen. So I think my thought on stagnation is really that sort of curve, that there are some subjects that have stagnated... So there is science and then there is applied science. The Mars shot I take to be applied science. It's not Nobel prize stuff, it's getting it exactly right, it's not trivial but it's a different job and maybe some parts of it would flatten out. Data compression got a big boost of excitement when wavelets brought in a new sort of possibility that's not perfect at all, but it opened a new door.

Would you call it a breakthrough in data compression?

– Conceptually yeah, it really provided a new class of functions. Everybody was using sines and cosines, couldn't figure anything else out, and here wavelets offered a new class of functions and fast algorithms. But in many problems sines and cosines are just terrific. So it's finding new applications... Usually sines and cosines are great for smooth functions and signals, but wavelets are better for signals with impulses, jumps and quick changes, transients.

Looking at the drawing you just made – does the peak have to do with applications, or...?

– Well, I suppose it depends on the problem. The peak in pure science probably comes before the peak in applied science.

Is wavelets a peak in data compression or is on the way down on the slope?

– I wish I were more applied to know the answer to that, but from my theoretical point of view I would say wavelets would be a peak. Cause they really opened a new door. People were creating new wavelets, the new functions were coming, it was very exciting. My own earlier experiences were with the Finite Element Method that structural engineers did, and that came twenty years before. So it was similar. There was a very exciting time when the method was unfolding and now it's a very powerful standard method with millions of dollars worth of code. It's still under development but the golden age is over.

Wavelets in data compression was in the eighties?

– Yeah, as often happens the engineers who had to do the actual compression had the ideas before mathematicians ever heard of it. It came in the seventies with something called filter banks. The idea of a filter goes back quite a bit. The filter picks out the frequency band that you want. People understood those. Then they put several filters together, and could reconstruct from different bands, that was the key idea. Then wavelets appeared and we understood that they were closely connected to filter banks. So that's a little history. And what would typically happen, I'm sorry to say this on the tape. There is a time when the mathematicians and the applied guys are excited about helping each other, and then it gets understood by the mathematicians and they separate. The applied guy still has to build this thing, faster and better. The mathematician says: 'Okay, I understood these ideas, how about these further fine points?' And the engineer says: 'I don't care about these fine points, I just want a thing that is cheap and works.' That happened with the Finite Element in my opinion. At a certain point mathematicians and engineers were connected and then the two groups separate. But this is just one persons comment. Once the mathematics gets too refined I tend to look for a new problem. So when wavelets came I thought: 'Oh, that's interesting'. So instead of talking to structural engineers I started talking to electrical engineers.

Where do you think the new impulses come from? Do they come from space missions or from private applications?

– I'm sure space has contributed. It's a good question: 'What were the exciting theoretical issues in the space program?' But I really don't know because I wasn't involved.

Yesterday when we talked to Ralph Koetter he said that coding theory didn't get much from space missions. He thought it got more from CD players and so on. Is it the same with data compression?

– I guess I didn't see the input from space missions to what I know about data compression.

It seems that the data compression schemes used in the current mission is really just a stripped down version of JPEG.

– Well, everything has its constraints. High definition television was 'the mission' for data compression. How do you get high definition TV when you have more bits than you can transmit to the receiver in the available time? In ordinary TV you transmit everything, but high definition has got to many pixels, to much information, and you have to compress. So that was a big question and a big political battle. The TV and the movie and the personal computer, people all had different systems that would cost billions to change. The space missions' constraint must have been small size and weight. And that would have presented it's own problems.

Where do you think data compression is going, and where will it be in ten years?

– If you're going to the JPL those are really the guys to ask, I don't know. It will be driven by the big problems and those guys will know. And then I'm sure Sony will know. I've several times been part of patent suits, someone says 'I've invented that, and Sony has used it' or 'Microsoft has used it'. You never know what kind of settlements they reach, they just tell you 'Okay, it's been settled.' But that's in Sony's MP3s and their JPEG, whatever. It's in their every day electronics.

So it's more or less in the hands of the commercial people?

– Yeah, that's what I think. I think it was remarkable that wavelets made it into JPEG standards because a change of standards is such a big deal that it really has to be better. I'm sure that's true also with NASA, I'm sure they have codes in which they've invested thousands of man-years. If someone comes along with a better idea, it has to be ten times better.

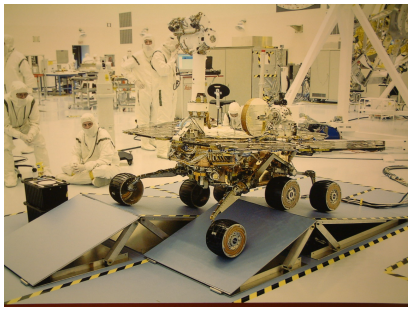
One last question. There is another aspect of the thesis having to do with the mathematics of the Mars shot being hidden from the common people. When you read about the shot in the papers it doesn't say: 'This couldn't be done without mathematics'

– Right, but because I'm one of the common people when it comes to the Mars Mission, I couldn't say what was used that was essential. But I bet there was essential math. Probably solving ordinary differential equations with great accuracy, prediction, estimation. Kalman Filters would probably enter. So there is a whole world there. That would be a good question to ask: 'What was the math?'

Thank you very much.

4 Interviews fra Jet Propulsion Laboratory

Følgende interviews er foretaget ved Jet Propulsion Laboratory (JPL) i Pasadena, Californien fra den 14. til den 17. marts 2005.



Figur 4.1 Vores fotografier af JPLs egne ophængte billeder. Til venstre: Test af MER-rover. Til højre: MER-airbags.

4.1 Doktor Aaron Kiely 14/3 2005

[...] *We have chosen data compression and coding theory as our cases, and then go deeper into that. That's one thing. Then we have a sort of overall frame on the thesis that has to do with the stagnation of science. For example in physics you could argue that there haven't been any great breakthroughs since the discovery of quantum theory and the theory of relativity, the same in biology with the discovery of the double helix. We are trying to see how it is in mathematics.*

- I think in coding theory... I think you could argue there has been a breakthrough somewhat recently, ten or so years with turbo codes. For now people are talking about Low Density Parity Check codes. It's not a breakthrough in the sense of... we have known what the Shannon limits are since the 40's so this isn't sort of a breakthrough in that sense. We have known how to get as close as we want to the Shannon limits but not with reasonable complexity so the breakthrough is that we can get an awful lot closer to the Shannon limit with complexity starting to be reasonable and for space applications where we have a pretty unique situation and where what we care about is the encoding and the complexity being low... both coding and data compression is sort of unique in that... On the decoding end on the ground we can afford to have something very complicated, the exact opposite of your typical Internet application because you have a compressed image and if it's a little bit complicated to produce that compressed image that is not a big deal. The big deal is being able to decode it simply using software, so it's the opposite of what we have in space.

The reason we are here at JPL is that we are trying to see if science is stagnating, we thought that we could come to a place like this where you'd probably expect to find 'state of the art' science. We were wondering: What's new in the current Mars shot compared to the Pathfinder for instance, when you are talking data compression?

– I'll show you. [Aaron Kiely begins to show a Power Point slide show on his computer.] What Matt Klimesh and I worked on is the image compression aspects, there are other data types on the rovers, but the imaging is what takes up most of the bits, from our perspective the important part. So here is a comparison. Do you know what a reconstruction curve is? Actually we should just start from the beginning. All the cameras that are on the rovers except the microscopic camera are steering cameras. This is more cameras than they have had on any rovers on Mars. So here is an example. They are all 1024 by 1024 pixels, so basically a mega pixel and each one is twelve bits per pixel and they are all grey scaled expect for the pancam, but even the pancam really... each individual image that comes down is a grey scale image, it just has a filter wheel that it can rotate through. So in addition to data compression they can reduce the data volume through pixelar averaging, for example by looking at just a window of the picture or something like that.

– So this is a nice summery, it is a little bit outdated, it is more than a year old but I think the fraction is probably pretty similar. This is the fraction of image data that you get from different cameras and so in terms of the numbers of mega pixels you get more back from the pancam than anything else by a large fraction if you express in terms of compressed data volume. The interesting thing here is that you can see the guys running the pancams are willing to tolerate a little more compression. They are using higher quality compression for the microscopic images; they are not tolerating so much data compression for that one.

– One of the things they do for every image is that they produce a thumbnail image The regular image is 1024 by 1024 and then they do pixelar averaging to produce at 64 by 64 thumbnail and all the thumbnails get sent back. An interesting thing is that on average a thumbnail image gets compressed to about 650 bytes using the ICER compressor and the header is actually 686 bytes, so on average the header is a little bit bigger than the image.

– We have got two image compressors on MER. There is the ICER compressor that Matt Klimesh and I produced, which has a lot of similarities to JPEG-2000. It is a progressive algorithm which means that as you get more and more compressed bits you get better and better overall image quality as you are transmitting this thing. It uses an invertible integer wavelet transform so that you can use it for lossless compression if you transmit all of the compressed bits and have your picture reconstructed perfectly. The codes are pretty similar to JPEG-2000. There also is the LOCO compressor which was developed by some guys at Hewlett Packard. Matt and I produced a modified version of the software that they are flying, it has a little bit lower complexity. LOCO is... the motivation factor there is that it is very fast. It only does lossless compression and if lossless compression is what you want it does it very fast and faster than ICER does. It is almost as good as ICER in terms of lossless compression. Compression speed is very important.

So you use LOCO for particular kinds of data or...?

– The scientists specify whether they want lossless compression through LOCO, whether they want ICER and if they want ICER then what bit rate and quality level they are going to require.

How about control data?

– ICER and LOCO are both image compressors. They are really specialized. They take advantage of the fact that there is a certain structure, a two-dimensional structure, in the image data, so if it were one-dimensional science data these compressions would be completely inappropriate for that.

– So here are the two compressors. This is the fraction of images that come down or image area. So you can see that basically the vast majority of it is lossy compression through ICER. [...] An interesting thing to notice is, that for a regular ICER image the average bit rate is about 1.1 or so bit/pixel. The most common request for an ICER image is an image compressed to about one bit/pixel. This is not a very direct indication of ICER's capabilities because it depends on the scientists using it, some other scientist may say 'well we really want better quality' and they may ask for 2 bits/pixel or 2.5 bit/pixel and for some applications if they have got... say a multi spectral scene with several bands in the scene

they may want one of those bands that are high bit rate and other bands that are only a quarter bit/pixel for example. This doesn't necessarily show how well ICER is doing, it reflects also what the scientists think what they can tolerate.

– This is a very oversimplified schematic of how data compression works in general. We got data coming in, an image or whatever it is, and then some sort of preprocessing state for example a wavelet transform plus quantization or a discrete cosine transform or something. The goal is that it doesn't perform any compression and in fact it is often a lossy process, it might throw out some of the data but the idea is to process the data in a way that makes it more receptive to compression through the entropy encoder. The entropy encoder is sort of the engine. Given some sort of probabilistic model of the source it compresses data or represents it in a more efficient way through something like a variable length code. That is sort of the big picture of what is going on. So for example for ICER what is going on is mostly a probabilistic transform. For LOCO it is in essential trying to project a probability distribution on the next pixel that it is about to encode based on what it has seen in the nearby neighbors.

Is it the same transform that's in JPEG-2000?

– JPEG-2000 is sort of open to... I think you can implement a variety of wavelet transforms that are compliant with JPEG-2000. We are using the same wavelet transforms that are in the XVID compression algorithm. There is a good technical paper on the ICER. [...]

Are you mostly working with producing software or mathematics?

– I'm really doing both. I don't know if it is always that easy to draw a clear distinction between if what is being done in one case is software but not mathematics. We spent a while in algorithm development coming up with an entropy encoder that is fairly efficient for the coding that is done in ICER which I think is a lot closer to pure mathematics but we also spent quite a while making ICER run faster... where we spent a lot of time to restrain the code where certainly you have to have some understanding of mathematics. Deep mathematics like quantum physics you have to have a certain understanding, my guess is, to apply it. In the experimental sorts of things you have an idea for a change in the software that would help make things faster and for a lot of times basically you don't really know how much of a difference it is going to make or if it is going to make any difference until you try it out. So in that sense it is sort of experimental.

– This is... are you guys familiar with entropy encoding? Here's an example. Suppose you had an image that had say only eight different shades of gray in the pixels. Let's say they were indexed from 0 through 7 and in one case lets suppose that all those pixels were equally likely. In that case the best you can do...

Oh yeah, it is just like Huffman codes for example.

– Yeah. For something like that the best you can do is just to assign a 3 bit codeword to each of the symbols and it is going to cost you three bits to encode it. Alternatively if you had something where they are much more skewed with a probability distribution where the 0 pixel is much more likely than say the pixel number 7, so if you used...

... a shorter code on that.

– Yeah, you want to use shorter code words for your very probable symbols. So that sort of trade off is sort of the idea on entropy encoding. Similarly you could think that maybe if you just look at each pixel individually maybe they are all equally likely but if instead you look at the difference between this [specific] pixel and its neighbor maybe now you know that it is very likely that they are equal so it is unlikely that they are very different. In case of LOCO or something like that you would have something much more sophisticated where, loosely speaking, if it thinks that now you are near a vertical edge in the picture then it uses a different model to predict the probability to use on the next pixel. So this is what we are doing, we are trying to predict a probability distribution on the next pixel that we are going to encode based on what we have seen nearby. We like the probabilities for each in ideal to be very sharply peaked so that we have a lot of confidence in what our guess is if that is possible and we want our guess to be right. If we have a good model of what that probability distribution looks like then we can do a good job with coding. So that's sort of the idea behind entropy coding and so doing something like a wavelet transform in advance can

help you spend more of your compressed bits on the more important information in the image.

- This is sort of the same thing I was talking about illustrating the idea of preprocessing. If they were all equally likely would cost you 3 bits, but if instead you know something about the probability distribution between a pixel and its neighbor you might get something that is much more favorable.
- This is sort of what I was getting at with LOCO. In the case with LOCO if x is the pixel that is about to be encoded these are the nearby neighbors that you look at to decide what your context model is... how you are going to predict the probability distribution on x.
- This is ICER now. This is an indication of what happens with progressive image compression... because of a bug in Microsoft Power Point this isn't displaying on my screen correctly. Okay here. This is what is meant by progressive compression. The idea is that as you transmit more and more data in the image and the overall quality gets better and better. So the one close up you can see here is 0.18 bits/pixel... this is a fairly sort of rough image but as you move up to about 1 or 2 bits/pixel you can hardly tell the difference between that and 4.7 which is lossless compression for this particular image. So that's what is meant by progressive compression. Actually I have a film that will show you the difference. This is one of the differences between Pathfinder and MER, that ICER is a progressive algorithm where Pathfinder was a sequential algorithm.
- This is the idea behind wavelet transforms. So here is an example of an original image. Think in a way of the transform as sort of a high pass - low pass filter pair acting spatially. First thing we do is to apply this filter pair to each row of data in the image, so the output is this low pass version of the image and this high pass version of the image here. Then we can do the same thing vertically here and somehow this little region here is sort of a low pass in the horizontal vertical direction and so on and this region here is high pass in the horizontal vertical direction. So if you looked at the histogram again some of the ideas are that we liked to have these histograms nice and sharply peaked for entropy encoding reasons. [...] So now we have gotten this gain that we have this nice histogram but still in addition to that we can apply context modeling. That is try to not only exploit the fact that the shape of this histogram is nicely peaked but try to predict what the next pixel is going to be in one of these subbands given the values of its nearby neighbors. And I want to point out that this stuff that I am talking about, we didn't invent this at the JPL, this is what people have been doing... wavelet based image compression for the last couple of decades.

Is this presentation available on the Internet?

- No I'm afraid not.

Is it possible that we get a copy of it?

- I'm afraid I can't. I haven't gotten cleared to give you any copy.
- So we have this advantage that for example this subband here already looks like sort of a low resolution, so we want to give it higher priority, we want to make sure that this piece of information is reproduced at a higher fidelity than say these subbands here. So the idea is that in general we are going to do this starting from the most significant bitplane working down to the least significant bitplane. So the idea is that at each state we want to pick the compressed bit layer from the subband that gives us the biggest improvement in image quality per compressed bit that we are going to transmit. So we tried sort of ad hoc... We can't exhaustibly compute this figuring out which one will be very confidentially intensive so we have some heuristic methods for deciding what's the next piece of information to transmit. And so again as I said we are also taking advantage of looking at the local context.
- Let me explain what this graph is. If we know that we have got a source where the probability of a 0 coming out is say 0.7 then we know from Shannon theory how well we can do to encode that. So this graph here shows redundancy of added cost, that's how far we are above the Shannon limit if we exactly knew the probability of 0 from the source. So the point is that on this vertical axis the numbers are very, very small so we are actually quite close there. We got something that's very good if we have a really good estimate of the probability of 0. So this is a technique that is called interleaved entropy coding that is being

used by ICER. Basically what it does is it breaks it down into seventeen little bands, so I don't know what these numbers are but let's say they are from 0.6 to 0.67, so that the probability of 0 within that range uses a particularly mode of the coder and once it crosses that threshold it uses another mode, it has got seventeen different regions that are being used by the encoder.

– This is a comparison. Arithmetic coding is another technique that is extremely effective for encoding. Entropy encoders are a little bit faster than arithmetic coding which is one of the reasons why they appealed to us.

– This is... For lossy compression the goal is to represent the image with as high a quality as possible and using as few bits as possible. So we can plot a curve that sort of measures the tradeoff between these two things. We call it a reconstruction curve. So here is an example, the horizontal axis is rate, the average number of bits per pixel. So you remember from that earlier graph we saw that for MER on average the scientists were using about 1 bit/pixel for many of the images and so you can imagine that if you are willing to spent more bits you can get an even better quality. There are two common ways that you can measure the distortion; mean square error, which is the average of square difference between the original pixel and the reconstructed pixel with coordinates there, so it is the average of that. The peak signal noise ratio is essentially one divided by mean square error but measured in dB, so it is confusing because if your mean square error is low then your peak signal noise ratio is high, so you want low mean square error and you want high peak signal noise ratio so it gets confusing. But this is what you typically would see. At low bit rates you get a very, very big increase in your image quality for a few bits. And you saw actually in that little movie that very quickly we got this okay-looking image there and then as we moved on... so we got this very steep drop down this curve as the image quality improved very quickly as we added just a small number of bits and then we reached a point where it was hard to tell if we were making any difference and that's exactly what happens, things tend to flatten out here and it gets harder to predict these bits and to do an effective job of coding. This shape is very typical for a rate distortion curve. So an interesting thing is that we don't really know what this curve is going to look like for a particular image and for a particular algorithm. And here is an example. These are three actual images from MER, they have very, very different tradeoffs. This one here has a lot of sand and there is a lot of details here, it is much, much harder to compress than say this one here that is a sun-finding image where it is almost all sky. It's actually not perfectly black, there is a little bit of noise in it but it is very easy to compress. It's very easy to predict that this will be very close to black and here is another image that is in the middle, in between the two. It's kind of interesting because if you're a scientist and you say 'well I know that I can accept this much distortion', you know, a tenth to two, well how do you decide what rate to climb for? You could have this blue image which needs about 0.5 bit/pixel or you could have this red image where you need more than 3 bits/pixel so there is a tremendous variation which makes it kind of inconvenient. Similarly if you said: 'Well I'm going to need 2 bits/pixel', you don't know what sort of quality you are going to get, you might be stuck with this red image where your quality is much, much lower than for the green image. If you have the green image you really like to compress it using a small number of bits and then spend those extra bits on something like the red image which is much harder to compress. This is a sun finding image here where it is a 651 compression because all we are really trying to do is to locate the center of the sun there. So on ICER we had two ways that you can control it, two parameters that make it convenient; we have a quality level where you basically just pick on this axis and then we have a byte coder which control the weight on this axis. So you specify both of those and ICER will try to produce a compressed image that meets your quality level using as few bytes as possible and it stops once one of those limits are met. So here is graphically what is going on. So you might specify a quality level which is along here and you might specify a maximum weight and so what it says is that, as you're compressing the image once you are able to meet this quality level then that is good for my scientific purpose and you don't need to spend any more bits. You can move along to the other image. On the other hand, I can't afford any more than say 4 bits/pixel so once it meets 4 bits/pixel it is time to stop and move on to the other image even if you haven't met the quality on it. So

that provides some flexibility in being able to trade the rate and distortion here. And because ICER is progressive, you saw in the movie that the quality would improve, so basically what happens is that it would continue to improve the quality until it met the quality target and once it recognizes that 'okay I've met my quality target' it would stop. So also interestingly it is faster if you have a lower bit rate.

So you wouldn't be able to do that with the JPEG?

- No. That's a significant advantage that a progressive algorithm has over say JPEG, at least the way JPEG was done on Mars Pathfinder. It was not progressive so instead in JPEG you have essentially just this quality level. If you have ever worked with JPEG software that allows you to control it, a lot of times they will say 'high, medium, low' and sometimes you'll be able to control a number that varies between 0 and 100 and basically you set that number to say 75 and you don't know how many bytes it is going to be and JPEG doesn't either until it has actually compressed it and says 'okay this is how many bytes it costs'. So on Pathfinder if they wanted some byte quota the software would run JPEG with a guess of what the quality level should be and if it was above or below it in terms of byte quota it would revise its guess and do it again so they had an iterative algorithm if they wanted to control it to a quality level. On ICER it is automatic, it just stops compressing.
- So one issue that we have on the deep space channel is we make errors or more commonly we loose packets. You are somewhat familiar with the channel code situation where you have an uncorrectable error pattern and in that case they declare the frame or the packet to be lost and so you're stuck without that. If we are doing a good job of compression we are trying to learn about the image we're compressing as we are compressing it, so we'll have these probability models that we are learning. We know that the way that we compress pixels of the image here depends somewhat on what we've learned of the pixels up here. So by learning and building these adaptive probability-models we can do a better job of compression. The drawback is that if you lose this portion of the image up here you don't know how to decompress the image. So when you have a packet loss you're stuck. For the Internet that's not a big deal because it gets retransmitted so quickly. We have retransmission from MER but it is much slower, it can be sometime the next day. That is pretty common. So you're stuck without your data for quite a while. So it is important to put some limits on the extend to which the way you decompress this pixel depends on the way that this pixel was compressed and so ICER had this sort of build in.
- Here's an example where we've got an image that has been divided into twenty-three segments just hypothetically and you can see that we get some advantages. The partitioning is actually done in the wavelet transform domain not in the spatial domain so these times are sort of approximate, but here is a case where in this segment for example we have lost fifty percent of the data but because it is progressive you can hardly tell that you've lost anything. Up here where you have lost ninety percent of the data you can actually see some ringing artifacts and you got a lot of distortion there but still you can make out the edge of the mountain and you have some idea of what is going on. Here, where we have lost a hundred percent, of course you don't get anything back. But this is one of the advantages of progressive transmission; if we had some of that stuff early in the packet we still get something reasonable like this. And so ICER has in it an algorithm where you just tell it how many segments you want the image to be partitioned into and it automatically partitions the image. The idea is that if you thought that your channel would be very unreliable you would use a very large number of segments and if thought your channel would be fairly reliable you would use a smaller number of segments to trade off to account for this.
- This is sort of a better illustration of what I was talking about, the partitioning is done in the wavelet transform domain. We saw this sort of wavelet transform phenomenon and so here for example everything that is shaded red here is part of the same segment here and so in the image domain the boundary is actually kind of fuzzy because of the wavelet transform. And this is a graphical description of ICER's algorithm for dividing an image. As I said, we are doing the partitioning in the wavelet transform domain. This is not the same as taking your image and just chopping it up into a whole bunch of little images and then compressing

each of those individually, the compression effect of this would not be so good, but the other thing is that if we did that you would see these boarders, as you zoomed in, you would see these boarders between these images even when no losses occur at all, okay? When we do the partitioning in the transform domain, if we have no losses you can't tell that the partitioning has been done at all. So much of this is nice but the question is 'did it come up' and the answer is 'yes'. This is a screen shot from the actual web side that went up and you can see all these black holes where they lost data and these things and so they got these back eventually because they were retransmitted but at least for a while they were stuck with these holes in the image.

– Here is an example. This is an image from a front hazard avoidance camera which they use in part for deciding where the instrument deployment arm is going to be able to reach. So here is a rock for example that they are interested in. This is a front hazard avoidance camera that they have unwarped so to speak and then what they do is they compute the stereo correlation algorithms to map out the terrain in front of them and then they try to decide whether the arm can reach an area or not. This color-coded map shows which regions they think they will be able to reach here. So this is important because it is fairly common that they will drive all day and at the end of the day they will transmit these hazard avoidance images to see where they are before they put the rover to sleep. And based on where they are and what they think they can reach or they can't reach that has a big impact on what they are going to do the next day. So it is very important to get these images back to be able to do these correlations to map things out. And so I have heard from Justin Maki, an imaging scientist on MER that they have had cases where they got just missing segments here and here but they've got enough segments right in the middle so they are still able to do this reach ability map so they are still able to plan what to do for the next day. If they didn't have the area containment and they lost a piece here they could lose the whole image so this is a case where it has really paid off and has saved them potentially a day of operations.

– So you were asking to which degree math versus software. This is an example of something that is closer to the software side. They came to us sometime before launch and said that they were happy with the algorithm but they really would like it to run faster because one of the main things the computer is doing is compressing images. Matt Klimesh and I spent a couple of months where we didn't do anything but try to make the software faster. And this is an indication of how much gain we were able to make. It is more than twice as fast now because of the software improvements we made. But again there weren't anything there that is a breakthrough, you know, a mathematical or scientific breakthrough, but it is certainly a case where understanding the math behind what's going on is important for speeding gains.

How come speed is so important?

– Because it really effects what they can do. If they have to stop and wait until the rover compresses all its images before it can move another several meters, that's time it's not doing anything except sitting there compressing images.

Do they have to stop and wait for compression because of navigation or why would they do that?

– A lot of the navigation is done based on people on the ground and things like that. At some point if you are just getting back all your images they need to be compressed and largely what they are doing is sending down images to learn about whether there is water on Mars. If the compression is running slowly it limits how many images they can get down and then it limits how far they can drive. The processor is a RAD6000-processor running at 20 MHz so it is not a very fast computer to begin with compared to what you might have at your desktop.

How many computers are on board the rovers?

– I'm not really the person to ask. What I think of is sort of the one main computer. I don't know what else they might have running that might qualify for processors, I'm not the right person to ask but effectively one.

– Here is a comparison of what we got on MER versus JPEG or versus Pathfinder which used JPEG compression. Here is the difference, this is a rate distortion

curve and this is PSNR [Peak Signal to Noise Ratio] so going up is a good thing in this case, it gives better image quality here and this is the bit rates. So this is on an actual Mars Pathfinder image, these are the actually rates obtained from Mars Pathfinder. You can see that we got a pretty big gap here. To put this in perspective as I said, if typically the images are compressed by one bit per pixel you can read up this curve and see what image quality that corresponds to and then read across here to see what rate you would need, if your statement is: 'I want this image quality, how much is this going to cost me?' Well, for ICER on MER it is going to cost 1 bit/pixel, for JPEG on Pathfinder is more normally 1.3 or 1.4 so there is a thirty percent gain for the images here. In fact PSNR is an objective quality assessment, you know, different images has different artifacts and so if anything this might actually understate the gain. One indication is that for JPEG on Pathfinder most of the images were compressed to about 2 bits/pixel whereas on MER they are compressing them typically to about 1 bit/pixel and that may have something to do with the quality of the reconstructed images.

- Here is a pretty vivid example. Here is the original image. These are both compressed with the exact same bit rate under JPEG and under ICER and you can see that the artifacts look very different here, you got this sort of blocking artifacts here where as under ICER you got these sort of ringing artifacts and an overall smoothing effect. So even though the PSNR for two images may be the same the subjective quality may be quite different.
- This is a comparison showing how well it does compared to JPEG-2000. The performance of JPEG [2000] depends a little bit on whether you are using an integer wavelet or a floating point wavelet but effectively they are all pretty similar here. They are all noticeably better than JPEG and they are all wavelet based. There are a lot of similarities between the algorithms; we are using the same context modeler as JPEG-2000 for example.

JPEG-2000 is not progressive or...?

- Oh yeah it is progressive. It can be used in a progressive way yeah.

But why didn't you just use JPEG-2000, it seems they are alike?

- Yeah it's a good question. ICER is lower complexity than JPEG-2000 and as we have seen speed is very important for these guys. The other issue is that we have been working on this and we have been given earlier versions of this algorithm to this back before it was the MER mission when it was supposed to be launched in 2001 and so we've had it and it worked even before then. By the time we gave them the algorithm, JPEG-2000 was just a committee that talked about things. There were no existing official JPEG-2000 that we could have given to them. If the mission had landed five years later we might just have given them JPEG-2000. So it had a lot to do with timing but also the other issue is that ICER is lower complexity. I can't say for sure because I did not write a flight qualified JPEG-2000 software to compare it to, but I think that if we wrote one it would be slower. In here we can see that it really did not give you much of a performance gain anyway.

But ICER is especially made for natural pictures whereas JPEG-2000 also includes computer graphics and everything else?

- JPEG-2000 is also wavelet based. It really is more suited for natural images. It is not going to work as well or at least it is not primarily intended for computer generated scenes. Like ICER it is really well suited for something that would come out a digital camera.
- This is the lossless compression performance comparing Pathfinder and MER over some of the test images that are found here. The images for Pathfinder are a lot smaller than the images for MER but here is an indication on average of these test images. It is about 10 bits/pixels or so for the Rice algorithm whereas ICER and LOCO are closer to 8.9 bits/pixel and you can see also that LOCO is almost as good as ICER for lossless images.

The Rice algorithm was on Pathfinder, right?

- Yeah so the JPEG algorithm, the image compression algorithm, is based on a discrete cosine transform and it is a lossy algorithm. You can't use it to get lossless compression. So if you want to do lossless compression you have to do something different and for Pathfinder what they did was the Rice algorithm.

– And I realize that 8.9 bits/pixel versus 10 bits/pixel on this set of example images, I mean it doesn't sound like a big deal but bits from deep space are pretty expensive so well, if it is ten percent more images you can get, you know, we spend a lot of millions to put the rovers up there in the first place. That is a big deal I think.

– I'll skip this. We looked at exploiting correlation in the spectral domain and I think I'll spare the details. So I guess this is all the slides I have.

Do you have more time?

– Sure I can...

It seems like you are working on a lot of practical problems. Are there any unsolved problems left for future missions?

– It's hard to try to identify... A big difference between data compression, or source coding, and channel coding is that channel coding has this nice Shannon limit, they know what the gap is between where they are now and where they'd like to be at the Shannon limit. It's a lot harder for us to say what room there is for a breakthrough to come through. And also like channel coding, and again the biggest channel coding breakthrough in the past couple of decades is turbo codes and that wasn't a breakthrough in... suddenly they can do something they couldn't do before except in practical terms. And certainly that is true here. Complexity matters for these missions; how quickly your algorithm can run, how hard it is to put in hardware. These things are all pretty important, if you have a good way of reducing the bit rate in an image by 20 percent that's a great breakthrough but if it is too complicated to put in the computer it doesn't really do us much good. So practical things are important. Unsolved problems? It's harder to identify because it really is more a mathematical thing, it's not really physics per say, it's not like we recognize that it be nice to have a grand unifying theory or you know a breakthrough in string theory we might recognize as being an area here. These bit incidents don't exists in the real world so it is harder to identify: 'What are the mysteries of the bits?' This doesn't mean that there aren't more breakthroughs on the horizon, it is just harder to anticipate what they might be.

But doing data compression with wavelets was kind of a breakthrough wasn't it?

– Yes that was a kind of a breakthrough but whereas in physics you might say 'it be nice to solve this mystery of the electron' or something, the question of dark matter or something like that. It would have been hard in advance of wavelets having been either invented or discovered depending on which kind of perspective you want to take on that, it would have been hard in advance for that to say 'we need someone to develop this wavelet theory'. And so what is the next wavelet? Without having seen it, it is hard to know what it is.

Will you be using ICER on the next mission?

– I think ICER is likely to be used for the hazard avoidance cameras in the Mars Science Laboratory. There is a competitive process to select the instruments and till the selection happens you don't know who is going to be in charge of the instrument and it is the persons in charge of the instruments who gets to decide what data compression they are going to choose to use. We can't force them to use ICER. So we try to help them make the decision for what is the right thing to do. That is under control of the investigator of the instruments so for the missions where they haven't yet selected the instruments... I can't tell you whether ICER is going to be used or whether something else is going to be used because I don't know. I think ICER is likely to be used on the MSL mission.

So the ones who produce the technology decide which data compression to use?

– Well the ones producing the instruments. There is some logic in that because it is the scientist who is in charge of the instrument who is supposed to be the best qualified to say... 'for these images I know what quality I need and I know what compression algorithms can achieve that quality'. So there is some logic behind that reasoning. The difficulty is that it is hard to be an expert on planetary geology and data compression so there is some randomness involved in what gets chosen.

The data compression algorithms are mainly in software?

– Entirely software.

There is nothing in hardware?

– That's correct.

And the decompressing down here on the ground, is that also entirely in software?

– Yeah it is done in software. The decompressing problem is... as I said if we had to, we could have a room of ten computers doing nothing else but decompressing. Only we do not have to, it's not that complicated. But yeah, it's all in software.

But it would increase speed wouldn't it?

– Decompressing speed is already fast enough. If you can keep up with the rate the images are coming with you don't really need to be any faster. Software is very convenient for the decompression anyway because if you want to trick something or change a parameter it is really easy to do in software.

So it is actually the opposite problem of what you have in channel coding where you want to do decoding in hardware.

– Yeah well as far as decoding in hardware for channel coding it's not that people love hardware so much, it's just that software isn't going to be fast enough. If they could do the decoding in software and keep up with the rate of which it is coming down I'm sure they'd be happy to do it but you can't do it that fast.

How many people have been involved in the data compression?

– Again first of all I'm familiar with the image compression data, as far as on mission data sets or other instruments you might be looking at I don't know, I can't answer for that. As far as the image compression it depends on how you want to count it in terms of, you know, who at JPL has been very actively involved? It's me and Matt Klimesh but on the other hand we didn't invent wavelets so there is a long list of people who have been making the breakthroughs. They don't necessarily know that they have been developing it so...

William Folkner told us that there are six thousand people working here at JPL, so if you are just two people involved with the image compression...

– Oh I'm talking just for MER. There are really about four or five of us working on data compression.

But it's a pretty small field?

– Inside JPL, yeah there aren't a lot of us doing data compression. One case is... if you want to look at say optical communications for example, is a case where – maybe it's not the best example but I suppose it is for the moment – optical communications problems that we are interested in for deep space are quite different from any optical communications problems that almost anyone else are going to work on so you might need twenty or a hundred people who do that. A lot of the data compression stuff that we are doing is largely just the same you would do for an image compression algorithm for the Internet. So there are very big chunks that are the same as for industry. We didn't invent wavelets here at the JPL because we weren't really forced to it. Not all of it but our needs are a little bit different. The bits for deep space are a little more expensive than the bits over the Internet and we have complexity considerations that other people don't have and in our case we want the encoder to be low complexity but we don't really care if the decoder is high complexity. So there are a lot of unique aspects for data compressions for deep space missions but there are also a lot of... we don't need to reinvent the wheel, we didn't need to reinvent wavelets. There are a lot of things that we can use so that's why you might find a small number of people working in data compression at JPL whereas you might have a lot of people working in optical communications or more people doing some specialized coding theory application.

So you would consider it more as giving already established theories a small twist?

– Yes that's not an unreasonable way to look at it.

Another question. In the 60's and 70's NASA had huge budgets compared to now. Has that reduction of budgets had any effect on the coding and data compression at all?

– I wasn't born until '68 so I'm not very well to give a whole lot of perspectives on what it was like to be here in the 60's and 70's [laughs]. I don't know...

We can try a different angle. President Bush seems to want to do something with the space missions, do you think that will have an effect on your job?

– I think it will, but it is always hard to predict what that effect will be. It is always hard to guess whether they are going to invest more money in more basic research or more applied research.

So when you do data compression you decide how to do it, that was your answer wasn't it. But are there any standards in data compression like in channel coding?

– Yeah there are existing standard data compression algorithms, there is the JPEG-2000 standard which is out now, there is the CCSDS group which among other things produces standards for data compression and things like that.

Space standards?

– Yes, CCSDS is the Consultative Committee for Space Data Systems. I think that is what it stands for. I'm on the data compression working group there. So when it seems reasonable and when we have enough people to get something done we try to produce standards for a wide variety of missions.

Is ICER part of those standards?

– No. For deep space our needs are a little bit specialized in that our data rates are typically much lower and so whereas for something that's in a lower orbit or something that can be collecting data and transmitting them in a much, much faster rate it becomes more critical that you have something that can operate really, really fast in hardware whereas for, you know, ICER on Mars we are not transmitting in such a high rate so having something that runs in software might be more reasonable. The considerations in those two applications may lead to a very different algorithm. There is an emerging image compression standard that is also wavelet-based that is not ICER. The performance isn't going to be quite as good as ICER but it is going to be something that is easier to implement if you need a really fast hardware application.

So the space standards are more concerned with television transmission or…?

– No I don't think so, probably remote end things, images, weather forecasting perhaps, it depends. I don't think it typically is something were they worry about commercial applications. It is typically science applications. It is not video. We have not yet done any work on standardizing the video compression standard. It's not so clear that we would do that.

I've been wondering about your shirt. Did you study at Virginia?

– That's where I got my bachelor.

How did you get into JPL?

– I applied for a job here after I got my Ph.D.

Where did you get that?

– University of Michigan.

Is it okay if we ask you that? The thesis also has an aspect of the human side of doing research at a place like JPL. Our questions are not going to be that personal or anything?

– I'll let you know if there is something I won't answer [laughs].

Is your job at JPL the first you got after doing your Ph.D.

– Yeah the first full time job, not counting summer jobs and so.

Would you say JPL is a place you come and stay for life or is it just a stop on the way to somewhere else?

– I suppose it depends on who you are talking to but I think compared to a lot of industry places this is more... I don't think that very many people come to JPL thinking 'well I'm going to be here for three years and then move on to something better'. I don't have that perspective. I'm not looking for a new job. You know, if something happened to the funding picture or something that made working here a lot less pleasant than it is now I would certainly look for a job but at the moment with things being where they are now, I mean, you get to work on the Mars rover, in how many other jobs do you have an opportunity to do something like that? Of course if something came up family wise, you know, but not in terms on what I get to work on here.

So it is not something kick starting a university carrier?

– For me it's not but yeah I could see why someone might do that. But if you have a Ph.D. and you did a decent job on that you can apply for a university job directly. You probably wouldn't need to come here, of course it wouldn't hurt you to come here for a little while but it might not be necessary.

You said you were born in 1968. Are the people working here mostly young people?

– Most of the people in my group were born in '68 I think. But when you say 'here' do you mean JPL as a whole or...

Whatever you have any comments on.

– The problem is that I haven't been around in industry outside so I don't have a whole lot to compare it to. We have I think a much higher fraction of people with Ph.D.s here so the age at which you might start at JPL is that much later than the age at which you might start elsewhere. I think that JPL is the kind of place that people tend to stick around for a while. You do get a lot of people who are here for several years. People are not here because this is where they can make the absolute highest number of dollars, they are interested in doing what they are doing.

The excitement of this job is that when the rovers land on Mars or is it working with the algorithms?

– Certainly the biggest spike of excitement is when the rovers land and you get the signal back and you realize that it didn't crash into Mars, but there is a long time to wait in between that so it can't be all, that's not enough, you have to really be interested in what you are doing. So there has to be, maybe excitement isn't the right word, but a significant interest in working on something new or, you know the work that we did on making ICER faster for the MER missions, we are not going to write a paper about how we changed this line of code to make this faster, it's wasn't that we came up with some great breakthrough in the way of doing things but there certainly is a lot of satisfaction in thinking that the algorithm is now twice as fast and this is something that's flying on Mars and it's going to have a direct impact on the mission. There is certainly a lot of satisfaction in that.

Did you work here during the Pathfinder mission?

– Yes, but I didn't have anything to do with the data compression, so all the information I have on what was used on Pathfinder came through other people.

Okay, well thank you.

Figur 4.2 På rundvisning ved JPLs sandkasse. Til venstre: Bjørn foran sandkassen. Til højre: Doktor Albert Haldemann, der viste os rundt på JPL, i dialog med Bjørn.

4.2 Doktor Matthew Klimesh 14/3 2005

What major breakthroughs would you say there have been within data compression?

– You are interested in maybe the last twenty or thirty years... Data compression has changed quite a bit. I think it is largely due to the computers becoming more

powerful and memory becoming cheaper. Previously people were limited to very simple ideas in data compression just because they didn't have much processing time and the computers weren't very powerful, so they just tried something very simple to see if they could get an improvement and if so they used it. Nowadays there are a lot more processing power available. The theory has advanced also. I mentioned wavelets, and there is a lot of mathematics behind arithmetic coding, a method in entropy coding. You know, basically if the probability of the symbol is known... if you are coding a group of symbols and you know the probability to each of them. It's very efficient in terms of being close to the theoretical limit. Arithmetic coding, I think, developed in the 1970s or 1980s... I don't know if you would count that as a big breakthrough though. It is nice and convenient for those who work in the field because they have data to compress and they want to take their generated symbols with certain statistics and they want to compress them, they just throw in an arithmetic coder and see how well it does. So they don't have to think about it at all. Arithmetic coding sort of grew out of the mathematics behind data compression, I think. Actually though, the ICER compressor doesn't use arithmetic coding, it uses something else – you can look in the paper and follow the leads there.

So those are the breakthroughs?

– Well, I'm sure there must be others. But again, data compression is not really a field that you would expect to have great breakthroughs. A lot of times you can measure how close you are to the theoretical limit. If you know you are only 5 percent from the theoretical limit, there can't be a breakthrough of more than 5 percent. But actually I guess for image compression it's not so easy to figure out how close you are to the limit. But still... how well we can compress... probably about 10 percent from the theoretical limit for image compression. But people have thought that before and been wrong. There have been a lot of changes in how people think on data compression.

Could you tell us a little bit about the making of ICER. How long did it take you?

– Well, I wasn't involved in it at the beginning. It sort of evolved over a period of time. It had one version that worked pretty well and we rewrote it from scratch. Trying to make things a little bit more efficient and fitting together nicely. But there wasn't really any mathematical events involved in matters.

– [temporary interruption]

– Basically it was most engineering, taking what we already knew how to do and putting it together in a nice package. But there were decisions involved also. Actually our development of ICER happened about the same time that the JPEG-2000 standards were being developed. There are a lot of similarities between ICER and JPEG-2000. Some of them are because we decided to make the same decisions they did and some of them because we thought 'I wonder what JPEG-2000 is doing on this' so we looked to see what they were doing.

So for how many years were you working on ICER?

– On and off we were working on it for 3 or 4 years, but it was just a small fraction of this time that was used. Some of it wasn't so much development as just making the code more efficient and making it run faster.

How about spin-ons and spin-offs. Do you think there will be any commercial products due to the ICER algorithm?

– Because there are a lot of similarities between ICER and JPEG-2000, and JPEG-2000 is a standard, there already are lots of people developing stuff for JPEG-2000. I don't see anything left for anybody to develop something based on ICER. Some of the decisions we made in the development of ICER were specifically for problems that happen in the deep-space applications. So for example I think it is actually a little bit lower in complexity than JPEG-2000.

Would you say that the ICER algorithm was sort of a spin-on from JPEG-2000 or were they two parallel projects?

– I'd say they were two parallel projects pretty much, but we were sort of keeping tabs on JPEG-2000, seeing what was happening there. We didn't really want to wait until JPEG-2000 was completely developed and then implement it, because 1) there are a few specialized features in space applications that ICER is more suited to and 2) we needed it sooner than that, so we couldn't wait for them to finish.

Do you know anything about the history of the use of data compression in the space missions?

– I don't know as much as I'd like to. The earliest missions didn't use data compression at all. The Voyager spacecrafts, which I guess didn't go to Mars, used what we call Rice-compression, which is done in hardware basically. It is a fairly simple algorithm; it compresses 16 samples at a time.

– Right now for lossless compression we do quite a bit better, maybe 15 percent better than the Rice-compression on the same image. So that is an improvement. ICER can do lossless compression and on MER we used a variation of the LOCO-I algorithm, which was developed at Hewlett-Packard and became the basis for JPEG-LS, the lossless JPEG, which doesn't really have much relation to JPEG-2000 – it is a different algorithm.

But the use of data compression on the missions also made the use of channel coding more severe didn't it?

– Yeah, when they started to use data compression the requirements for the coding went up. I don't have a good feel for much though.

We also have some more 'personal' questions regarding the human aspects of doing scientific research at the JPL. Where did you take your education?

– I grew up in Michigan and I went to the university in Michigan. I got both undergraduate and graduate degrees there.

Did you come straight to JPL afterwards?

– I worked for Michigan for about a year as a post doc. But actually till that point I hadn't worked on data compression at all. I didn't start working on data compression until I came here. Actually that is not so unusual in engineering... what I worked at in Michigan was still on the side of information theory and in fact a little bit more into channel coding. A lot of the mathematics is the same and when they hire someone, what they really want is someone who is good a solving problems. Having experience in a specific field is useful, but they are similar enough... there's a lot of stuff that can be transferred from one to another and I had taken some courses in data compression.

Do you think JPL is a place you will stay for the rest of your carrier or are you moving on?

– I'll probably be here for a while. I don't know if I'm typical, that's more a reflection on me than on JPL. I'm the type of person that goes a place. I don't like to move around a lot. There are a lot of people who come here and stay here for a long time. We get a newsletter every two weeks and one of the things it mentions is people who retire after being here a lot longer than... there are people who have been here for 50 years before retiring.

Would you say that JPL is a place where the research being done is mostly basic research or applied research or engineering?

– Well there's a lot of all of that. JPL is a big place. Building a spacecraft like the Mars Exploration Rover there is a lot of engineering involved but JPL also has scientists. I'm an engineer basically. As far as engineering and basic research, we like to do basic research but there is always some pressure to do stuff that will be useful next year or during the next two years. I think that is true everywhere, it is hard to get people to pay you to do basic research so it's a good thing to do when you are a graduate student.

So when you do data compression, is it always a question of taking things off the shelves and giving them a little twist or do you have to invent new things?

– Well we do some of both. We do some new development. For example in the development of ICER I would say both were involved in that. There is some theoretical figuring out the best way to do this and there is some figuring out good ways to program it, which I would say is just an engineering problem. So there is a little bit of both.

Something totally different. What are the rover images that are being transmitted for, are they primarily for doing research or for catching the public's attention?

– Well... They certainly care about catching the public's attention, but I'm not sure how much of a driving factor that is. For example I know that the MER rovers by now have taken a total of something like 60,000 images, maybe more

than that. So obviously not all of those can be released to the public... well I guess they actually are all released at some point, but the public doesn't see all of it, so...

Are the images the most effective way to do research?

– I'm not really a good person to talk to about that. But I do know that... I think the images are important for the research. They do a lot of other things with their spectrometer tools and their thermal emission, they look at temperatures. I do know that for MER the images are used quite a bit just to pile operations into navigation and rover driving and things like that. But I guess that is not really a science operation. All of the cameras on the rovers are stereo pairs except for the microscopic imager. On the ground they'll computer-generate 3D-representations of the surface and use those in the planning.

When you get an assignment like ICER are there anyone who tells you what algorithms to use or do you decide that yourselves? Are there any standards you have to follow?

– There are some data compression standards being established but they are sort of... largely concerned with orbiting spacecrafts... things we come up against with getting signals from Mars... the constraints are a little bit different. So we feel free to basically design whatever we think will work best. They generally don't tell us what to program – it can happen – but basically they want us to tell them what they should be using.

The ICER and LOCO-I algorithms are concerned with image compression. What about the data that aren't images how are they compressed?

– Well there is a lot of just engineering data which I think is not compressed at all, but that volume of data is very small. There are some other types of data being generated and I know they are using some general data compressor which is not all that different from the UNIX compress function. But we didn't develop it although I did look at it and I'm not quite sure what data they are using it for. I think that images are still the vast majority of the data so image compression is more important than any other data compression

Do you know anything about the budgets, the funding that JPL is getting?

– Oh I would just be guessing. I try to not worry about that as much as possible.

4.3 Doktor Jon Hamkins 14/3 2005

We're doing a master thesis in mathematics about the mathematics involved with the Mars mission. There are a couple of aspects to it. There is a general survey of what math is in the mission overall and then we're trying to dig deeper into two aspects of the mission; data compression and channel coding, to see if there is anything new there or if it's more a usage of existing theories with a twist.

– Okay, sure I can tell you a bit about the history of what's been flown. Space communication is unique in that, well for deep space communication there is a very high premium on the complexity of what's put on the spacecraft and no premium on the ground, so for example when they're doing an uplink from the ground to a Mars spacecraft they're assuming that we can just blast power from the ground, as much as is necessary and the converse is also true, we have very limited power, something less than a Christmas tree light bulb to transmit from there to here so we have humongous, seventy-meter deep space antennas trying to piece together the signal that is coming down. So all of the mathematics that's involved in the decoding side of the channel coding for the downlink signal is somewhat unconstrained. Whereas the very first probes just because of their simplicity didn't have any kind of channel coding on them... when they started using convolutional codes, the first things they tried were things other people were trying for other kinds of applications which is sequential decoding, I don't know if you've heard about it but basically it's the kind of thing were you have a trellis description of your convolutional code and there are potential hypothetical paths through this describing the data send and a sequential algorithm will say: 'Okay, let's try this path' and it keeps going until the observables don't match with your computer metric along the path. Just like the Viterbi algorithm, except you can determine when it doesn't make sense and you can try going back and then take another path. So you can limit your

complexity in how many paths you're willing to try and eventually you will get one that's reasonably good. So the early missions used sequential decoding. Following that Andy Viterbi was actually at the JPL and he said: 'Oh well, it's not too hard to just try and compute the best path that's heading into each state' and so he defined the Viterbi algorithm that gave you the most likely path through this trellis and that was... well it could be a little less or more complex than sequential decoding but at least it gave you the most likely solution. Shortly after that... and this was not appreciated at the time... even though this gives you the most likely path corresponding to the most likely sequence of bits that was send as opposed to some other sequence that differed in some subset of bits between one and the other... it didn't necessarily give you the most likely bit at each position. So if you want to know what is the most likely value at a position the Viterbi algorithm doesn't give you this. It just gives you the entire string but the particular bit is not guaranteed to be the most likely. So the group of four authors devise that what you really need to do is: In order to identify what is the most likely bit at position i you can do one forward path and then do one backward path and keep track of a certain set of variables in a certain way and then you can compute it and this is called the BCJR-algorithm. The mathematics of exactly how that works is detailed in I think it's a 1974 paper, the initials BCJR are the initials of the writers. As it turns out nobody ever uses the BCJR-algorithm for just regular convolutional decoding but it turned out to be the most important fundamental algorithm that's used in turbo decoding. When turbo codes were first published in 1993... first of all at the conference they were presented at nobody believed it... [drawing on a piece of paper]. When turbo codes came it was just a tremendous improvement after what had been decades of usage of these other codes. Nobody believed that it was true because the codes jumped right over the cut-off rate. The cut-off rate is... it was widely believed that without exponentially increased complexity of your decoding operation you couldn't exceed a certain performance limit called the cut-off rate. These turbo codes just jumped right over it and this led to a renaissance of channel coding research... the first thing everybody did was to rush home to simulate these turbo codes and indeed they performed much better than the other codes even though they were composed of the same old convolutional codes that everybody had been looking at all along. But the advantage was that they had been combined in a certain way and the key was that they used the BCJR-algorithm for the constituent code. So the structure of a turbo code is... you have input data coming in. You send it into a convolutional code, you take the same data and permute it and you send it into another convolutional code. Then you take these to outputs and you merge them together to one bit stream and you send it. The advantage of this is that with very low complexity, with very simple constituent codes – the interleaver can be very long – you get an effective block length increase. At the decoder you use the BCJR-algorithm to decide... I'm not explaining this very well... but the end result is that for very simple constituent codes you can get tremendous improvement. JPL's involvement in the development of turbo codes was on the standardization of the particular codes that are used now in deep space communications. The Mars Reconnaissance Orbiter mission, which is launching in 2005 is using turbo. The Stereo and the Messenger missions are also using turbo. Going forward, even though these are much less complex than other types of codes that would have the same block length the second renaissance if you will of coding theory has been Low Density Parity Check codes. Low density parity check codes as it turns out were detailed in a thesis by Bob Gallager in 1962... he wrote a book, I think it came out in 1963 and was just called LDPC-codes in which he said: 'Okay, you got a parity check matrix which is $n - k$ by n and you populate it with... for example six ones in each row and three ones in each column and these codes are very good. He gave a simple decoding algorithm that was even simpler than the turbo decoder... and they performed very well. When these codes came out we didn't have the computing resources to simulate them properly, so David MacKay who is a student of Robert McEliece really led to the reinvestigation of these codes and showed that in some cases LDPC-codes are outperforming even turbo codes, which nobody appreciated before. Now it turns out that turbo codes and LDPC-codes if they're designed properly are comparable in performance, but the advantage of LDPC-codes is that they are much less complex to decode than turbo codes and they have the advantage that

the particular structure of the decoder is parallelizable. That meant that even though they were more complex you would still want them because of the fact that you can distribute the computations. So what JPL is doing now is... we are in the process of standardizing the LDPC-codes that are going to fly in the future and so there is a mission called MTO which is launching in 2009 where we are tentatively planning to use LDPC-codes. That's where we stand now. There is a lot of mathematics behind the algorithms of LDPC-codes... I recommend you to go to MacKay's homepage. He's got a lot of material on LDPC-codes... The LDPCs are curiously really the first codes that are being defined by their parity check matrix not their generator matrix, so even though you have the parity check matrix this doesn't tell you how it's encoded. So it has been the case where people have found a very great LDPC-code but they didn't actually know how to encode the data even though they knew how to decode and so sometimes backing out and deciding the right way to a lot of this has been difficult. So the emphasis now, because these matrices can be very large you want to have some structure in them so that it's not completely random – that makes it much more complex to deal with. When you start imposing structure, that could affect performance. So there is a lot of very subtle research going on. So anyway that's where we are in a nutshell.

The current mission uses a Reed-Solomon code and a convolutional code, right?

– The majority of the missions flying now are concatenated. So the data comes in and is Reed-Solomon encoded, then it goes through a block interleaver and then it's convolutionally encoded. The reason for that is you see... they suggest using the (7,2) convolutional code, but say you are transmitting all zeroes and you look at the error patterns of the decoder... if everything works properly you get all zeroes, but if you look at where you get the ones they are random. But because of how the Viterbi algorithm works they tend to get bunched up, so you get burst errors. This is great for Reed-Solomon codes because if you got a Reed-Solomon code on top of it then number one: All of the bits that are part of a Reed-Solomon symbol... the standard is a (255,223) Reed-Solomon code which means that each code symbol was living in the Galois field of size 256... and so it takes 8 bits to represent one field element so eight bits in error here [points to paper] would only represent one Reed-Solomon symbol in error and so since this can correct up to 32 symbol errors you could handle many bursts and the Reed-Solomon would still correct it. And the interleaving... in addition to getting bursts you want to spread the bursts so you have long periods where there are no errors at the output of your convolutional decoder so the bursts would tend to be bursty as well so the interleaving would scramble these things out in that way and the Reed-Solomon codewords could be decoded. This technique has been used for thirty years and there are still missions signing up today that are planning to use concatenated Reed-Solomon codes...in particular what's called the Electra Radio which is a proximity one Radio where for example...now when we send Mars exploration rovers out... [draws on paper]... here is Earth, here is Mars... we have a rover here, and as it turns out it does have capabilities of communicating directly to Earth but it's got a pretty low data rate. A much faster data rate is going through Mars Global Surveyor, so more than ninety percent of the data has been relayed through MGS. The short links [between MER and MGS] is called proximity links and the future of proximity links for NASA is going to be through this Electra Radio. So all the orbiters and landers in the future are going to be flying an Electra Radio, which has expanded capabilities over for example MGS. The Electra radio right now supports Reed-Solomon plus (7,2) convolutional codes.

Why not turbo?

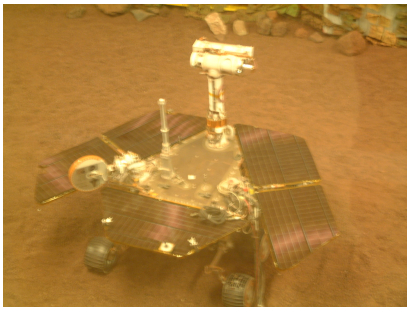
– Oh, because of the time it was built it wasn't ready yet. The process of flight qualification is very long actually. Even though turbo codes was 1993 it was less than a couple of years ago that turbo codes were flying in space. The process from invention to demonstrating it in a laboratory hardware like we have here to flight qualifying them on radiation hardened parts... and you know missions that are signing up for a very complicated space craft... they are out to minimize risks so they want stuff that has been flown in previous missions, they don't want something new. It's kind of contrary to the spirit of exploration. They don't want risks even though we are confident that it works... it is a risk to a mission if it hasn't flown before. So when they sign up for a new thing it is really

a big win. Once they start seeing the advantages, increased power efficiency and those kinds of things generally they come around. So even though there were off the shelf radiation hardened part for concatenated Reed-Solomon codes there weren't for turbo codes. So Electra and a number of other missions will still be using Reed-Solomon. For the ones where power is really a critical thing, which it often is, they're going to be using either turbo codes or LDPC-codes. Each has it's niche of where it's applicable.

Why are the dimensions of the Reed-Solomon codes so large? Why not use a smaller code?

- You have to remember that they are not intended for themselves. They are intended for use only with the convolutional code. So you look at the burst error characteristics of that convolutional code and make the code so it's just good enough to give you the performance you desire.
- Anyway I have to go. Good luck on your project.

Thanks.



Figur 4.3 Stadig på besøg ved JPLs sandkasse. Til venstre: Uffe foran sandkassen. Til højre: Kopi af MER-rover. Denne anvendes til testkørsler på JPL.

4.4 Doktor Jacob Matijevic 14/3 2005

- When I first arrived here over twenty years ago there were still efforts to hand-implement certain mathematical models for certain applications. And there were specialist applications here for specialists in the applied mathematical sciences who worked here to make those applications possible. But over time much of that has been incorporated in fairly standard and available simulation and modelling packages – computer packages. Expansions have been introduced slowly over time to these packages and that's basically how the engineers here do their job. Instead of going back to first principles they apply these tools... the foundation theories are from the eighteenth century to a large degree. I've seen a lot of improvement on the foundation work that is actually applied here. Now there are the fairly simple theories and techniques that are applied here to our job, you know, the Mars mission. I'll try to concentrate on the surface operation areas but I encourage you to talk to others who can tell you more about some of the modelling and simulation that enter into the design phase as well as into the development of the basic launch; the launch trajectory, launch parameters that make it possible for us to get these missions to Mars in the first place. I think there is also another entire set of simulation tools that are involved in the activity of simulating the entry and landing events at the surface, which I think were major applications. I don't believe any of the launch, cruise, landing or surface operations are in some sense known entities. Even in the place where we succeeded to land there are a number of variables not fully accounted for simply because we're still learning about the planet and its environment.

- In your email you seemed to focus a little bit on the controls aspect for doing the driving and steering. Mainly what we're taking advantage of to at least

create the driving pattern, is the fact that each of the individual six wheels is controllable at least for steering and so we can create in essence any kind of arc condition. Because of the individual control we can modulate the speed of the motor turns. That gives us a means for being able to accommodate surface interactions between the wheels and the terrain. The foundation is actually fairly simple. We're using in each case a simple proportional integral derivative; the control algorithm is very linear. The way it was implemented for us was that it was placed in a piece of firmware, I don't know if you know much about field programmable gate-arrays. So instead of actually running the algorithm inside a computer we run it inside this little chip, and essentially give the chip addresses of a portion of memory that allows it to put certain variable parameters in for the speed control. We actually do this control on voltage so we can modulate the amount of power that's actually applied to the motor. And we simply go from there. We kind of do this in a sort of staircase fashion. We'll turn the wheels, drive the wheels to move and turn the wheels back. Ahead of that will be a software package that will come up with the desired path we want to take in the terrain. In essence all we do is come up with a simple translator of how we implement that path by doing a combination of these steerings and drives. So the fundamental thing here is a simple linear control application. Above it is a planning application, which we do on each of the steps in the drive. There are actually a couple of modes we can use and what I'll describe is the one where the onboard system makes most of the decisions. What it is essentially trying to do is to process a view of the terrain around the vehicle and then make a decision as to where it wants to go – what is a safe path. So it creates little arcs that represent the kind of fundamental path the vehicle can take, and it chooses amongst the possible paths it can take; which one seems to satisfy a certain safety criteria. It's sort of a little simulation: 'If I move my vehicle over this area, do I violate any constraints'. So it is more of a logical evaluation of a series of possible paths, some of which may be backwards. And so we try to propagate that one step at the time and repeat this process over and over in the course of doing a drive.

So the amount of artificial intelligence involved is really rather limited?

– It's very limited, correct. It's kind of a compromise, you know, between attempting to be more sophisticated in the kinds of things you try to look for versus the need to actually make progress with your drive. We can't drive all day – there is generally not enough power or not enough time in the day – it's basically a day-operated vehicle, and so you try to balance the rate of progress with the other things that you wish to do. On vehicles such as ours the job is not solely to move across the surface, but actually do measurements before we proceed. Of course it is important to get to a location where we could get something important in terms of measurements. But it is also important for us to document where we've been – make measurements at selected places along the way. So this simple logic is what is on top of the underlying control. Now, that's kind of a driving strategy for the vehicle. The analysis lying before this... I worked on it well before the mission began, it was a real design phase... we had to come up with models that could convince us that the vehicle was not going to break a wheel or do anything that would cause it to not make good and predictable progress. So in that instance we actually did little simulation models to model the flexibility of the individual components based on the individual masses and that helped us with the basic drive-strategy.

All these packages you talked about, are they commercial packages?

– For the most part, yeah, the packages are available from commercial vendors.

– If I can switch to another area of surface operations – in the thermal area we have take into account the fact that this is a vehicle that is seeing the thermal cycles that are present on the surface of Mars. A model-based version of the vehicle is implemented in a package. What it does is it implements the key transfer equations which is essentially a series of boundary value problems that gives rise to what takes place when components are heated and are radiating into a combination of structural components which are connections to the components and the atmosphere of Mars. We do not fly a vacuum package [laughs]. We intake the atmosphere once were on the surface of Mars and so in our components we

have to take into account the fact that there is not only simple heat radiation but also convections that are consequences of what the atmosphere is like. So the thermal models that we put together to represent the power on the components try to represent what we think the temperatures will be at individual points in time. We use these transfer functions effectively to come up with models of how the components will behave. From the combinations of those plus our understanding from missions done in the past, we came up with a design that allows us to effectively be able to trap heat inside a box we call a warm-box where the computer and the main pieces of electronics are located, and we can kind of control that like a separate environment from the outside. Essentially we're collecting waste heat with thermal isolation and then externally all we do is to sort of wrap individual components to let them avoid contamination from dust and things like that.

So it is a big challenge to predict how the instruments work. Something like measuring the temperature on the surface, that's a really hard problem.

– It is. A lot of this was done by inference as much as anything else. The key thing about the development phase is that we came up with a test program that allowed us to eventually take these vehicles and put them in a chamber and attempt as best as we could to replicate the environment we thought we would see on the surface to check that when we went through the operations that the components would match what we predicted. Once we got to the surface we actually continued to run this model – it is actually a daily operation for us on both vehicles. We only get a limited amount of information each day – we are not awake all the time – so a lot is based on the temperatures we can see and working back to make an inference as to what the environment we are seeing on the surface is. The models have gotten pretty good over time as to predicting the actual temperatures, both inside the warm electronics box and outside as well.

These models, were they involved in the Pathfinder?

– Versions of them were. That's correct. This particular package that we are using wasn't developed at the time. This was ten years ago. But the basic principles were there in other packages. It is part of what we need to do for the surface operations. Another part of this is that we over the course of the last twenty years have put together what is in essence a surface model for the environment on Mars. This was the result of combining understanding of how the atmosphere varies over the course of the year with what measurements we have taken from the various orbiters since the Viking days basically. We've come up with what we consider to be atmosphere models and ground models for different points of time during a Marsian year, which we can use for the purposes of modelling how the surface mission will be. The latest generation of this is a global circulation model, which involves – I'm going to speculate a little bit about this – but it involves basically these models of the Marsian atmosphere all the way down to the surface. For the surface we've picked up quite a bit of data from things like the Mars Global Surveyor, Mars Odyssey and the Viking missions as well, and they essentially tell us how the heat that is collected by the surface during the day creates a kind of circulation through the atmosphere causing convection currents over time. Those measurements can tell us at least what we should expect the temperatures to be in the winter and the summer. We use those as early environment models to run our simulations through, to decide how well the components will work. So we ran versions of our surface operation three years before we ever landed on the surface. Another reason for looking at this is that the same models were used for analyzing the entry of the vehicles into the atmosphere. Based on the time of the year and the day we were arriving and the conditions we were expecting to see, we could run this through simulations of how the density of the atmosphere would affect what we would see in terms of our heat shield. They were also a factor in our implementation from the early Pathfinder days to what was done for our MER-mission. So that is all done in simulation before we ever build the equipment.

We saw a TV-documentary were they did something on the parachutes. It seemed that they tested them more physically than in simulations.

– It was a combination of both. We did drop tests. We did wind tunnel tests with the parachutes. But even before this time it was through models of the profiles

of these devices that we came up with things like what the entry angles would be, what sorts of release points should we be looking at, as well as designing the algorithm that checks for height above the surface and finding out at which time to deploy the parachute and at which time to fire the rockets for slowing the descent. All of this was based on what we expected to be the environmental profile that the vehicle would see as it came down to the surface. So this was all done in simulation.

It's a bit surprising that the landing with the parachute is so difficult in a way because you think of parachuting as something everyday-like. When you compare it to sending a picture from Mars. Now that's a big eye catcher and so 'it must be very complicated'.

– Yes, but the thing is. Here on Earth, there are many things we know through years of use of these devices, simply as a result of our experience here, but we've landed now five times... [laughs]... successfully at least, and each time is at a different time of year on Mars and at a different location all of which can be significant factors. For our next landing mission we repeat this whole process because we basically don't land the same place. And we don't land the same time of year. So that makes the difference.

– I'll see if I can get to another area. In communication theory, you are thinking about the individual coding strategies. Most of the data is now send in relay from Mars Odyssey and we're essentially transmitting data with much less coding overhead. It is essentially doing checksums and a retry-protocol that we use, because data is only being send hundreds of kilometers as opposed to millions of kilometers. It's much simpler to do this, and the overhead is much less. I think maybe the message from Mars is, that when you are in a situation where you can do this kind of relay-communication it's much less of a problem than when you are doing it over these long distances. So most of the coding, I think, is now implemented for things like memory storage devices on board the vehicle where we used basic cyclic redundancy codes for storage. It is meant to allow us to account for the fact that these devices are sitting there and eventually will get affected by cosmic rays. Really what we are trying to do is preserve the integrity of the data that we store there. So in segments of the memory we have these encoding bits that we check when we are doing our reads and writes and such. So that is probably where more of this is applied now. The other thing was that we intended to use the ICER algorithm which is actually a wavelet based algorithm for compressing our images and sending back to Earth, but it turns out that the relay-communication work so well that most of the time we send back our data with lossless compression and very little compression-artifacts put into the system. And that's it for the coding. There is not much used.

So the encoded data is just send from the rovers up to the surveyor?

– Yes and Mars Odyssey has a big buffer of data that it stores from all of its instruments. And it essentially packets those with traditional Reed-Solomon coding and sends those back through our Deep Space Network.

So the more serious coding will be done by the Odyssey?

– Yeah. [...]

What do you think the next generation vehicle will look like?

– We're sort of in the midst of starting a design for at least the next version of the vehicle system. The thing that really distinguishes the next vehicle from these is the computing resources, which will be substantially greater in the next vehicle. That gives us a chance of being able to implement more sophisticated estimations and control algorithms. Probably not the lower level wheel-drive systems as much as the higher level systems that apply the planning strategies. This will allow the vehicle to drive longer distances by itself. I think everybody is happy with the imaging system so I suspect we'll just replicate it. Maybe there will be more sophisticated compression algorithms we can apply. I think we'll probably try to add atmosphere packages next time around. Also the next vehicle is supposed to do sample collection. That means it will work difficult problems, not just simply taking measurements, but also be able to drill into something and collect something. That's more of a challenge. Especially because we don't know what we'll be looking at. Part of the issue has always been: 'How hard are the rocks on Mars; what is the soil you're trying to capture; how loose

or solid is it; how to capture it and how to store it; how to keep it more or less intact so you can bring it back?' Things like that.

So the reason why you don't have these facilities; is it money issues or time issues or...?

– It's those things plus there is this orderly progression in the missions here. You have to view this, as really new territory for us, and one of the thoughts were that you would try to get some characterization first and then you could go further.

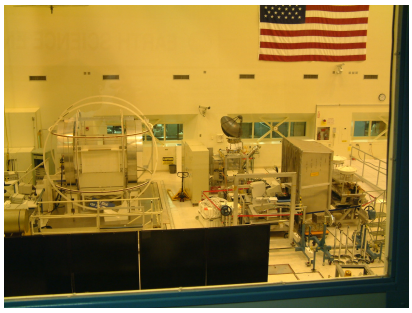
So steady progress.

– Steady progress. Yeah.

[Interrupted by Albert Haldemann.]

AH Hi. I'm Albert Haldemann. So have you finished asking questions? I'll take you to lunch and I'll show you the indoor sandbox where we do the testing. Just so you'll get a sense of it.

Great. Thanks.



Figur 4.4 Fortsat rundvisning. Til venstre: Et af JPLs laboratorier. Til højre: Uffe og doktor Albert Haldemann med udsigt til laboratoriet.

4.5 Doktor Miguel san Martin & doktor William Folkner 14/3 2005

MsM Let me tell you first what my role was. I was associated with the Guidance and Control System of the spacecraft, which consist mostly of the flight software and hardware that was on the spacecraft. Some of the software goes also into the support equipment that is used to test the guidance and control system. It is all a bunch of programs that we use on the ground to do the theory behind the guidance and control techniques and testing. There are three phases of the mission; cruise, EDL [Entry, Decent and Landing] and surface. In the cruise we are spinning the spacecraft. We use conservation of angular momentum to keep our spacecraft from turning. The most sophisticated piece of software that we have onboard during that time is a Kalman filter. I don't know if you're familiar with Kalman filtering. A Kalman filter is a statistical framework, or algorithm perhaps, to mix information from different sensors. So you have a dynamic model – in this case it's a simple rigid body, which you represent mathematically and then you have some sensors which measure where the Sun is. And you have a statistical noisy model. To figure out your attitude [position of spacecraft relative to a frame of reference] you need to look at two stars. One gives you two axis. Two you get a full three axis fix. So you need to look at the Sun and another star. We like to use the Sun because it is easy to identify, so it's safe. It's always there. Stars you need to figure out. I'll tell you a little bit about the stars later. So this Kalman filter is an algorithm that allows you to... for example given the mathematical model of the spacecraft and your measurements, you can

estimate what your attitude is and also solve for additional things. Our inertial properties are not perfectly well known. You know what the principal axis of a rigid body is? It's the axis you want to rotate. We tried to align that with the z-axis and estimate the angels. There are certain parameters that are observables. So we try the best we can to estimate the ratio of the cross product of inertia. This formulation is pretty common. There are several books, you know. It was invented in the '60s... You can come up with a sequential filter that allows you to optimally combine the information from three things; your dynamic model, your sensors and your star tracking into optimal information about your attitude and the inertial properties of your plant. So that is going on all the time. It's a well known aerospace industry. We compute a covariance matrix that tells us how well we estimate these things, of how well we are doing. Of course the covariance matrix is a theoretical representation, how we are really doing, nobody knows. We do simulations here on the ground where we have a model with a real attitude and we put noise into it and let the estimator tell what attitude it is and we compute the difference from the actual, and that's the error. Then we compute the covariance and see where the error is within the covariance. So that's a typical thing. The actual control laws are very simple for a spinner. If you want to move in this direction you just apply some torque in that direction. It's as simple as that. As you get nearer the target you hit it less and less so you get a convergence. We try to do it as simple as we can. Everything is very ad hoc too. In order to be able to use the stars to compute our attitude we need to know: 'What star is that?' You need to know how to look in a catalog and find out what its position is so I can determine what my position is. Unfortunately the magnitude of the star, how bright it is, is not unique. Given a position where we can measure the magnitude, there are multiple stars and the dimmer you go, it's sort of exponential, the dimmer you go the more. The brightest star is Sirius; you can look at it and see 'that's it'. But there is only one of those. Then you go to the next level and there are three and suddenly you have a lot. So magnitude is almost worthless, but we still use it. So we need to look at the patterns, you know distances between stars. We have an algorithm that collects data for two minutes; it's a scanner looking around for stars. So we have these blips, okay. In order to identify which star is that blip is we compute two things: How bright it is, the sensors tell us that and then we compute the angle between that star and this. Then I go to the catalog and I look for stars with that brightness and that angle, but believe it or not that is not unique either. So this blip could be this, this or that star. So you get a list of candidates for each blip. Then the way we do it, again different people do it different ways, you can start playing with star patterns. It's a pattern recognition problem. The way we do it is using the Sun. That gives us two axes. Then we compute an attitude quaternion. Are you familiar with quaternions? Quaternions were invented by... Hamilton!

Yeah, a complex number with four coordinates.

MsM Yes, that's the way we represent attitude. Attitude has three elements. When you use quaternions you only have one redundant element and if you do the math of attitude propagation on quaternions that is numerically very beautiful and very simple, so we use quaternion math. So going back, based on the star and the Sun measurement, you start using the candidates. This blip is star blah. So you compute the quaternion and you go to the other ones and compute the quaternions. When you get the right star in each of these blips you see that they use the same quaternion. You do an avoiding scheme. The combination that produces the largest number of matches is the correct one. So you are trying to identify, which star is which and identify your attitude in the process. All of this is mathematics of almost high school [laughs].

How old is this locating algorithm?

MsM Very old. People have been playing with variations of this since the '60s. Some versions are more clever than others. This one for instance cheated because we used the Sun. 'I'm in three dimensions and I don't know where I am. By looking at the Sun I know two dimensions.' Without using the Sun, this problem is called the 'Lost in Space' problem. Imagine that you get a camera and take a picture of the stars. And only the stars without looking at the Sun. Figure out what the attitude of the camera was. It's more tricky when you only use the stars. We have a sun sensor and so we always have the Sun. So why not use it.

If you go to Google you'll find millions of references to this type of algorithm. But the thing needs to work, be well calibrated and all that crap, okay. So it's always a pain in the ass. Imagine now that I did the star identification – in the case of 'Lost in Space' you take a picture and you have twenty or thirty stars. Suppose you know all of them. So you have twenty or thirty measurements of attitude, how do you statistically combine them? There is one algorithm that somebody came up with that is pretty clever. I can't remember the name of it. If you look up 'Lost in Space' you'll probably find it. They try to find the eigenvectors and do decomposition. Somebody solved it, and that's it. So that's the sophistication that we use. During EDL we have again sensor fusion problems. You know what sensor fusion is? It's when you try to fuse data from multiple sensors to try to get the best estimate. So during EDL for example we have an inertia measurement unit and we have a radar to measure attitude and velocity. Essentially what we use there is pretty much least squares. That's all we use. Actually we don't even have sensor fusion there because we use the least amount of sensors. We need the radar to measure our attitude and velocity to time when to inflate the airbags... excuse me when to fire the rockets. You need to time into within a tenth of a second. Because you're coming at about seventy meters per second so in a tenth of a second it's seven meters of error in altitude. You need to estimate what your altitude is and what your velocity is. What's even worse is that I'm flying on uneven terrain, so sometimes I measure the altitude with respect to here and sometimes with respect to there. How do I trust all the measurements, they are noisy and you are coming very fast. The basic technique here is least squares. In least squares you draw a line and you look at some points, but some of these points could be totally bullshit. They could be total lies. So you look at the largest errors and you remove those points and then you do another least squares. So you do a sequential technique to weed out... it's kind of like voting between all these measurements. You try to reduce the residue, which decides the quality of the fit. Some of the points are liars we call them, mathematically obviously. The threshold that you use is a function of the altitude because, you see, the higher the altitude the more noisy this thing is. So we have a simple linear function that says 'at this altitude you use this threshold to eliminate the liars'. As you get closer to the ground you tighten it up. After several recursions of removing the liars, you fit again and then you stop because you don't want something that converges in infinite time because then you crash. So this is a sort of thing you have to do in flight. So then you check the quality of the fit. If the fit is not good quality, if it doesn't look like a straight line... I know that I'm coming down with a velocity between fifty meters per second and ninety meters per second, so if my velocity comes out outside these bounds we don't trust them. No estimate this time. So a lot of this is not necessarily very sophisticated mathematics, but you need to validate your estimate because you are not there to say 'uh oh it doesn't smell good' and you cannot put your threshold too tight because then you might never get a solution.

Is this something you work on every time you do a mission or could you just implement it in software and then upload the same program?

MsM Well, it depends on the type of problem. For example this thing I was telling you about the stars. It actually was developed for the Mars Pathfinder Project. MER did an identical spacecraft to get us all the way to Mars. It was a spinner using the same sensors, the same hardware, so yeah we actually did the same code and we fixed the things we didn't like, you know. And we flew it. The radar algorithm was the same in as Mars Pathfinder but I rewrote it because I didn't like the way we had done it. So now I feel happy about this thing, but now we are going to land with a different kind of landing system, now with airbags, so that requires a totally different type of sensor. The Kalman filtering techniques, least squares, fits, you know, they are all in the same bag of tricks. In this new MSL mission [Mars Science Laboratory], the new Mars lander is getting more mathematically sophisticated in two places. First we are actually flying the spacecraft through this entry phase. Instead of being a ballistic trajectory... are you familiar with this... you put a center of mass offset on the spacecraft so instead of being perfectly aligned with the wind the spacecraft is going to have a little tilt with the wind. So when you go in your car, instead of holding your hand in the wind like this [gestures] you hold it like this and you get a lift.

You use that lift for two things. One is to be able to be at a higher altitude when you open the parachute, that's a good thing. The other is... by rotating the spacecraft with thrusters and modulating where your lift vector is you can do two things. First of all move this way [points] so you can hit the target. Also if you are overshooting the target you reduce your lift so you drop faster into the atmosphere to get to a denser atmosphere and slow down, and vice versa. You see that you are falling short so you increase your lift and stay longer in the lower density. This was, believe it or not, designed in the '60s for the Apollo project. The Apollo spacecraft flew automatically using... calculus of variations I guess. Your typical limitation problem I guess. That's how they formulated it, you know with a cost function at the end trying to minimize the errors at this point. Then they do the adjoint equations, with the lambdas, are you familiar with that? It's an optimization problem. That's a sophisticated control law, compared to our typical thing. The other thing is, once you get to a certain altitude you need to fire the thrusters. Once you light the engines in the end, you just want to kill the velocity and land wherever you are. You don't look. You just land. Now we want to add what we call pinpoint landing, which is landing within a hundred meters of a certain place. Because there may be some important scientific target for example a thermal, with water coming out. Or a very dangerous place where you only have a hundred meters to land. So we have an algorithm that looks down at craters... it's like a star measuring thing. Instead of stars it looks at craters. And identify the craters by diameter and location with respect to the other craters. So then you can figure out where your spacecraft is. It's an imaging process algorithm. So now you know where you are and you fire your thrusters and say: 'I want to land there'. So there is an optimal control problem of how to get there without using all the fuel. There is one trajectory that minimizes the fuel. People are working on that right now.

[William Folkner arrives.]

MsM Hello. How are you doing? I'm Miguel san Martin.

WF Hi Miguel. I'm Bill... Folkner.

MsM I hope they are not disappointed, because we don't use any mathematics...

WF We do. But not in our day to day lives. It's all built in.

MsM That's right.

You were saying that there are three stages of the mission.

MsM Yes. There are the cruise, EDL and the surface operation. The surface... I'm from the guidance and control system... is also very basic and we are using tricks that, you know, sailors have been using for centuries. We look at the Sun. Actually this time to compute where north is... there is no magnetic field on Mars, you know, appreciable magnetic field, not very good for engineering purposes. And you need to know where north is because you need to point the high gain antenna to Earth. So we have a model of Mars onboard the spacecraft to know what the Mars orientation is at all times and to know where the Sun and Earth is. But we need to know where the spacecraft is on the surface. We have an accelerometer that measures the gravity. That gives you two axis, but you don't know where north is. So there are two methods. One is, okay I search for the Sun and knowing all the geometry you can figure out where north is...

WF ... because you know what time it is.

MsM Exactly. We have the time.

Both The old navigation.

MsM If you know the time, you can use the Sun to figure out everything you need to know. The problem with this is that to get high accuracy the alignment of the sun sensors and the tilt sensor need to be very good. The other problem is that at noon you're screwed. At noontime you get singularities because the Sun gets close to vertical. That's a bummer. So I went to the operation guys and said: 'Listen, is it okay that we around noon, plus minus one hour, can not do an attitude fix?' And he said: 'No, absolutely not. You have to figure out a way that we can always figure out from scratch what out attitude is.' Okay, fine. What we ended up doing was instead of using the Sun and the gravity vector; you use the Sun now, and the Sun later. So essentially you look at the Sun for ten minutes. We don't move anything; we leave the camera alone. From that you

can figure out all axes at any time of the day. So there are no more singularity problems and it's all done in one single instrument. In ten minutes worth of measurements you know enough to find a big angle, so it's a quasi singularity right there, but because it's all within the same instrument I can tolerate a much smaller angle between the two things. You see what I am saying? So there may be a degree error between the Sun and the accelerometer, so if you are very close then one degree can give you a large twist. So that's it. Not much. The most important is that you have millions of these little, simple things [laughs]. And that's the trick; to make them all work, and talk to each other and make sure that no parameter is tightened too much or too little. The complexity of the space problem is keeping it simple. Dealing with a whole bunch of little simple problems. You know, once in a while you get a nice juicy problem. We are getting them now for the MSL, where we are getting more sophisticated mathematical problems to guidance. Optimal control problems through the hyper sonic phase.

WF Well, you hit a lot of mathematics in your descriptions, right, because you need to know the positions of the axes of Mars around the Sun as a function of time and you need to know what the orbits around the Sun and the Earth were. There is a lot of mathematics hidden in what you just said.

MsM Hidden! They are hidden!

WF They are hidden from us. We've worked all that out for us in the tables. So to know where the Sun is now, you just look it up. Somebody had to figure it out the first time.

MsM So we are resting on the shoulders of giants. That's what it is. Newton, Kepler. These guys are there.

You said there were three stages. Is there also a forth stage that is actually the first, a pre-stage that involves computer simulations?

MsM The three stages are the operation stages of the mission itself. The project itself has several phases; design phase, testing phases and then you have the actual mission operations phase.

So the test phase would be where the computer simulations would be?

MsM That's correct, but we continue to run these simulations all through the operations phase. Statistical programs that figured out what the probability of a successful landing on Mars was. We loaded the last version of the flight software three weeks before we landed, that's when we finished it [laughs]. It ended up being the right thing to do. But you try to avoid that.

If you actually send software when it's in flight, the channel coding you use must be very reliable, right? You can't accept any errors in software programs.

MsM That is correct. This is not my field of expertise, but we use error checking, like check sums. The software we send to the spacecraft is a series of files. You do checksums and if the checksums are bad, you'll retransmit. If you know that this image is right then there is the process of building the flight software onboard because the pieces need to be rebuild. That in itself has some error checking. Once all of those checks are done then you do the equivalent of Control+Alt+Delete. Up to that moment you are still on the old versions of the software. Then you tell the spacecraft: 'Go load the new software'. The hardware is designed so that when it reboots, if it doesn't get to a certain point it tries going through all the different images of the flight software and eventually it will go through the one that is working, in theory. So you always try to give yourself a way out. If you screw up, and the flight software that you put in is wrong the spacecraft is stuck. We even have the capability to send to the radio a command to do the Control+Alt+Delete without the software in command. But that means that the radio needs to be on, which means that if the software turns it off then you are screwed. You need to build all of this ahead of time and think about all of the ways that you might shoot yourself in the foot. And you may think you have it all covered but you always find something.

WF Were you around for the Viking Lander 1?

MsM No.

WF Well the Viking Lander 1 had a design life of about thirty days, but it worked for six years and it finally stopped working when they incorrectly sent the command to point the antenna down. After that they could not send any more commands. Otherwise it would still be working today.

MsM We had a similar problem on Pathfinder, I mean again it was supposed to last a month and lasted three months. We had very little intelligent software on the surface. In part because we only needed it to last a month, you know? These projects are actually excellent exercises in game theory. Really. Because Washington tells you: 'I'll give you seventy points if you land safely and you take one picture. Then I'll give you ten points if you last one month and then I'll give you ten more points if you last three more months. That's how they gauge you and they say the mission was successful or it was not a big success. So guess what. When they told us: 'It's not that important to last long on the surface', we didn't put so much effort into it. So the conclusion was that it was a very dumb system that didn't have the capability to call us back. It was command driven from the ground and didn't have the persistence. We added that persistence in MER. We call it communication behavior. The spacecraft always tries to contact us, but even there we found... we almost lost it too. On Pathfinder it was a runaway thermal condition. We turned the thing on in the middle of the night to do a science observation...

WF The battery couldn't be recharged. It could only be discharged thirty times.

MsM That's correct, but we had a way to actually do the mission without the battery.

WF Except it didn't know what time it was. So it didn't know when to listen, and it lost the capability to listen to Earth at the right time.

MsM That's correct, but we never came up with a plan for transition into that mode of operation. We should have said: 'Today we'll start the 'no-battery mission' and these are the procedures to handle that.' But we didn't do that. To do science at night you need a battery, obviously. Anyway I have to get going. It was nice to meet you. I hope it helped.

Very much so.

[Miguel san Martin leaves]

WF There is a lot of really good work going on here. But like I said, there is so much mathematics hidden, because it is worked out once and then we go on to a new problem. We don't think about it.

It seems that a lot of the mathematics you use is in software packages and so on.

WF Yes. There are many different software packages. When we start to build a new antenna we build new software pointing it and we have to resynchronize the time and compare the new program to the old program and then we go back into the mathematics, but we don't do that very often. So you can say 'well you just take two pictures of the Sun and you know where you are'. Yes, but... that's assuming somebody told you all the tables of where the Sun is and there is a lot of mathematics in that.

Yeah. And in taking the picture too.

WF In taking the picture. I don't know about the mathematics of taking the picture.

But it is digital, so there is bound be mathematics in it.

WF Then there is the mathematics of programming it. You might not consider this mathematics, but it is. Turing machines and all that.

So what are the plans for the rest of the day and tomorrow?

WF So right now nobody is planning to talk to you ever again [laughs].

[...]

We would like to know something about the trajectory from Earth to Mars. There must be a lot of mathematics in that too.

WF Again a lot of it is hidden in the programs. I know the rudiments of picking a trajectory. If the Sun is here, the Earth is here and Mars is here [draws] it turns out that you always want to launch here [points] into an orbit that ends up there when Mars is there [points]. That's the minimum energy ellipse. That will get you from there to there. This is never physically realizable because this works only if everything is in a plane. So because the planes of the orbits are tilted it usually turns out there are various local minima in the energy. I used to have a book full of these things, which I don't anymore. There are sort of two local minima, which we usually call a type 1 and a type 2. One gets to Mars there and one gets to Mars there [points], and depending on the relative inclinations one of

them might be slightly lower energy than the other. There is typically a saddle here. For this amount of energy [points], which is a lot more than that amount of energy, it's roughly independent. But then, if you are willing to wait another orbit you get into a type 3 and a type 4. So for Mars type 1 is a seven months cruise time and type 2 is ten to eleven months, I don't remember exactly. There are these contours and so you can actually launch on this day and arrive on that day [points] and you can launch on a later day and arrive earlier, if you are willing to pay that energy difference. But these days are not far apart. Typically they are on the same type of trajectory. For MER-A and MER-B one of them was launched later. And the later was on a bigger rocket, so it could launch with more energy, but I don't know if that ended up landing first or second. In principle it could have landed either before or after for the same amount of energy. So you would choose depending on lots of different things. There are a million different factors that go into choosing launch dates and arrival dates. It turns out most of JPL's missions arrive on July 4th, because that's the national holiday. Sometimes it must be deliberately chosen [laughs]. It may not be a very significant energy difference when choosing between these contours, so you might as well choose July 4th, because it sounds patriotic.

Is it true that nowadays you tend to rely more on the trajectories and try to use less brake fuel?

WF When I plotted here [points] – the trajectory guy calls this C-Free, this is an amount of energy I have to burn at the launch. There is a corresponding plot for arrival date and departure date [points] that is usually called V-infinity, which is sort of the inverse. How much fuel do you have to spend to go into orbit when you get there? That's a more varied problem because when you go to Mars you have a lot of choices of whether you want to aim low of aim high. If you aim low, you have to spend a lot of energy to get into the orbit. If you aim high, you don't have to spend that much, but you'll get into a very long orbit, which is very unstable, maybe. It's a little trickier because it depends on what orbit inclination you want to get into and what orbit period you want to get into, altitude, how fast you want to do your first correction maneuver. Many, many more parameters.

But you would think though, that the less fuel needed for braking the craft would mean that you could have a heavier load.

WF That's true. Something important that has happened in the last few years is... there is an atmosphere on Mars... and we've gotten into this habit of aero-braking. It was there for the first time on the Magellan spacecraft at Venus, okay? Usually when you do a maneuver at apoapsis... are you familiar with the term apoapsis and periapsis? The closest point to the planet is the periapsis and apoapsis is... anyway this far point. If you do a maneuver here you mainly affect here [points]. So they go into orbit and they do a small maneuver here to lower the periapsis so it will fly through the atmosphere and slow down. The slowing down mainly affects the apoapsis. So having slowed down, it will come into here [points] and then there, and there, and it will get into a circular orbit without carrying the fuel because they are using the atmosphere to do the braking. It's a very real effect. For Magellan it was not designed to do this and they just tried it to see if it would work or not. Mars Global Surveyor, for example, had a spacecraft that looked like that; the solar panels were actually angled like this [gestures] so it would be more aerodynamically stable. Once they've done enough braking and get into near circular, they do a little maneuver that raises the periapsis, so it will come back up and stay above the atmosphere. Then they can change the angle of the solar panels to get more power, because for this angle you don't get so much power. But you don't need the power if your instruments aren't on. There are millions of things going on. So aero-braking is on thing to avoid carrying this fuel. Another thing people have talked about is aero-capture. When you fly to Mars and hit the atmosphere for the first time, you'll try to get a big drag to get the stopping fuel loss entirely from atmospheric drag. It's a very delicate maneuver. You have to be very precise and you have to get a lot of stopping to go into orbit. So this is C-Free again [points to curve]. C-Free has units of kilometers squared per second squared, so it's the square of the velocity. It is called C-Free because it is supposed to take out the constant part of the energy needed to go from Earth to surface orbit. So a launch to Mars will usually have a C-Free of about five or about two kilometers per second

relative to the Earth's orbital velocity. There is a seven kilometers per second burn needed to get from the surface of the Earth into orbit. It usually ends up with two kilometers per second to get to Mars, but sometimes it can be as low as one, sometimes the best answer at any given time may be fourteen or fifteen depending on the inclinations. At the other end, you have the same problem. You're coming into Mars at a pretty high velocity and you go into the orbit of Mars. So we usually have a V-infinity of about two kilometers a second. When we do aero-braking we do two kilometers a second chemical propulsion to get into a higher orbit and then every time through the atmosphere we get a couple of hundred meters per second and so we can accumulate them slowly to build up to a couple of kilometers per second. To do aero-capture you have to get a couple of kilometers per second that one time to go into orbit. That means you have to aim very low in the atmosphere because Mars doesn't have that much atmosphere. You are coming at this very high velocity. When you do the orbital velocity you are not coming in nearly as fast as when you are coming in from another planet, okay? So aero-capture is very delicate, it hasn't been done yet. I'm sure the Phoenix mission will not do it. That's 2007. The Mars Science Laboratory, MSL, which is launching in 2011, I don't think they will do aero-capture either. I'm guessing 2015 or something like that, because then you'll want to have a lot of mass because you want to bring a sample back. You have to fire a rocket to get out of Mars and come back to Earth. So there it'll probably be advantageous in cost and weight and everything.

How much mathematics is involved in the parachutes?

WF There is a lot and a little. There is more in this problem [aero-braking] actually than in the parachutes. I'm not an expert on any of this. Now the parachutes, I remember seeing a television show where they talked about the parachutes for MER. The parachute for MER... when you look from the top it has a hole in the middle. In the original the hole was too big and the parachutes would not inflate. So that's a very complicated problem physically, but I think people more or less do it trial and error. I'm actually working with somebody now who works a great deal with parachutes. I could ask him, if you could talk to him.

We were thinking that there might be some solving of differential equations...

WF For the heat shields they do a lot of finite element, because they have these very high temperatures and pressures at the points of entry, so there are people who do that to try and find out how thick to make this, and what material to make it out of, whether to make it dense or not so dense. So I know a lot of people do that, but people I know who do that are at NASA Ames [Ames Research Center]. I can try to find out if there is anybody here who does heat shields. It's not that the mathematics of the process is so complicated, it's because it's so nonlinear that they can only be solved iteratively on a computer base. It's like predicting the weather, and predicting the weather is not very hard in some sense. You know there is a high pressure offshore and it'll keep the air dry here and not pump pressure in from the south... You know, it's pretty simple, but to try to predict where that high is going to be two days from now is a very hard problem. So I think this is something like that. You know the heat shield is going to get hot and you know the physics of when the atmosphere strikes it such and such, but it is just so complicated that you have to use very elaborate computer models. It's not that the mathematics is so complicated.

Okay. We would like to talk about some more general things. Miguel san Martin was talking about three stages of the operation. Flight, landing and surface. Before we came here we tried to look at the stages to find out what mathematics were in them, to get an overall view of the mission. Communication would be all over, right?

WF Yes.

Here [*surface*] *would be compression of images and channel coding. The channel coding would also be here* [*cruise*] *since they are sending software to the rovers while they are in flight.*

WF For Mars, the landing is two minutes, so all the mathematics of the parachutes and the heat shield go into those two minutes.

How about the airbags, is there any mathematics in that you think?

WF There is a little bit. The airbag you make a certain size of a certain material and you have a certain mass on top, and you hit the surface at such and such

velocity. For the MSL mission they thought a lot about whether they could use airbags or not, because when you get to a big enough mass for a given velocity and such you may or may not burst the material. That's about the extent of the mathematics for the airbags. The entry-descent-landing is a very complex stage. Not so much in communications, but it is coming in at a very high velocity and so you have to deal with aerodynamics of a very high velocity entry. So in a very nonlinear supersonic atmosphere regime you have to deal with the properties of this material at very high pressure and very high temperature. So the mathematics is a little bit unusual. You have to deal with the airflow around this thing. It carries accelerometers onboard to know when it is speeding up or slowing down and at the appropriate time it deploys the parachute and then they have rocket motors they fire and radars bouncing a signal back and forth to see what the surface is like. Then they cut of the rockets when they are near the surface, so it is a very complex mathematical thing, mostly aerodynamics at high Mach numbers. So it is very aerodynamical and very much a control problem. I don't know if you've talked to anyone about the control theory, that's a whole branch of mathematics in this. There is an awful lot of real time control theory going on during those two minutes.

But there is also control theory involved in the rovers.

WF Not the same type of thing. When I think of control theory I... have you ever heard of Kalman filtering?

Yeah. Miguel san Martin talked about it.

[Temporary interruption. Going outside.]

WF It's much nicer outside anyway.

WF So control theory isn't so much in this except for the spacecraft when it is on its way to Mars. Two weeks out we get a radio measurement and so we use this control theory for the navigation process as well, but it is not so much a real time thing, because we have hours and days to reprocess the data to make a decision about where to go. There is mathematics in everything. There is control theory, aerodynamics, orbital dynamics, Newtonian gravity, bodies going around the Sun. We use general relativity, that's mathematics of physics but it is curved space, right? So we have to get the curvature of space modelled right when we analyze the radio signals back and forth to figure out where the spacecraft is compared to where it is supposed to be for going to Mars. It is not so much a factor once it's on the surface because there you mostly care about the direction to the mountains, right. That's not so much a general relativistic problem, but if you want to know the light time to Mars to the nanosecond then that is...

WF We don't do any map theory for instance. Four color map problem. You know that? We don't do any of that. We don't really use any abstract algebra, group theory and that sort.

Except in the channel coding.

WF They use abstract algebra and group theory in that?

The Reed-Solomon codes are based on Galois Fields.

WF That's news to me. I didn't know that, all right, interesting. Linear algebra is a field of mathematics we use all the time. Matrices. That's in the control theory all the time. There is Riemannian geometry in the general relativity. Calculus. I don't know, are you guys math majors?

Yeah.

WF So you know more fields of mathematics than I do.

What about the surface? You said that the control theory is not very...

WF It's different. You really should ask Jake [Jacob Matijevic] about it. They take pictures of where the Sun is, they take a picture of where the rocks are and they chart a path that goes in between them, but I don't know if they just do it by trial and error or if they have some kind of mathematical theory. For the rovers you pretty much say: 'I'm here. Oh, I would like to drive to that bench over there [points], but this is in the way and this is in the way, but it looks pretty flat over there, so why don't I drive towards that trash can.' So it looks pretty flat, and it'll just drive over there for a day taking pictures all along and when I get there I think I know how far I've gone, but maybe I'm not sure. So I look

all around again and I say: 'Oh, the bench is over there now and there is a ramp down the sidewalk that I can drive to get towards the bench.' I don't know if that's a very mathematical process. It could be but it just doesn't seem like it. They measure the wheel turns and I think they can measure the force on the wheel, but they can't tell whether the wheels really turn in place or whether it goes forward. But there are hints. From the amount of torque the wheel took to move they can find out whether it is turning freely or not. If they didn't take pictures they would still have some estimate of where they are, based just on the wheel turns, but usually for the type of thing they are doing it is better to just see if they are going in the right direction or not. The rovers have a little bit of autonomy. Like if they are going to the trash can and they start out but they end up being in the dirt here and the wheels are sliding to much, it might decide to go left a little bit and then right a bit to see if it can get a little better traction. But it won't go very far without an eminent action on the ground. It might go around one rock and spot a rock it hasn't seen before. I think there is some pattern matching software onboard so that if the rocks are there and that wasn't predicted it would make a course adjustment. I doubt that it happens very often. I think most of the course planning is done on Earth and I think it is mostly based on what they see and on how far they think they can go in that direction in that day. You should ask Jake that. I don't really know how much autonomous navigation they use.

He told us that the amount of artificial intelligence involved was very limited.

WF Very limited, yes. So I think if it could see a rock and it knew that if it turns left or right it'll be okay, it would do that. If it saw a rock and didn't know the best thing to do it would just stop and wait. So I don't think they are using very much. I don't think they really need to because it can only go in a day about as far as you can see. So I think almost all of the steering is done on the ground. So maybe finding out where it is from the Sun is the most complicated mathematics they are using in turns of surface operations. They also take these pictures with different spectra and try to figure out what the materials are. So there is mathematics involved in figuring out if this rock is the same they saw before. They saw a meteorite. I don't know if you know that. They were taking pictures one day and they saw a rock that looked different in the red and in the blue from most of the other rocks that they saw, and because of that they could figure out that it had more iron in it probably. They saw the pictures down here and said: 'That may be a meteorite, and not from Mars.' And because of that they decided to drive over there and investigate it with alpha proton X-ray spectrometer, are you familiar with that?

No. We are not physicists.

WF Oh. You are not physicists. Hah. Well this table here [knocks on table] is made out of atoms right? And so there is an instrument, alpha proton X-ray spectrometer that have radioactive sources that emit X-rays and gamma rays. So if an X-ray comes down here at the right wavelength it'll knock an electron out and you'll see a certain energy from that electron, or certain color if you like, and so this instrument basically can tell you what the composition of the rock is. But you have to be touching it almost to get enough signal to noise to determine that. So that is atomic physics. There is also something called a thermal spectrometer, which is a camera that looks at rocks both in the infrared and the visible. By the ratios of the thermal functions and the different colors they can kind of guess what type of rock it is. But it is not as precise as this [alpha proton X-ray spectrometer]. So they use this [thermal spectrometer] to look for interesting materials to go use this [alpha proton X-ray spectrometer] on. And they use the pictures themselves to kind of just guess. So this is a very powerful instrument in terms of picking out rocks. It was pretty interesting that they found a rock that fell from the sky on the surface of Mars. That is pretty neat. Most of the instrument people are not here, they are in Arizona and in Chicago and Germany.

Where in Arizona?

WF I think it is Tucson, but I'm not sure. Let us see. There are people looking at the pictures and trying to figure out what the hydrodynamics were that caused the flow of the rock or the water. But it's really not done as part of the operations. It's more interpretation of the pictures.

Is the quality of the pictures taken from the rovers good enough to do basic research? Or is it more of an eye catcher for the public?

WF Well, all of the pictures taken are good enough to do various things. For instance the second MER, when they took the picture, the first panorama, it caught a shiny rock in the picture. That was tremendously important, because in every other picture taken the surface of Mars has been covered with dust. So finding that flat shining thing existing at all was tremendously important scientifically. So that was the first thing they went over and looked at to get a picture of what the material was like under the dust. The pictures are very interesting for that. Albert [Haldemann] is kind of an expert on counting rocks of different sizes. You can infer what the atmosphere is, when the rocks fell, how old the surface is from the distribution of the sizes of the rocks in the pictures. So the general pictures are very important. It's hard to assign dollar value or euro value to different bits of information. So certainly there is a lot less science value per picture bit than per alpha proton X-ray spectrometer bit. But there is science value in all of them. And you have to have the pictures to get this [the rover] to where you want it to be. It's not like it's wasted. I thought about this myself. There was a CNES, French space agency, mission planned to put four landers on the surface of Mars at once, and the primary interest was to do seismology. But they were going to have a camera on all of them anyway. What good was the camera? Not very much. It sends nice pictures, but it really wasn't very interesting. Especially compared to what the rovers can do. From the pictures, you know you can tell whether it is a windy day or not. It just wasn't that interesting scientifically. Let's see. Morten Bo Madsen was he the one that sent you to me in the first place? He takes pictures of the dust on the magnets. So there is a lot you can do with the pictures. [...]

WF So we should probably hook up on Wednesday. Do you want to do that by telephone or do you want to come out here? Where are you guys staying at?

We're staying at the Pasadena Motor Inn.

WF I'm very sorry to hear that... No, it should be okay.

We're in room 504, but we'll probably try to call you.

WF We can go by the big spacecraft mockups on your way out. Let's do that.

4.6 Doktor William Folkner 16/3 2005

We have an overall frame on the thesis that has to do with the stagnation of science: For example physics. There haven't been large breakthroughs in pure physics since quantum theory and the theory of relativity. If we take biology, since the double helix there haven't been any great discoveries. We are looking at how it is with mathematics. We are mostly looking at coding theory and data compression. One thing we've learned from our trip in the US is that if you go to MIT and ask them about the stagnation of coding theory and data compression they more or less agree with you that these theories are in a period of stagnation, but if you go here the scientists you talk to totally disagree.

– It must depend on the type of the person you talk to – a lot. For instance all of the coding people you've talked to here probably deal with the Shannon limit and the Nyquist limit, right? And all of these codes are bumping up against these real physical limits. And so they talk about the turbo codes and the (14,2) versus the (7,2) [convolutional code] – they are all gaining small increments of performance – they are not gaining factors of ten in performance, so I think that in terms of coding theory they probably have reached the limit. But that's me talking. The radio people here would probably disagree. The next breakthrough in coding theory is going to have to come from quantum entanglement, which people are working on today but not for communication with Mars. People are trying to send information faster than the speed of light, people are trying to do quantum computing instead of the traditional one-zero computing, so there's a whole potential to do drastically different things than we do now in the practical sense in terms of coding theory. I'm not the right person to tell you definitively how that would go. But in terms of practical radio coding, let's see... probably the coding theory is more or less stagnant, but the technology for transmitting information clearly is not. Ten years ago this didn't exist [points to the MP3

recorder]. So the ability to store information has changed a lot, the ability to transmit information too. There is a lot of wide spectrum things going on, there's a lot of pulse spreading going on, there are many, many different things in terms of transmission of information going on, right? The Internet, for instance, using fibers to carry bits has just gone up several orders of magnitude in the last decade. We used to dial up to Australia and we were lucky to get ten thousand bits per second from anywhere in Australia. And now with one piece of glass fiber or a satellite link we can get gigabytes per second. It's tremendously different in terms of capabilities, but the fundamental coding theory is not very different, because we're still counting photons, and so we've reached the 'counting photons' limit. So I don't think you'll see a big breakthrough in coding theory until the quantum state information coding. I doubt anybody here a JPL works on that.

We were thinking that part of the explanation would be that at MIT they are working with pure math and pure science and people here would be more practical or applied and involved with technology.

– I can speak more for physics than I can for coding theory. I've said everything I know about coding theory. Quantum entanglement for coding theory, that's a breakthrough that will happen in your lifetimes, but probably not next year. So there's a breakthrough waiting to happen and there are people working on it. But in terms of physics, I mean quantum mechanics came from the '20s, general relativity came from the '20s, but it wasn't until the '70s that people understood how the weak force and the electrical force could possibly be described by a common mathematical framework, and then later in the '80s this was extended to include the nuclear force, so the fundamental understanding of those three fundamental physical properties has expanded greatly in the last thirty years. Now the last ten years I don't think a lot of progress has been made. But that's something sort of inherent... how can I say this... the way we look at the world both in terms of biology, the stars, physics and stuff like that is limited to the type of measurements we can make, and so we're approaching the limits of physics where to find out how gravity will cope in with the other forces we have to accelerate photons together with forces a million times bigger than we can now, and it's not that there is a fundamental physical limit that is preventing us from doing that, but you know it might take more energy than the output of the sun to actually do it. So there are people working on ways of accelerating particles together more efficiently so we can get these measurements and find out what happens at these higher energies, but progress hasn't been very rapid because of the photons again. So there are ideas about using light to accelerate particles and things like that so there is a lot of work going on but not a lot of practical progress has been made. Because there has been no practical progress made in that type of measurements it makes it hard for us to test our theories like where they end and probe the boundaries. So the fundamental understanding of physics has been slowing down in the last twenty years. I would not say in the last hundred years. In the last twenty years there has been slowdown because we can't make photons collide hard enough. There has been tremendous growth throughout my school age. The understanding of how the various forces interact. So the level of sophistication of physics in terms of describing the mathematics of the world and the universe grew greatly through the '70s and the '80s because we became able to measure these things at a much more fine level. Now the mathematics of describing that are so complex compared to quantum theory that it is much harder to convey to the general public. And it doesn't apply to very much that we see in our normal everyday lives, so there is not a strong emphasis on learning it, okay? But a lot is still happening in physics. So we're at a limit now where... the foreign administrator said this and I think he's right, which is 'that to really probe the limits of the universe we now have to try to look into the big bang rather than try to do it in the laboratory'. And that's very expensive too, but at least we can get energies there that we can't get on the Earth. So the most astounding discovery in the last hundred years, I think, in physics or astronomy is the idea that the universe is expanding at an ever faster rate. That discovery was made in about the year 2000, and we are still stunned, we had no idea this was the case. So there is a fundamental physical discovery there that we had no concept of describing. So we are not done. We are not stagnating. Things are being found out all of the time. In fact enormous things are being found out.

But how about the fundamental theories, have they changed at all?

– So they changed a lot through the '70s and '80s because ever more complex mathematical structures had to be developed to describe how these different physical forces interact at different energy levels, which the measurements were there to support as we made ever bigger particle accelerators. The sophistication of the mathematical theories describing the interaction of matter advanced tremendously between 1960 and today. In the last ten years it hasn't advanced too much. There are people trying really hard, but they have no measurements to pin down really. There are physical theorists working on things like string theories, on manifold theories and trying to come up with the physics and a mathematical structure that will encompass not only the complicated mathematics that describes the forces as we understand them now but also extrapolate to gravity, okay? So there are people working on that. We're trying to describe things that (a) we can't measure easily and (b) that are very complex mathematically. And so a lot of these physics theories including string theories make it difficult to make a computation that arrives at a result that you can then go and measure. In quantum mechanics there is a problem of renormalization – are you familiar with that concept? Basically if you have two point particles like two electrons and you park them arbitrarily close together then the forces that's attracting them is infinite, because the force goes like 1 over r square, and so it becomes infinitely large. When we start doing this quantum mechanically we have all of these infinities and it was not known how to deal with these infinities until the '60s when Feynman and a few other came up with renormalization where they tried to say: 'There is this infinity and that infinity, but the difference is somewhat small and we can measure that in the laboratory'. But before that there were just a lot of infinities. Back in the eighteen century there was a similar problem called the ultraviolet catastrophe which is – if you thought of light being quantized then you can have higher and higher and higher frequencies which have infinite energy, so you should have an infinite amount of energy everywhere. That was a big problem in physics, until people said 'well it has to be confined'. So quantum mechanics basically resolved this ultraviolet catastrophe. Renormalization resolved the ability to calculate these infinite things to come up with finite answers that we could analyze in the laboratory. This came in the sixties. So the mathematics has grown much more sophisticated since then. But I'm an experimentalist so I haven't followed all of the mathematics, but the theories that are being used now to try to combine more things are more mathematically inclined... so physics – the easy stuff has been done. We know when something falls it hits the ground.

But you still don't know why...

– Gee, that's a harder question. You can always push it back: Why, why, why... So if you assume that mass exist and it deforms space then that explains mathematically a lot of measurements that we make. But where did the space come from and where did the mass come from? I don't know if you can ever go back to nothing and come back with something. So in physics at least we try to concentrate on what we can measure and what we can calculate. So every theory we come up with should explain all the measurements we made before – and to be treated really seriously it should predict all the measurements we haven't made yet. Then we see if that happens or not. So there are a lot of experiments that people would like to do, but they are fairly expensive. The equivalence principle, are you familiar with that? The equivalence principle is something Einstein came up with which basically said that all matter – there is the strong and the weak equivalence principle – should fall the same in the curved space time. So the matter should not interact with the curved space. So people are testing that all of the time. There are laser measurements between here and the moon that are made all of the time and they are constraining the possible difference more and more. But at some level they should be different. Because at some energy you would expect to find a difference between the strong type of nuclear force and the weak type of nuclear force and how they interact with gravity. Right now we're in the low energy limit. We either need more accuracy or higher energy. So there are a lot of experiments people are looking at to understand what really drives gravity. Dark matter is another good one. In clusters of galaxies, galaxies orbit each other and they orbit each other as if there were a lot more mass than we can see. So what is that mass due to – we don't know. So that's

the dark matter question. But the accelerating expansion is just blowing everybody's socks off. So there is a lot of stuff left out there to be found out. Now biology – you should probably talk to a biologist – but I'm in fact astounded by the progress made in the last ten or twenty years. Do you guys know much about proteins?

No, we don't know much about biology actually...

– Well, okay. I'm actually interested in it, but I don't know a lot about it. But in biology you have cell walls, and the way information is transmitted from the outside of the cell to the inside of the cell is through proteins. I'm way beyond my realm of competence here, but I'm just trying to give you a hint of what to look at. Different molecules come floating by, and some of them can bind to the end of this molecule [on the outside of the cell] and others will not because they are the wrong shape. If you actually looked at the arrangements of the molecules on these complex chains they form a physical shape. So for instance the benzene ring floating by there will not fit in. But it turns out these molecules that form cell walls – I think they're called lipids – and certain molecules can come and fit in the space there. When they come they can kind of bend in the molecules a bit there and it causes the other end of the molecules inside the cell to bend out a little bit maybe, okay? And so it turns out that understanding how a cell works is down to the point: What kind of combination of atoms forms a molecule that has the right shape to fit into this lock if you like, and cause this reaction. That's so astounding compared to what we knew thirty years ago about biology, which is if you took a blue-eyed person and a brown-eyed person you probably got a brown-eyed child. I mean, can you tell the difference there?. It's tremendous. So in physics one of the sidelights now is figuring out how to design a molecule given a set of atoms that has the appropriate shape. So for a molecule with two hundred different atoms, how do you hook them together so they are stable and has the right shape. It's called the protein-folding problem. Another thing is the genome. The genome is a long complex molecule and for the human it's all been sequenced. We know where every molecule is for the average human being. Every single molecule. We don't know how to interpret it yet, but we know that certain groups of molecules along that chain encode certain types of proteins – we know that. And so we can design molecules that will in fact cut those chains at certain points. And add certain new sequences. It's astonishing. How can you say biology is stagnating? It's only just starting, okay? Biology now is almost where physics was in the nineteen hundreds. We now have the tools to understand the basic building blocks of life, how to manipulate it. So in the next twenty years we'll see an enormous change. We'll be able to design people that are seven foot four, sorry two meters and six or something, with a certain body structure. We're almost able to start that. And that's because we have the tools to actually look at the individual atoms of the cell now, which we couldn't twenty years ago. So biology is just starting. It wasn't even a science twenty years ago in my view because it wasn't quantitated. You couldn't predict the mathematical output of some mathematical input. And now we can. We can say: 'If we design this molecule like this it will fit in the cell there and cause this reaction and give this chemical out'. It's incredible.

How about the space missions?

– Okay, space is harder. Because space is up against – not physics – but chemistry. We have to get mass into space, and it costs energy to get mass into space, and velocity to get mass into space, which means that for everything that goes up we have to send something down. So there has been no fundamental advancement in the physics or the chemistry if you like for sending things into space for forty years probably. In fact the NASA space agency, well they've done a number of silly things in my view over the last couple of decades where they would spend a billion dollars on this technology and 'oh it didn't work, let's try another one'. So over time they kind of went from this to that to the other. Single-staged orbit, hypersonic flight – a whole bunch of different things. And now people say: 'Suppose we go back and do the Apollo launch vehicle again'. 'Because it's cheaper. It's money that is limiting our ability to put things into space. As is we are doing more and more things in space, we are discovering things we never knew existed before; we are landing rovers on Mars and picking up meteorites from other planets. We're doing a tremendous amount. But to make a quantum leap if you like in space flight technology... it's hard. There is the

fundamental chemistry of burning things and injecting them. There is an idea around that seems like a good practical answer. It's difficult, but it looks better than sticking with just burning hydrogen and oxygen forever, and that is the idea of an elevator. Have you ever heard of the space elevator? There are people working on molecules that are strong enough to build a space elevator. We know that it's possible. We know that a single crystal diamond is strong enough. So if we can build a single crystal diamond one meter in diameter and 300 kilometers long then we have a space elevator. It's physically possible but we don't know how to build that yet. That's the only way I can think of to make space flight cheap enough to talk about sending hundreds of people to colonize Mars. We can send somebody to Mars five years from now and bring them back, but it would cost a huge amount of money, and to send a hundred people to Mars, I don't think we could practically afford that until we get something like this. So we're limited by our chemistry at that point. But our ability to understand what's going on in the universe outside is growing much more quickly than that. There's a mission here called Terrestrial Planet Finder under development, which is being designed to be able to take pictures of planets around other stars and measure the properties of their atmospheres to see if they have water vapor in them. That's something we couldn't have imagined to do thirty years ago. And it still might be twenty years off. But in fifty years we'll have pictures of other planets around other stars and see if there are oceans and continents there. So the golden age of space exploration is just started. But with space colonization we're stuck.

But how about the planetary missions?

– I was surprised by MER how much more information you get by moving around. By being able to find bedrock; by being able to find a meteorite. It's tremendously more powerful than I ever expected, so I've been very pleasantly surprised by that. But in the end, well NASA right now seems to be dominated by the search for life. I think that's because they think the public is more interested in that, and that makes some sense because the public is paying the bills. So there is a lot of emphasis from NASA on searching for life. People are trying to build machines to detect life on Mars, but I don't know how much progress they've made. It's probably cheaper to get a piece of Mars and bring it back here. We did that with the moon in the late '60s and we ought to be able to do that with Mars we just haven't done it yet. And that will tell us whether we have life on Mars or not. Now, probably there isn't, but I used to think it wasn't very important until a biologist told me: 'Suppose you have only one type of writing and that's the only thing you've ever seen written. It would be hard to imagine say computer disc encodings – it's just totally foreign to you.' And so we have one type of DNA on our planet. If we could find a second one to compare it to that would be tremendously powerful, right? So there is a big science payoff if we could find a different type of life. I don't think we'll find it on Mars, I would say Europa is more likely, cause it probably has water underneath it. Water is a very peculiar material in terms of dissolving things, and expanding when it freezes, you know, everything else gets smaller. It's got enormously strange properties. So it is reasonably likely that life needs water to exist, but we don't know because we haven't found it anywhere. So I would say in fifteen years we have a sample back from Mars and will know if there is life on Mars. But if we find it, it will open up whole new classes of things. But we probably will not. Maybe Europa or maybe even Titan. We just found out two months ago that Titan has mud on its surface. It's got hydrocarbons on it surface, carbon oxygen. We know that now. We guessed before and now we know. So we're finding out things about our solar system. I would say once we get back samples from Europa, Mars and Titan and find no life we would largely be done, right? But that's fifty years off. In the meantime it would be nice to start colonizing Mars, but that depends on the space elevator. In the meantime we can try to learn about planets in other star systems and see if we can find life on them. We will not be able to tell whether it's different from ours probably in a hundred years, but we might be able to find out if life exists in another solar system by a hundred years from now. So I think space science is exponentially growing in terms of possibilities. Science is not stagnating. It's not.

There is a lot of publicity about the Mars mission and also President Bush has...

– His exploration issues?

Yeah. Is there any effect on your work? Do you get more funding? Does it make more things possible?

– It's not known yet really. He announced this about a year ago, maybe two years ago. The bosses at NASA haven't decided what to do about it yet. The only thing that is clear cut I think – but you're really talking to too low a level manager here, you're only getting my scientist view of things – he's decided that the space station will be completed to keep agreements with the international partners which is a good thing, a stretch from him, but that it's not a good science lab, which again is a very good conclusion. I think the space station is worthwhile for having people learn how it is to live in space. But it is far more expensive to have an astronaut do an experiment than for us to launch an unmanned rocket to do the experiment, far more expensive. So if you want to do science you're better off not doing it on a space station. So those things are limiting and that's good. Let's stop wasting our money there. And the decision not to use the space shuttle any longer than the space station is also a good decision. Because it's clear they're reaching the end of their life. And it probably wasn't a very good decision in the first place. So, so far we're on good ground. But then he says we going to design something new to take men to the Moon and Mars 'and by the way, we're only going to spend one hundredth of the Apollo program on it'. So that's a hard one to buy. So NASA is trying to figure out how to do something practical for the amount of money that has been allocated. I think for two or three billions a year which is about how much they have to play with, they could probably build a capsule that can take humans to the space station or that could take them to the moon and bring them back eventually over ten years. But it's not really enough to do a Mars mission. So I think we won't really know if there's a practical Mars program until we see the next president – to see if they will build upon what this president has started or if they'll just scrap it and start all over. So I think the current president has a number of good ideas but he hasn't allocated enough resources to do the job during his turn, and he hasn't shown a good enough reason for spending that much money to the general public to justify it yet. So if we get more or less money because of that we don't know.

Well, if you were *to decide how to spend the money, what should they be spent on?*

– Well, how much money do I get to spend? You have to tell me my allowance and I'll tell you what I'm going to buy.

The same amount you have now.

– So NASA gets about fourteen billion a year... okay I don't know enough to do the whole space program. The Earth science program in the United States has kind of stagnated because they don't have a vision of where they are trying to get. It seems clear to me that they should be trying to understand the long-term climate, but it's not an easy problem to solve. So I don't know enough to focus resources in that area. I see lots of people battling and proposing mission to do this and that but I don't know enough. But I guess I would spend a lot more money on people coming up with models and theories to understand what we need to measure than spend it on space. I don't see enough effort going into finding out what needs to be measured. I think the Mars program is okay. We need to bring samples back from Mars; we need to bring samples back from Europa. That is going to take a long time. But men in space, that is a hard problem. There is a society started by a guy named Gerard O'Neal about putting colonies in space stations and that's a lot more practical than doing a colony on Mars. It's still very, very hard and it's not clear that that expense could be justified. So this is a problem I would say society has, NASA has. I think we need to have humans in space because I think humans need to look beyond Earth, but it is hard to find a goal that is achievable with a reasonable amount of resources. It's tough. So the space station was sort of an intermediate way station point. But it's not really on the way to anything. So it's a hard problem. I think we will se the Chinese going to the moon in the next decade. So that may put some pressure on NASA to do the same. But that is a short-term thing. I don't see a long term plan yet.

– I have to escort you out now. So shall we pick this up again tomorrow?

Sure, that would be great.

Figur 4.5 På besøg i JPLs museum. Til venstre: Doktor William Folkner og Bjørn. Til højre: Doktor William Folkner.

4.7 Doktor William Folkner 17/3 2005

– ...[about trajectories] There is not a unique minimum, but there are usually two minima that are similar. This is called the Hohmann ellipse [draws]. So there is the sun, here is Earth's orbit and here is Mars' orbit. So when you launch the Earth is here and Mars is there. And seven to eight months later Mars would be here and Earth would be there. So this is the Hohmann transfer ellipse, that's the ideal perfect minima and that is unique. But because the orbits aren't planar it turns out there is this type 1 where you will hit Mars here, and there is the type 2 where you hit it there, so it is the difference between seven and ten months orbit. The MER spacecraft were both sent on type 1.

So it was the same minimum?

– It was the same minimum, but one of them was launched further from the minimum. They both got there in about seven months, give or take. Sometimes it turns out because the planes aren't coplanar that it is energetically a little bit more advantageous to try not to hit Mars near here [points to drawing] but here at a later time. So the travel time will not be seven or ten months but perhaps two years and seven months or two years and ten month. Those are type 3 and type 4. There is a Japanese mission to Mars, I can't remember what it's called, but it was supposed to get to Mars in 2003, but they actually launched six years earlier. But they had a failure in their injection to orbit and so had to take another minimum approach, so it took longer to get there. Missions to Venus will often take multiple revolutions because it is energetically cheaper and the time scale is not so long – for Mars the synodic period is two years, for Venus it is less than a year – so waiting another synodic period is not that expensive. And you might gain more in terms of energy of launch vehicle than you pay for additional operations time.

It's like a cost-benefit analysis?

– Everything is a cost-benefit analysis. The whole space system is a cost-benefit analysis.

So is it always the Hohmann trajectory that is being used?

– It's always similar to that. For going to the outer planets, like Saturn, it turns out it is very expensive to go from here to there. There is a lot of energy difference because the Earth is orbiting at thirty kilometers per second - and Mars is orbiting at twenty kilometers, so it is about ten kilometers a second in difference – but Saturn is certainly going zero. So three times as much velocity, ten times as much difference in energy to go to Saturn as to Mars. And so a lot of times missions that are going to Saturn will fly from the Earth, go out a little bit and come back and fly by the Earth again, and on the second time by they will actually pick up a little angular momentum from the Earth to transfer them into a little better orbitory standing to Saturn. So most missions to Saturn or Jupiter in the last few years have taken several flybys of the inner planets before finally going out. The Ulysses mission did not. The Ulysses mission just went

on a Hohmann trajectory and got to Jupiter in two and half years. The Cassini took five or six years to go to Jupiter because it took these extra flybys to try to buy angular momentum instead of paying for it with rocket fuel.

How old is the Hohmann trajectory?

– I don't know the date or the name, you ought to go search it out, but it is not much older than Newton [laughs]. Once you know the planets go in circles around the sun it's something that is pretty obvious to think about. So it might be a hundred years after Newton, but it is old.

It hasn't changed in the space age.

– No. So planning trajectories to Mars it is basically a Hohmann ellipse plus taking into account the changes of the inclinations. So there are some specialized computer programs that know approximately what the right answer is given the change of inclination at that time. It is fairly easy to crank out all the possibilities for the next fifty years. I used to have a book with plots for all the possible launch opportunities for the next ten years. Mars orbit is pretty eccentric. Earth's orbit is pretty circular. It deviates from a pure circle by a number called the eccentricity, and Earth's is about one over sixty seven, about one percent. For Mars the eccentricity is point three. So it is pretty elliptic. So that is why it in some years is much more expensive in terms of chemical propellant to go to Mars than other years, because of the eccentricity. So it makes a big difference. If you wait the right amount of time you can do it cheaper. So 2003 for instance is a very inexpensive time to launch to Mars. 2005 is very expensive to launch to Mars.

Would you call figuring out what trajectory to use a standard exercise?

– Yes, for Mars it is a very standard exercise. For the other planets it tends to be more complicated because you'll trade flybys of other planets for angular momentum against the mission operations time. For the Earth-Sun Lagrange points, which are like there [points], you know between the Earth and the sun, that's much more difficult, because it is an unstable equilibrium. There are some semi-stable points that kind of go in orbit around that point, those are called halo-orbits and they are semi-stable. But it turns out it is impossible to predict exactly how much fuel you have to use to make correction maneuvers to stay in them. You can estimate them, but there is no way to predict perfectly. So there is a lot of mathematics going on in those types of unstable orbits. For going to Mars it always comes down to: 'Do you want to do type 1 or type 2?' So if you are doing a manned mission you would almost certainly prefer type one because it is seven months instead of ten months and so it is a lot cheaper. Anything to Mars with a man will almost always be a type one. Unless you have some huge advance in propulsion so you can just power out and not care about the energy change. I don't see that happening.

Before the space elevator?

– Well, the space elevator is for throwing mass in orbit. It'll take a huge amount of mass in orbit to really fire to Mars and get there in three months.

You told us about technology yesterday when we were at the visitors-center. Will there be new demands for technology in future missions that haven't been in these?

– There are always demands for new technology. And the main reason is that throwing mass into orbit is expensive, so basically any way we can get our job done for less mass saves us money. So there is a lot of interest in having lighter things, and one of the technologies that are interesting is solar panels. Trying to convert solar light into energy – if you can do that twice as efficiently you only need half the area, which means you only need about half as much mass. And the mass of the solar panels is not insignificant. So there is a lot of money spend on developing more efficient solar panels for space. Not just for NASA, but for commercial satellite missions. And so people have done lots of different things from different types of semiconductor junctions, from silicon oxide to gallium oxide. People have talked about: 'You take your little solar cell, but what if you put a magnifying glass in front of it?' So you concentrate the sunlight more. You get some efficiency gains that way because you have a higher density of sunlight. That is something that happens a lot. Another big area people are talking about is ion-propulsion engines. A standard rocket that we launch from the surface of the Earth burns two chemicals together and the way the rocket

goes up is it spews a gas from the chemical combustion down, and the velocity of the chemical particles times their mass tells you what the force on the space craft up is. It's very simple. If you could spew the particles out faster you wouldn't have to have so much mass. So there has been a lot of work over the last twenty-thirty years to come up with a new kind of engine, that eject mass not based on chemical burning, which limit the velocity of the ejected particles to lets say a hundred meters per second or something like that. But there is a new class of engines that use a gas like xenon gas – xenon is a nice gas to use – they take xenon atoms, ionize them and they put a grid here [points] with a high potential, high voltage of maybe five thousand volts or more. That tends to draw the xenon ions out and eject them at velocities that are ten times higher than chemical velocities. So you can get a factor of ten in the amount of propellant that you have to carry. There are a few disadvantages. One is that you need high voltage power supplies in space, which tends to be a little unreliable. But they have flown them now, so how can I say this. The huge satellite company, which is now Boeing, actually flew, I think, the first one. They use a considerable amount of fuel doing correction maneuvers to keep them in the right longitude above the Earth for various communicating purposes. So they build the xenon ion engine to those trajectory projection maneuvers. NASA launched a mission called Deep Space One back in – I can't remember '98, '99, something like that – and used a much bigger ion-engine to provide propellant for Earth orbit out to a comet. There is another mission coming up called Dawn, that is going to fly by three different asteroids and the only way that is possible is by using this ion-engine, so they don't have to carry so much propellant around all the time. The amount of propellant you use is – if you have to do three maneuvers then on the first maneuver you have to have enough propellant to move the propellant for the next two maneuvers. So if the amount of propellant for the third maneuver is equal to the mass of the space craft, the amount of propellant for that [maneuver two] is equal to twice that and the amount for that [maneuver one] is equal to four times that. So it becomes very hard to go to multiple targets. So using the ion engine with the factor ten improvement in efficiency the amount of propellant for the last maneuvers is only equal to ten percent of the mass of the space craft and this is only twenty percent [maneuver two]. So the amount of propellant that can be saved is more than a factor ten, it's a huge advance. So the Dawn mission will go to various asteroids. There is a mission that ESA launched called Smart-1, which just went into orbit around the moon. It launched into an Earth orbit and used this ion engine to slowly get to a higher and higher altitude until it got to the moon. Then they use the engine to slowly slow down until they are in orbit around the moon. It took about six months, but not very much mass. ESA is going to use the same technology on a mission to Mercury called Bepi-Columbo. That will probably launch 2008.

But this ion engine is only to be used in space?

– So the problem of the ion engine is that in accelerating its particles through this high voltage you get a big efficiency but the size of the grate you can make physically and keep it practically sized won't allow you to throw very many ions out. So you can't lift the rocket off the surface of the Earth. You can't get a 1-g force out of an ion engine yet. People would love to figure out how to do that. Once you're in orbit you don't have to do all the thrust right away, you can spread it out over time. So actually planning the orbits to Mars with the ion engine is a little bit different mathematical problem than just doing one burn and waiting. It perturbs the orbit calculation a little bit but not very much.

How much of this calculation would be done in advance and how much in flight?

– It'll all be done in advance. You do this before you propose the mission, because you need to know how much fuel is necessary so you know how big a rocket you need to have to launch it and how much money the mission costs. The cost of the launch vehicle is a significant fraction of the cost of any mission, so you work all this out in advance.

So no adjustments while it is flying?

– We always do small trims. Because the engine is thrusting and you think it is passing through the center of gravity in angle 39 degrees and it is really 39.1 degrees, so you are always doing correction maneuvers. The launch vehicle usually has an injection with an error of about one percent, so the biggest correction

maneuvers for the Mars mission is right after that. So the first correction maneuver is always to clean up this error, which is only one percent. All the other correction maneuvers are much smaller. Once you get to Mars you have to stop to go into orbit, that's a big burn, though much smaller than the launch. To go into orbit is a kilometer per second compared to seven kilometers per second for launch. So a lot of iterations go on. Trial and error. But you try to minimize it as much as you can because it costs a lot to plan all these things, and so you try to have as much possible planned before launch. So this is a big area of technology. Another big area of interest here at the lab is – not so much for Mars, but for other missions – is the mass of mirrors. For instance, the Hubble telescope has a three-meter diameter mirror. That mirror weighs 800 kilograms. That's a lot. It's almost as much as the whole MER, so people work a lot on making mirrors that are much lighter. People have built mirrors now for the next generation space telescope. It's now called the James Webb space telescope. They make it out of beryllium instead of glass, and they comb it out so it's got little ribs and stiffeners. So the weight of a two and a half meter mirror now is about eighty kilograms instead of 800 kilograms. And so for doing space astronomy that mass lightening is tremendously important. For astronomy again, the technology of cooling objects is very important, because a lot of the things we would like to see in space are visible only in the infrared. Because of the universe's expansion they are red-shifted a lot. So there are a lot of infrared things. If you just take your camera and try to take a picture of an infrared object mostly you would see the heat of the camera. Unless you cool your camera down to near absolute zero. So a lot of missions depend on near absolute zero temperatures. The technology for cooling and detectors has changed a lot in the last few years. Properties of cameras have improved tremendously over the last decade. You probably have a digital camera now, right? Well, the digital cameras made for the space telescope is not as good as your camera. And they cost fifty million dollars. So the technology in doing that has advanced a lot. Computing and processing power is also always nice to have.

But you don't have as much as is in a personal computer, do you?

– Well, the MER-rovers probably have more computing power than the original pc's from 1984. But that's a lot compared to the space shuttle that has less computing power than that because it is older, so computing power is nice to have, it allows you to be sloppier with other things, but it doesn't enable anything else fundamental. It allows you to put some of the problems on the ground in orbit. That can save you a little bit of money in operations time, by preprocessing the data you don't have to send so much data back so you can save solar power. That's a minor technology. Startrackers are important so you know the attitude of the spacecraft at all time. That has evolved slowly; it's not as critical.

It seems that a lot of the work you do has to do with improving orders of magnitude – in all aspects of the mission.

– Yes. Well, the launch cost is about the same. So many dollars per kilogram into orbit is something that has been tough to beat. So that is typically a third or a quarter of the total mission cost. They're trying to get the mass and everything else down. It is tremendously important, and that's why we would like to have a thirty-meter mirror instead of a three-meter mirror, that would give us two orders of magnitude more light-collecting capabilities and if we can make that for the same mass then we can launch with the same launch vehicle. So the cost of the mission will be the same but with a factor hundred more capabilities. So we're trying to push the capabilities a lot. It usually means pushing the mass of everything else but the launch vehicle down. We are always fighting mass here. Mass and money are closely tied.

But also in the other areas. You talked about the ion engine. That was also orders of magnitude. Also the data compression and the coding aspects they are also about improving orders of magnitude.

– Yes. And that is something where processing power is an advantage. Because if you can preprocess the data and always send the good ten percent back instead of sending it all back and editing on the ground then that is more expensive in terms of processing, but you save so much in terms of the mass of the communication system that you have to fly. You can have a much more capable

instrument for the same telemetry then. So data compression onboard and data processing onboard is very important. There was a mission I was involved with – the seismology mission on the surface of Mars, which will fly some day – the seismology data, you would like to sample the vibration of the ground at several thousand times a second. Most of the time there is nothing there, but once in a while you have an earthquake. If you send all of that data to the ground most of it is wasted. So you would like to process onboard and say: 'Whoops, there was an earthquake. I'll just send that little snip of the data.' So a lot of thought has gone into to processing the data. Images or the silence.

But the fact that the MER-rovers actually send to the Mars odyssey must have saved quite a bit?

– I don't know if they have thought about it. It's hard to do a cost-benefit trade there. There is definitely a gain there, okay. There is the cost of putting a radio on the orbiter of Mars and there is the cost of that radio sending it back to Earth. It's very expensive to get mass off of the surface of the Earth and into orbit. And it is very expensive to get mass in orbit onto the surface of Mars. So if you can push most of the telemetry systems into orbit then you save a tremendous amount of mass. The fact is that the MER-rovers have radios that can talk to Earth and another one that talk to the orbiter. So they have the mass penalty of them both. But the radio that talks to the Earth requires much more power per bit. So if they had to send all the data to Earth that they now send to the orbiter they would be able to drive much less distance because they would need energy for the radio instead of the wheels. So that has been a big advantage.

It seems that it is also two different coding schemes that are used for the transmission from the rovers to the Earth, and from the rovers to the satellite.

– I don't know about that. I know what the data rates are roughly and I know the strengths of the radio signals and how much power they need, but I don't worry about the coding. I would guess the coding is not very different. Well, maybe it is...

Matijevic said that. Transmission from the rovers to the satellite would be doing checksums or something because retransmission would be very easy.

– We're getting into radio system design here. If you have a rover and an orbiter and you want to send data back and forth then you have to decide whether you want antennas that are steerable or not. There is a trade there. It turns out if you don't want to steer the antennas because it costs money to build the steering motors then for a given amount of power you get better data throughput for a longer radio wavelength. It's a simple geometrical calculation; let's not go into that. Because of that they are transmitting from the rover to the orbiter at a radio wavelength of about 75 centimeters. So they don't have to steer them. If you had steered antennas then it's much cheaper per bit, in terms of watts, to use a shorter wavelength. The rover to the orbiter has unsteered antennas so they are using a long wavelength whereas the rover to earth requires steered antennas so they are using a three-centimeter wavelength. One of the problems with this long wavelength is that to transmit effectively you need an antenna about the size of the wavelength, so the antenna is physically bigger. The other problem is the radio signal from the orbiter to the rover. Some radio power will go directly to the antenna, but there is also the chance that it will bounce off some rock and then go into the antenna. So you get a lot more interference between the antenna and the rover itself, its camera, solar panels and rocks around it than you do if you have a shorter wavelength on a directed antenna. So because of that I would expect, I don't actually know, that the radio transmission to the orbiter and to the lander probably has very rapidly varying efficiency. You might go through a very bad minute and you might want to encode you data differently, so instead of doing just a straight (7,2)-code you might want to use a code where you encode the bits partly in this minute and partly in that minute. So if you have a noisy minute you would have a clean minute later. That would lead to a very different coding strategy. I could imagine this being important. Or you can just transmit it again the next day if you lost it. I don't really know what they have chosen to do.

We have some other question in a different area. What is the most exciting discovery you have done on Mars – the most exciting output of the mission?

– The most exciting for the MER-mission was to characterize the properties of the bedrock. I'm afraid I haven't followed that completely. There are meteorites found on Earth in Antarctica that are thought to come from Mars. The reason they are thought to come from Mars is that there are pockets of the rock that holds sealed gas inside. When those pockets are opened on Earth the molecular content of the gas doesn't look like Earths atmosphere, it looks like Mars'. But the rocks we found on Earth don't look much like the dust on Mars. So by being able to find bedrock at the second MER site and comparing it to those asteroids... that was really very exciting for me, because you want to see if these rocks you find on Earth are characteristic of Mars.

So were they from Mars? Were they able to tell?

– Well, they are pretty sure they are from Mars, but the question is: 'How characteristic are they for the planet?' 'Were they strange rocks or were they typical rocks?' So I think finding out that they are more typical is very important. I don't know if you know, but in at least one of these meteorites in Antarctica that are thought to come from Mars they see little hints that something was alive.

But that is old news isn't it?

– That is old news. But being able to track the rock back to bedrock on Mars, I think was very significant. I don't know. You have to ask a geologist.

So in your opinion that is the most exciting output. How about input. What has been the most exciting new thing in this mission to work with?

– Well, the fact that the rover could go around and find such different types of rocks and be able to characterize them was very surprising to me. It was gratifying. I think the MER-scientists, the geologists thought that that would work out. I was kind of skeptical. But the fact is it turned out to be incredibly important – far more important than other things. In fact a lot of people would probably say: 'The most important thing that has come from Mars is the hints we are finding that there was water on the surface.' Nobody really doubts that it was there. But there is no direct evidence that it was. So the MER rovers have been able to go and find rocks that were in layers. I doubted that they would ever find rocks that are layered, but in fact they did, so they were able to show that water existed on the surface for a long period of time. That direct evidence of water is definitely one of the most interesting things and it really came from the mobility of the rover. From the Viking landers, other orbiters, the previous Pathfinder rover, nothing really came out like that. This rover was able to go far enough to find distinctive enough rocks to look for this type of behavior that really give an insight into the past climate of Mars. So that was very significant – more than I ever expected.

So if you compare the MER mission and the Pathfinder mission, is the difference to be found in the rover or are there any other areas?

– The Pathfinder mission was not really meant to be a science mission. The Pathfinders mission was to prove that you can land on Mars fairly cheaply. The Viking landers that landed on Mars used a very expensive rocket landing system. The total amount of money for the Viking mission was about two billion dollars and that was in 1974. So that is about eight billion dollars today. So the whole purpose of Mars Pathfinder was to show that you could land on Mars for under a half a billion dollars using airbags to just bounce off, survive and then open up. So it was really a technology mission, there was not really a science payload. But it showed that you can land on Mars much more cheaply. The MER rovers then took advantage of that demonstration to put a more expensive science payload on a long-lived roving vehicle. So MER is the payoff of Pathfinder. You can't compare them scientifically. They weren't intended to be. One was the precursor to the other. The Viking landers were phenomenal, but at eight billion dollars you can do a pretty expensive mission. They had a lot of life finding in the mission and arms that dug through the soil thirty years ago. That was fabulous. But the MERs are better for less money and that is because of the advance of the technology – of the landing, of the communications – of everything.

Do you see any differences in the math involved in those missions or is it basically the same?

– It is very similar. In terms of design of the trajectory, doing the computations for finding out what time of day it is and where to steer the antenna, which direction to put the arm in, it's very similar. I don't think the mathematics has changed very much.

Perhaps except the fact that Pathfinder and MER are driving and the Viking landers weren't?

– I don't consider that a big mathematical difference. It is a huge technological difference, but not a big mathematical – I don't think there is a different class of mathematics invented to make it happen.

That could be control theory. But if they are not really using it...

– The Viking had powered descent. They had rockets coming down, a much more complex control than Pathfinder or MER. Much more complex.

Miguel san Martin said that the math in the future missions would be more demanding than in the MER mission.

– I don't see why. He must know something that I don't know. I don't claim to know everything. Why would that be? I mean in some sense you would like to have artificial intelligence on Mars so it can decide what is important and what to do. But is artificial intelligence mathematics?

Yeah, we think so.

– I think so too. Artificial intelligence has progressed very little, I think, in the last thirty years. I'm really disappointed of how far it has come. There has been a lot of interest here in having artificial intelligence capabilities in the spacecrafts and so far they have not been very demanding. It's hard to get artificial intelligence with good judgmental capabilities. So if that ever happened I would say: 'Yeah, that is a big breakthrough.' For the rover to decide were to go itself and to decide whether it should get up the slope or not. I mean, people here are spending weeks looking at pictures to decide. If the rover knew what to do you could get ten times as much science back in the same amount of time. It would be a tremendous breakthrough, but I can't see that happening. The next rover plan is supposed to be very big, but I don't know how much automation it really has. Miguel might know.

We have a totally different line of questions for you also. The professor we went to see at Brown University is an old professor of applied mathematics, Philip Davis. He thought that it would be interesting for us to study some human aspects of the people who are doing the science in the Mars mission. So that would be like: 'When did you get your degree? When did you come to JPL? Did you know when you came to JPL that you would probably stay until your retired?'

– A lot of people come to JPL to work for a long time. I got my Ph.D. in physics from the University of Maryland. I thought I would come here to do planetary exploration. You know, space exploration is a long-term thing. A project takes about ten years from thinking about it to getting the data back. We use personal computers on our desks a lot more than they did twenty years ago and so the type of planning we can do helps us a lot in designing more complicated missions. So computing power is a significant factor in our everyday lives and things are much quicker because of that. There was a big change – I think I talked about this the other day, when Dan Golden, the NASA administrator tried to make smaller cheaper missions more often. When I joined the lab in 1988, JPL had not launched a mission since Voyager in 1977. That is ten years with no launched missions, so it was a very quiet, sedate, calm place to work. Now it is a very stressful, busy, exciting place to work, so it has changed an awful lot. But that is not because mathematics is involved. Some people say it is because the Soviet Union doesn't exist anymore. I would say it is more anthropology or philosophy. You could say that doing smaller missions more often is a philosophy that has been applied to space exploration with very definite effects. Most of the people I know has been here a long time or tend to be here a long time. There aren't many people who come here who just want to be here a year or two and then go away again. People who want to do space work tend to want to do space work. I think this is the best place in the world to do space work. The people here are all very, very good. They are all very dedicated and they want to make these things fly. There are other space institutes around the world but I don't think any of them has the breadth of experiments going on as we do.

Is your funding larger than theirs?

– Well, Goddard Space Center has a larger funding than JPL does. Houston has a bigger, but the manned missions are bigger. Certainly sending men into space is a significant endeavor, and I'm sure they're excited about their jobs, but a lot of it is about safety, whether or not this thing will blow up, and rehearsing things. A lot of ours is planning and running numbers through trajectory calculations, so is one more exciting than the other? I don't know. But in the end we do lots of more science experiments. But they do a lot more 'humans in space' experiments. It's a matter of taste. There is no right and wrong there. We have very little to do with manned space.

When you came here seventeen years ago, did you have to do a lot of work manually, with your hands?

– No, no. Let's see the IBM pc came out in 1984 so when I came here most of the work was done with online data entry and program editing and what not. Ten years earlier when I was in college, I had a part time job working for the air force doing punch cards and looking at printout. It took a lot longer to develop software. When I was in college calculators had just come out. By the time I got here everything was computerized. The biggest advance in computers, between when I started here in '88 and now, is disc storage – just the ability to store information. It's so much easier now. We used to have ten megabytes or one-megabyte disc drives, so we always had to delete things and decide what to delete and what to keep. Now we just keep everything. That makes life much easier.

You still have to make those decisions on the rovers, though?

– That is all done pretty much the same. You still do those pretty carefully and think your way through. It's not that different from what it used to be.

We talked to Kiely and Klimesh, and they said that they had to redo some of their programs for optimization of speed for the MER mission. They did a first version but had to do another one.

– Well computing power in space is not as good as it is on the ground and, I don't know if you use any Microsoft products, but they are clearly not optimized for speed. Programmers now write programs as if computing power is infinite and they don't optimize them like we have to do for the spacecraft. Our ground-systems are not as optimized because they are so much faster than the spacecraft. This is not really a software development environment. We write a lot of software, but I wouldn't call it state of the art, right? People who do banks or keep track of where airplanes are, probably have much bigger software environments. Those are huge software jobs compared to what we do. We are doing one thing at a time, and try to do it very carefully. We have a much wider range of environments to deal with and lots of different kinds of programs. But our programs are probably much less sophisticated. I mean, we have very smart people writing programs here, but it is just not as demanding. Everything has to be done right.

For how many years do people tend to stay at JPL? They do their Ph.D. and they are done when they are how old?

– 25 or 30. There are a lot of people here who has been here longer than I have, and I have been here seventeen years. More than half, I would say, have been here longer than seventeen years. A problem in doing the smaller missions is that you can only do them with reasonably experienced people, and because they are small you don't have the budget to have an experienced person train an inexperienced person. So JPL is getting older on the average, because we don't have a big mission to afford enough people to have senior and junior people. So JPL is short of junior people. That is not a problem yet, but in five years or ten years it will be a disaster. The management here knows that and is trying to deal with it but it is not often you fix problems until they occur. They are trying to get ahead of that, but it is a difficult thing. Because we are trying to do so many cheap missions we depend on experienced people and we are not budgeting training inexperienced people. It's probably true throughout NASA and the aerospace industry.

You don't have many Ph.D. students? That would be one way to get cheaper labour?

– I don't like that ethically. I think if you have a higher educated, trained, young person you should pay them like a higher educated, trained, young person and

not give them half of what they could get at a bank. So I don't like to think of Ph.D. students as cheap labour. But there are people who don't agree with that. There are a number of post docs here. There are five thousand people here. How many post docs are there? I don't know, maybe a hundred and fifty – something like that. We don't have a lot of graduate students and part of that is because almost everything we do here has to be done on a particular time scale. Very little of our budget is spent doing research where if we get the answer next year or the year after that, it doesn't matter. Everything has to be done on a schedule. So it is not a good training environment for graduate students. Graduate students here are not used much, because the people who could supervise them are busy doing other things.

Okay, thanks. I think we're about done.

4.8 Doktor Mark W. Maimone 17/3 2005

... we were hoping you could tell us something about the steering aspects.

– So I assume you are talking to other people about the orbital dynamics and the cruise. Right, well the kind of processing we have on board is able to take pairs of images and do stereo image processing to map the pictures into distance measurements. Most of that is actually done with integer arithmetic because you are just looking at two arrays and you are sort of comparing them to find out which pieces correspond. So images all have some number of bits per pixel, usually we use eight bits per pixel, so we are looking at it with two matrices with numbers 0 through 255 and we try to find how they match up between images. So we are doing something like correlation to do that comparison. We take a window of seven pixels by seven pixels in either image and you try to find... given a window in one, find one that best matches it in the other one. So we are using correlation there, anyway that could be a continuous function but we are just doing an integer discrete version of it. Then once we've done all that work we get a matrix that has instead of brightness something we call disparity...a name for what the difference is if you were to overlay the two images. Overlay the two, find the same feature and look at the number of pixels difference between them. You'll find that nearer things have a larger disparity, they are farther apart. Farther away things have a smaller disparity. Before we launched we calibrated the cameras. We took pictures of a 3D target that we built so we knew exactly what its dimensions were. By taking the pictures we can calibrate the system to know what the lens is doing to light coming into the system. The basic idea is that if you have points that correspond you can triangulate to find the range of the target and that's how the rover understands the world around it, by triangulation. So it's all integer math to do the first part but then eventually you have to get into floating point operations because you are projecting it into a world model and for every point in the image we want to know how far away that point is. It is as though you took measuring tape to like 20,000 different points in front of the rover. When it is done, processing a pair of images is about how much data we get. So there is correlation to find matching points and there is triangulation to do the ranging to the target and what that gives you is basically a cloud of points, you have (x, y, z) locations for what the terrain around the rover looks like. Then you get into the next processing step which tries to figure out how safe it would be if it drives in a certain way. And the way we continue processing is, we have this cloud of range points – you have a lot nearby because on the image you see the ground close by, there're a lot of points nearby on the ground and fewer farther away – so we take only points that are reasonably close by, four or five meters from the rover, and we put all of those points to a plane. So you do the planar fit and that gives you an idea of what the local shape of the terrain is, you get the tilt of the plane, the parts of the tilts the rover would be at if it were to drive there, you look at the residual of the planar fit and that gives you an idea of how rough it is, whether it is flat or has a lot of rocks and then you can also take the plane equation and look for large steps, you'll be looking for tall rocks or a deep ditch or some combination and you mark those as hazards, as places you don't want to drive.

Who marks the hazards, you do or the rover does?

– The rover does. Given all the (x, y, z) points the rover looks at different patches around it. We divide the world into small cells of about 20 centimeters on a side and you look at a disc centered on each cell. For the plane you do all these sorts of assessments to see, well, how would it be if the center of the rover were at this point? It is sort of orientation independent because it is just modelling the rover as a disc. So you just look at the disc, fit the plane and you look and see where there might be step hazards or tilt hazards or if it is too rough. And the rover is doing all of this onboard, it doesn't bother sending back to Earth, it's just processing right there. Once it has done all that analysis in 3D it makes an assessment of how safe it is. So we set the bounds like what's our maximum obstacle height, how much of a tilt are we going to let it get into and given those constraints it takes the plane parameters and for every cell evaluates those different filters. So every filter gets a contribution of how rough it is, what the tilts are, how are the step hazards and it takes the minimum assessment of all of those. So you may be looking at a 30 degree slope but no rocks so the step filter would say, 'probably you can go there, it's perfectly flat', and the roughness filter would say, 'no problem, you can go there, it's perfectly flat', but the tilt filter would say, 'no it's too much of a tilt'. So given that assessment we get a 2D color or intensity varying assessment of how safe each position is. So given that assessment the rover then considers its different options; you can go forth, you can go back, you can turn in place, you can arc to the left, arc to the right, forth and back. Every path gets evaluated at every step and it is sort of integrating the weights we assigned along the way, it's basically saying, 'what if I was to go along this path, how safe would it be?' So every path gets a vote, it's assigned a vote, an unsafe path gets a low assessment and safer ones are higher and if it didn't have any data to work with we tune that but we generally give it a very low assessment because we don't want to go some place where we don't know anything about the terrain. So that all gets done for every possible path and from there... when we command the rover we never tell it to sort of drive randomly, we always ask it to go to a certain location, a certain goal, so whatever the safety assessment is it then tries to see, 'well which of those will help me to get toward the goal?' So as long as there is a safe path more or less towards the goal it's going to pick that. If there is a rock in the way and it can't go directly towards to goal, it is going to go a little bit off to the side, sort of steer around it as it goes. If it is too close to a rock or a wall or a ditch, what it will do is actually back off, back away from it... because it is considering all the paths; forward and backward. So it will pick whatever is safe. First and foremost it has to be safe, secondarily it will go toward the goal but the backing off behavior helps it getting around some small obstacles because it sees it can't go that way, it'll back off and try to get in from another angle on it.

So the procedure is that you look at the pictures that it's taking of the 360 environment...

– It's not a 360. It is generally just from one pair of cameras, so if it is from the body cameras it is about a 120 degrees field of view. Unfortunately I couldn't use the body cameras on Opportunity because the sand was too fine grain. At the levels of resolution we are able to process it just looks like a gray image. So we can't find any corresponding features because they all look the same. So I had to go to the mast cameras and try a narrower field of view, about 25 degrees.

So the people uses the pictures to figure out the goal, but does the rover figure out the path for itself?

– Well we can command it either way. What I've been describing is the system where we just tell the rover the final destination of where we want it to be and it chooses its own path. There are other ways we can use it. We can also have the rover do an analysis of the terrain but instead of allowing it to choose its own path we say, 'go this way' or 'pick this particular path if you think it is safe, if you don't think it is safe just stop and abort, don't do anymore driving'. So the phrase we have been using when it picks its own path, we call that *auto-nav*, autonomous navigation, and when humans pick the path but ask the rover to make sure it is safe we call that *guarded motion* and the other mode would be where the rovers doesn't use its cameras at all and we just tell it 'don't even look just go here, we're telling you how we want you to go' and that's either

directed driving or *manual driving* or *blind driving* because it is not taking any pictures.

But it is still just four or five meters?

– No Opportunity has gone over a hundred meters in a series of blind segments. The justification is: We have the body cameras of a 120 degrees, mast cameras with 45 degrees of view and mast cameras with 18 degrees field of view. Using those cameras it is possible to get an idea of the shape of the terrain out to about 50 or 60 meters reliably and still even as far as a hundred meters, less reliably. But generally based on the terrain humans will look at the pictures and say: 'Well, the terrain as far as we can measure is perfectly safe and from what we can tell even beyond where measurement is possible it is very smooth and there are no rocks, so we're just going to command it to go a little further.' Because when you drive without looking it is much faster. It has a 20 MHz computer onboard which takes a while to do this image processing. So basically if we need to cover a long distance then we'll start with blind drive and follow it at whatever distance we are comfortable with. We have gone as far as 100 meters and as short as 5 meters if all we have are the body mapping cameras. So whatever distance we are comfortable with we'll start with that and then turn on the other system. It is kind of strange but when we originally developed that system, I thought that it would be used most of the time as just a backup safety system. But because it takes so long to do the processing instead of using it as a backup they are using it basically when they don't know what the terrain is going to be like, when they have insufficient details to understand the terrain, then they turn that on and the rover drives itself.

Once you pick out a goal how does the rover know where it is?

– The rover has several instruments to maintain its position estimates. We have what is called an inertial measurement unit (IMU) which is basically driver scopes and accelerometers. So the accelerometers tell you where gravity is and the drivers will tell you where your change in attitude is. So whenever it is moving we have that unit turned on and plotting data about what its current attitude is. So we actually know our latitude very precisely, the maximum drift according to the specifications is about three degrees per hour, and that is just while it is actually moving, so that is how we maintain the latitude. And to measure position the nominal system is simply counting wheel rotations. So we'll send a command that says 'drive for two meters' and the software then convert that command into a number of radians to turn each wheel. And we have encoders that measure what it actually does, if the wheel stalls or something it would know that it is bad. And that's how we updated the vision.

But what if the wheels are just digging sand?

– All right. Given that system the rover would not know if it was stuck in one place just turning up sand. So we have another capability onboard which we use sometime but not always because again it takes a while to do processing. Again we are looking at stereo pictures or pairs of pictures and the system is called *visual odometry*, it [the system] is trying to describe that instead of using wheel rotations to get odometry you are using pictures to measure odometry. Well it's all there onboard the rover, it's all done autonomously. The rover picks out an interesting feature by itself and in between some motion command, half a meter whatever or turning 10 degrees, it will take before and after pictures. And if that motor is turned on it will be able to compare the pictures, find the features in common between the two sets of images and measure the difference. So once you know the difference in the image then the cameras are calibrated, we know in 3D where everything is so we can then use that to back out what happened to the whole system. So when we drive at that mode we usually plan shot segments and points cameras in some directions where there will be features. This is a problem for Opportunity because a lot of the terrain is very loose sand, no interesting features. But generally the times we are using it is when we are on slopes and you can always see something interesting there. So cameras are pointed at interesting features, it moves in small steps and updates its mileage along the way.

If you imagine the rover being in a totally anonymous landscape and all the wheels just digging sand all the time, would you be able to tell?

– No, no. Well maybe possibly by looking at the rovers tracks you could see something. But in that case if we really cared about that, if we really felt that we would be slipping too much we probably do something like command a course that had enough of a deviation so you would be able to see it. So there are always options. Then you say 'well what if the ground is not sand, then you can't see the tracks'. If you can't see the tracks probably it is hard enough that it will go. Put us on a sheet of ice... well [laughs]. No, we were landing on the equator and we knew that there weren't sheets of ice there so the system is designed for that environment.

How much artificial intelligence is onboard the rovers?

– It depends where you want to draw your line. We don't have a planner in the usual AI sense because a planner is something that generally considers many possible behavior paths and minimizes some functions to achieve some goal and we are not really doing that because people are always choosing the goals. But on the other hand you could say that in every step it is being intelligent about building up its own model of the world, it's looking at a new image of the world and has no assumptions about what was there in the first place, it's understanding the terrain... so it is being artificially intelligent about seeing what the terrain is. And it is also being somewhat intelligent about picking its own path... it's actually being kind of stupid in that regard because it is always just trying to go towards the goal, it's not smart enough... If we were to get into a blind ally, you know where you go in and have to turn around and go a long way back, we would be stuck. I had proposed doing something like that before but that didn't get incorporated so we don't have that kind of capability. To me the interesting artificial intelligence onboard is the terrain analysis and assessment because we are not telling it anything about the terrain. It is figuring that out on its own and staying safe. But it has many other autonomous features that you wouldn't think of as general AI capability but the whole system, the fact that it goes to sleep and wakes up every day, can deal with the fact that there are communication passes that come at unscheduled times or could interrupt whatever it is doing... If it is in the middle of a science measurement or something but it notices a communication pass, it's smart enough to stop that process and back off and welcome the communication passing. So the system is architectured in a way to keep it in touch with Earth and when it is not in touch with Earth to do intelligent things.

How much control theory is used?

– Well we have to drive the wheels so there is software that controls the motors there and all the instruments have motors and have to be controlled so there is some amount of control theory being applied there. I don't know what has been published about the details of those algorithms so I can't tell you anything about that. But there are a lot of motors, there are motors that drive the wheels, that spin the cameras, that control the arm and when we landed we had a lot of motors that were used simply to get up off the lander; stand up, spread out the wheels, spread out the legs, open up the solar panels, pull up the mast and do all those things.

Just for plain interest, are the rovers capable of doing those in reverse?

– No we specifically designed it not to be able to do that because... for example the solar panels had motors that open up and when they open up there is a latch there that locks it in place because we were afraid if there were very high wind we didn't want them to fall over again.

What is new on the rovers compared to the vehicle that was on the Pathfinder mission?

– Well Pathfinder had the smaller rover Sojourner and that actually was pretty smart for what it was. Sojourner had an even slower computer; it was a 0.1 Mhz, a 100 khz processor, so it couldn't do very much. But Sojourner had the ability to do the same kind of obstacle detection on a grosser closer scale. They projected laser stripes out and did the same thing to find matching points in images but they only checked 20 points – on MER we check 20,000 points – so it was able to autonomously see what the terrain looked like on a gross close level. MER drives a little more quickly than the Sojourner. The Sojourner mission lasted for three months and in that time it drove over a hundred meters. MER rovers have driven more than that in one day now, so being bigger has its advantages and being able to go a little faster.

Is that due to the engine or...?

– It's just larger, the wheels are larger and we are able to command it better. With more solar cells you can get more power. So the Sojourner actually had more autonomous capabilities than was actually used. For instance Sojourner could have done its own driving for long distances, but it was never given the chance because by the time they were willing to test that technology the mission was over. Even on MER it was a slow progression, we started making very slow measure drives and we hadn't worked our way up to a hundred metes drive until – when was that – I guess about a hundred sols later. So after the main mission was over, we started trying to cover a lot of ground and learn how to increase our drive distance.

How much RAM does the computers onboard the rovers have?

– I can't keep it straight. It's like a 128 to 256 Mb.

And how much physical memory for storage?

– I don't even remember. It's split up, we have EE-prom and we have Flash. [...]

What will be new in steering in future missions?

– Well I don't know yet. They are still designing. A lot of times you want to apply something that you understand from before. So for instance the suspension system on Sojourner and MER are very similar. It's called the *rocker-bogie suspension system*. There is one axel that has a differential inside, so when one side goes up the other goes down – so they call that the rocker. On the end of one of those arms there is something that is just free, it is not connected to anything – they call that the bogie – that's just loose. The system tends to average out any climbing behavior, so if one wheel is climbing a rock the body tends to shift only by a quarter the size of that rock. It's not like a car that would have to climb and get rigidly over that. So those things are all the same. The new MSL [Mars Science Laboratory] rover should be larger, but the pictures I've seen all show a very similar kind of suspension system. It's possible that we'll be able to steer the wheels more but maybe, maybe not. It is still early in the design. It is actually sort of amusing that if you track the history of what is promised in the next mission they had to keep increasing their goals because we keep exceeding our goals.

It seems that the rovers are just growing in size. Can you keep doing that?

– Well you can, it depends how much you are willing to launch. That is the trend; to make it bigger so it can cover longer distances.

But you could also give it a more powerful engine or...

– Well yeah, that depends on your power source. We can't get a gas engine up there because we can't refuel. They have actually announced that they are considering using a nuclear power source on a rover. We are not talking a huge nuclear reactor we are just talking a small one. Of course it can give you more electrical power which gives you flexibility because you are not constrained to be powered by the sun. But it doesn't really give you that much more power, I don't even know the power numbers but it is more or less the same. The rovers generally move very slowly because we want to be able to overcome any obstacles, so it is generally in a high gear ratio so you can make sure that the climbing works. That means that speed is very slow.

Would you call designing the steering a standard exercise?

– Well space is never standard, there is always something confounding it. A lot of the research-rovers that you'll read about have steering motors that are above the wheels which means that the wheels can steer any amount in any direction. But that wasn't possible on MER because we had to make everything fit inside the landing shell. It had the motors on top at one point but when they tried to squeeze everything into the shell they found out 'well these motors will be sticking out the shell, so we have to do something else'. So things move around and you have to deal with all these constraints. When designing a space system you can't optimize for one environment; it has to survive launch, cruise, entry into the atmosphere, landing and then you get to drive on the surface. So anything you want to pull off of the surface has to survive all those other phases too.

How much of the technology that is involved with the rovers would you say is 'state of the art' and how much are things that are being done in everyday technology?

– Any space mission generally uses old technology...

... for reliability and so.

– Right, right. So that's why the computer is so slow. I mean by today's standard you can get a hundred times faster computer for your desktop for not much money. And in terms of ability we have to real the fact that we are sending things to extremely cold environments with temperature shifts of a hundred degrees every day, so you need something... to survive this environment. It depends on your definition of 'state of the art', if you are talking state of the art for Earth these things are slow and stupid, but if you are talking state of the art for surviving in space they are setting and breaking new records every week.

We were told at some point of time that the software for controlling the rovers weren't actually finished at the time of launch...

– That's true.

So they was transmitted while it was in cruise?

– Well, it's a little misleading. We launched with software that could have run the whole mission. But we were able to send new software uploads to the rovers and we have done it several times. So we launched with one version, we send a new version before we landed, we ran for a few months with that version, and in April both rovers got an upload of new software and went with that for the last eight or nine months and just this month we sent up yet another version of new software. The thing is that for NASA this is a very compressed design, this mission was only approved in August 2000 and both rovers were launched less than three years later in summer 2003. So on the time it was approved that's when we started designing the thing, all of the mechanically designs, the science instruments. There are some leftover designs that you try to reuse but it is always a question of integrating everything and meeting the new constraints so you know what the system is. So we didn't have complete test vehicles to work with until about seven or eight months before the launch. Certainly all of the hardware was done, but most of the software that was concerned with launch and cruise and even the entry through the atmosphere that all absolutely had to be in place before we could launch. But the surface part of the system, because there was so little time to do any testing... everything else had to work in order for the surface mission to happen; it had to launch, it had to survive seven months in space, it had to go in through the atmosphere, it had to survive landing, it had to open up on the ground and drive off of the lander. All that had to work before anything else could happen. So the emphasis during development was on all those other subsystems. The surface mission part... of course it was a goal to do that, but it had to come second to all those other things, so we didn't actually get to do all of our field testing until after we had launched. We could have driven the mission with the software we launched with but as a result of the field testing we found some little things that would work better if we fixed it and change this and that. So we were able to do these improvements after we got on the way. So all of the changes that were made were related to mobility. Since we did some other testing after launch, it made sense that we would have to make those changes later.

Did you have to retransmit the whole software package or just parts of it?

– We have done both. We have sent the whole package and then we have also just recently sent the changes that we had made.

How does that work; do you have to reboot the whole system?

– We reboot every day, often several times. When you send a new version of course the rover first makes sure it is the right thing. You check it to see if all the files are working probably. Humans on earth also review all the data before we ever reboot, make sure that everything was sent correctly and then once on the set we have the ability to reboot into a new version and replace the old version with a new one... but we always have backups, we can always revert to the previous version.

Do you transmit to the satellite and from there to the rovers?

– Again we can do either, but actually you have to talk to someone else to get the details on that.

How many people are involved with the steering?

– Well that depends how you count. On the software team it is not only the software developers it is also the testers and the people reviewing the code that's written, to make sure it is correct, it doesn't have problems in it and there are system engineers who write the requirements of what the machine has to do. There are dozens of people involved with that.

But not hundreds?

– The whole mission at its peak had about a thousand people working on it. There were often hundreds of people on the whole mission but again not specifically on the steering because a lot of the focus was on these other aspects of the mission.

And the work is mainly software development?

– Well, I mean, of course the mechanical team has to design a system that can accomplish something useful. A lot of the mechanically designers did the design and the testing. Then once that system was build, there was software being developed to support it. It is not a one person effort.

When you design the system are there any standards you have to follow?

– Yeah, the whole mission had coding standards that we had to meet, standards in every step, extra testing, quality issues and design reviews... all these things. Part of the proposed mission was to have a certain level of software standard so that is how we implemented it.

All the software is done here at JPL?

– Yeah, yeah, the software team was based at JPL... yeah.

What computer language did you program it in?

– The computer system is related to Power PC. A little bit of Assembly language but mostly C. Part of the mobility system was C++.

There's a lot of programming for code efficiency and minimal use of storage space and stuff like that?

– Yeah, that's part of the software requirements, it has to work with the resources it's given.

You said earlier that you did test drives. How much of the testing of the rovers is done in computer simulations and how much is done physically?

– As much as possible in both. Before we had the vehicles, I had to work in some sort of simulation environment. We actually had the software working on an earlier prototype. Back in 1999 the plan for the 2003 Mars rover mission was that it was going to be a send and return mission. That got cancelled. But before it got cancelled we actually built a prototype rover with an earlier design and we used that to show that the autonomous driving capability would work driving around obstacles. So I had that real environment – you can call it a simulation because it wasn't MER software going on. Then as we were developing MER software we did have ways of simulating aspects of the rover but never... you don't have a complete simulation system for all the terrain and all the possibilities and all the imaging, that's very difficult. So we did as much as we could in simulation but it can never substitute for real field testing. And it was having a real rover built, the final vehicle built, and running it outdoors that really helped us make it robust and make it able to function on Mars where we didn't know what we were going to see.

The work that you do with the steering, do you think any of that is going to lead to spin-offs or is the work involved with steering mostly spin-ons?

– Some of the software that we develop are able to be licensed and some people do license it. But as you asked before: 'Is this new stuff?' The new stuff about it is that it is working on a vehicle on Mars, but because the vehicle is so limited; it moves so slowly, I don't know if it will have any scalps here on Earth. One of the things that I am proud of is that the system, the driving system using the cameras, is being used in places that people haven't seen. The rover has detected obstacles on its own that people haven't been able to see and driven around them, and at least in the context of being on Mars it is a pretty robust system. Have you seen the DARPA Grand Challenge? In the US they had a contest to build robots that could drive from Los Angeles to Las Vegas in ten hours. You have to drive about 30 kilometers an hour or something like that to be able to cover the distance. They had a few dozens entrants and nobody

made it. One went about one or two kilometers. That's not to say that it wasn't good systems but they were done very quickly and they were able to make some shortcuts and that's what came back to stop the progress, that they had to make some shortcuts. For Mars I wasn't allowed to make any shortcuts. I had to try and keep it as robust as possible even if that meant going more slowly. Speed is not the issue. Of course you want it to go quickly but it is much better to be safe and not move than move quickly and risk losing the mission. So if there are spin-offs it would be either on future missions, or possibly the fact that the software architecture works well. That could influence future missions or future Earth based systems, but we'll see how that goes. They are just now starting to publish papers on the software architecture, and it will be up to other people to evaluate whether it is useful. The software used on the steering, the low level software, will probably be used on other missions, at least as a demonstration of what is possible.

Do you think the overall design of the vehicles – a small car with wheels – will change in any way?

– There are all kinds of research projects looking at different vehicle configurations. We have walking robots, we have inflatable robots, we have flying robots, we have robots that can walk and drive depending on the configuration. So it all comes down to what is most appropriate for the mission. The people designing the mission are given requirements: 'We have to be able to go 20 kilometers in a year, so how do we want to do that?' Make a proposal and it will be evaluated... Often in any space mission, you fly what you know. The safe bet would be that we will see very similar kind of systems. So far we have had 6 wheels drive and 4 wheel steering that's likely to continue even though you could get better performance if you could walk in some cases or if you could get 6 wheel steering. With anything in space mass is what kills you, the more mass you can save the better off you are.

On some animation on NASA's homepage we saw vehicles that were modelled over scorpions. Is that something being considered?

– There are projects here that are building different kinds of designs like that. There are spider robots and one that is climbing cliffs... Yeah that is all considered but it depends on what the goal is. For the next mission they haven't really announced exactly where it is going to be yet. If the mission were to go to the edge of a canyon and walk down you would see a very different design than one that is supposed to drive over long distances. No one answered yet. We still build systems for a specific kind of terrain.

There is another aspect of our thesis that has to do with the human side of the Mars mission; the people involved in it, where they come from and so. For how long have you been with the JPL?

– Seven years.

Did you come directly after doing you Ph.D.?

– I did a post doc in space robotics. I did that for a year and a half before I came here.

Where?

– In Pittsburgh, Pennsylvania, US.

Was that the same place you took your Ph.D.?

– Yes.

Do you think you will be with JPL for a long time?

– Yeah, I mean, it is a nice place to be. I don't get to drive robots on Mars any other places.

When you came here did you plan to stay your whole career?

– That's a tough question these days, it's very hard to tell how things are going to go. When I was a kid I saw the Viking missions on Mars and I thought that would be pretty neat, hunched over the terminal looking at these pictures that nobody else would see for a year before it got published. But of course now every picture we take gets published on the Internet the next day so it's not much of an advantage but it's still nice.

We talked to William Folkner and it seems people that join JPL are more or less here for life.

– There are people who have worked here a long time. It is actually great to meet people who worked on the missions that inspired me as a kid. So yes a lot of people do stay for a while, but it is also tough these days because you can't always do that, you don't always have the option of staying. If there aren't the missions to support your area of expertise you have to do something to stay funded. If there aren't the missions and you don't have the research funding, you can't stay here. So you always have to work for what you are doing, and as long as the activity is going on, if it is still interesting you stay with it. You know, people on the Pathfinder mission... a lot of people stayed to do the Mars rover mission but not everyone, a lot of people left to join companies, to become professors and to do other things. They've done it once and they are happy and they want to do something else. I'm still hit by the space bug, I like to do something space related. Who knows what will happen next. It would be great if commercial opportunities in space got more interesting. Right now communication satellites are the only aspect of space that is supporting itself economically, people pay for long distance phone calls, they pay a lot of money. But so far it is not possible to do anything else in space that is self sustaining. But that is changing, have you heard of the X-price? Back at the turn of the 20th century in order to encourage aviation, airplanes, they established a series of prizes; 10,000 dollars for the first to cross the English Channel, 20,000 dollars to cross the Atlantic, a 100,000 to go around the world. Whatever, I don't know the numbers. So someone established a 10 million dollars prize for the first commercial ship to launch into space with three people. Launch, come back and launch again within two weeks. And somebody won it last year. A private company was formed to go ahead and launch this thing and it turns out that for about 20 million US dollars they designed, tested, build, launched and operated this vehicle. It wasn't really three people but it was the mass of three people. So it's the first really reusable spacecraft, the shuttle is reusable but only after months of refurbishing, rebuilding and you know. It coast half a billion dollars for each shuttle launch and here for 20 million they designed and operated a whole new system. It can't do nearly as much as the shuttle, it only gets you into a suborbital trajectory so you get a few minutes of free fall and that's about it. But it is still a great step forward because anything we can do to lower launch coasts will help make space more accessible for all sorts of purposes for research, for commercial use, whatever. Richard Branson is the billionaire who owns Virgin, Atlantic, he commissioned that company to build him, I forget, three or five spaceships so a few years from now you'll be able to buy a ticket to go into space for like a hundred thousand dollars where as the last space tourists they had to spent 20 million for one ticket. So for the prize of one ticket to the space station they build a whole new concept, a whole new ship that can go over and over and over again. That kind of thing is great and if we get more of that then it will be even easier to send Mars rovers or anything else because you get the launch costs down and make all sorts of things more possible.

Doing the space missions takes a lot of people who are experienced in the field doesn't it?

– Usually yeah. At NASA you have all the people who have all the experience, so you want to draw on that to prepare for the next one.

So there is a lot of senior-junior stuff going on?

– Yes.

William Folkner said that JPL aren't hiring too many new people so there is sort of an overload of older scientists.

– I don't know the current statistics but it certainly was true when I came here that NASA as a whole was not hiring very many people over the '70s or the '80s... once the Moon race was over they sort of parked it down. Well there is always some but it wasn't a large amount. So it is true that until the '90s there was a big gap and now a lot of the engineers who were here in the '60s building the Apollo and everything else are retiring or are retired. So yes we are backfilling but the filling in has been going on for at least a decade, so I wasn't aware of a new trend of not hiring, I know people who have been hired within

the last couple of years. I don't know the agency trends because I've been sort of buried in this one project.

All right. Thank you very much.