Copyright and Citation Considerations for This Thesis/Dissertation

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

- NonCommercial — You may not use the material for commercial purposes.

- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

How to cite this thesis

CONSTRAINED SEQUENCES AND CODES FOR BINARY ASYMMETRICAL OPTICAL CHANNELS

VOLUME 1

C. Menyennett
CONSTRAINED SEQUENCES AND CODES FOR BINARY ASYMMETRICAL OPTICAL CHANNELS

by

Calvin Menyennett

THESIS

submitted as partial fulfilment for the requirements for the degree

MASTER OF ENGINEERING in

ELECTRICAL AND ELECTRONIC ENGINEERING in the

FACULTY OF ENGINEERING at the

RAND AFRIKAANS UNIVERSITY, Johannesburg, South Africa.

SUPERVISOR : PROF. H.C. FERREIRA

DECEMBER 1991
ABSTRACT

During the past decade the optical disc has become increasingly popular. Write-once optical recording systems will mainly be used in data storage systems in which archival aspects or mass storage requirements prevail. In write-once optical data storage one is faced with an asymmetry between marks and non-marks due to a practical lower limit of the mark size. In some optical fibre communications there is also an asymmetry present in injection lasers and it may be feasible to use asymmetrical codes.

In this study information theoretical methods are used to find values of channel capacity for sequences complying with binary asymmetrical runlength constraints. Different coding methods are used to construct encoders and decoders for generating and decoding these sequences with high values of efficiency. The power spectra of maxentropic binary asymmetrical runlength limited sequences complying with different runlength constraints are also investigated.
OPSOMMING

In die afgelope dekade het die gebruik van die optiese skyf geweldig toegeneem. Lees-alleen optiese skywe word hoofsaaklik gebruik in gevalle waar groot hoeveelhede inligting vir lang tydperke gestoor moet word. In hierdie stelsels bestaan asimmetriese insetbeperkings wat betref die minimum lopielengtes van die binêre simbole as gevolg van die minimum putgrootte wat bereik kan word met die huidige lasertegnologie. In sommige optiese vesel kommunikasie bestaan daar 'n asimmetrie in die injeksie-laser en dit mag voordelig wees om hier ook asimmetriese kodes te gebruik.

In hierdie verhandeling word informasieteoretiese metodes gebruik om waardes vir die kanaalkapasiteit van hierdie stelsels te vind. Verskillende koderingsmetodes word gebruik om koderingskemas te vind wat hierdie asimmetriese sekwensies opwek met hoë waardes van effektiwiteit. Die drywingspektra van maksentropiese binêre asimmetriese lopielengtebeperkte sekwensies word ook ondersoek.
Remember that the greatest things have not yet been done, the greatest discoveries in science have not yet been made, the greatest advances in social life have not yet been achieved, the greatest triumphs of the spirit have not been won. They wait the coming of the right men and women.
DEDICATION

This thesis is dedicated to my Lord and Saviour, Jesus Christ, who has bestowed upon me the talents and ability that have made this thesis possible.
ACKNOWLEDGEMENT

I would like to express my sincere thanks to the following persons and institutions that contributed a great deal to the completion of this project:

Prof. H C Ferreira, my supervisor, for his interest, concern, encouragement, support and contributions during the course of this project.

Prof. I Broere, of the Mathematics Department, RAU, for his ideas and contributions to this project.

My mother and sister for their support during this study.

Mr. L Botha and other fellow students in the Telecommunications Laboratory, RAU, for their ideas and contributions to this project.

My family and friends for their moral support during my study.

The Foundation for Research Development for their financial support.
PRESENTATION

A considerable amount of numerical results have been calculated during the course of this project. The normal procedure is to include all these results in the thesis. The quantity of numerical results are so vast that it was decided to include the numerical results in Appendices in a separate second volume entitled:

CONSTRAINED SEQUENCES AND CODES FOR
BINARY ASYMMETRICAL OPTICAL CHANNELS
VOLUME 2 - NUMERICAL RESULTS

The reader is urged to consult this accompanying volume whenever it is felt necessary.
## CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>i</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIST OF SYMBOLS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>viii</td>
</tr>
</tbody>
</table>

1. INTRODUCTION

2. OPTICAL CHANNELS AND CHANNEL CODING
   2.1 Optical channels
   2.2 Model of a coding system

3. GENERATING FUNCTION, MARKOV MODELS AND CHANNEL CAPACITY
   3.1 Input restrictions
   3.2 Information and channel capacity
   3.3 Binary asymmetrical runlength limited sequences
      3.3.1 Generating function for \((a, \beta, \gamma, \delta)\) sequences
      3.3.2 Generating function - relationship to channel capacity
      3.3.3 Channel capacity of \((a, \beta, \gamma, \delta)\) sequences
      3.3.4 Channel capacity of \((d, k)\) sequences
      3.3.5 Channel capacity of \((a, \beta, \infty, \infty)\) sequences
      3.3.6 Channel capacity of \((d)\) sequences
      3.3.7 Relations between channel capacity for sequences with different values of parameters \((a, \beta, \gamma, \delta)\)
   3.4 Channel capacity of charge-constrained RLL sequences
4. SYNTHESIS OF THE NEW ASYMMETRICAL BINARY MODULATION CODES

4.1 Sequence-state coding methods
   4.1.1 The (1, 2, 3, 3) modulation code

4.2 Sliding block code algorithm
   4.2.1 The (1, 2, 7, 7) modulation code
   4.2.2 The (2, 3, 6, 6) modulation code
   4.2.3 The (2, 4, 4, 4) modulation code
   4.2.4 The (1, 2, 5, 5, 5) modulation code
   4.2.5 Other dc-free modulation codes

5. ERROR BEHAVIOUR OF THE NEW CODES

5.1 Channel model
5.2 Results on error propagation properties

6. POWER SPECTRA OF MAXENTROPIC RLL BINARY ASYMMETRICAL SEQUENCES

6.1 Power spectrum of maxentropic \((a, \beta, \gamma, \delta)\) sequences
6.2 Power spectrum of maxentropic \((d, k)\) sequences
6.3 Spectral nulls at frequencies other than zero

7. POWER SPECTRA OF THE NEW CODES

7.1 Power spectrum of the \((1, 2, 3, 3)\) code
7.2 Power spectra for the new codes

8. DISCUSSION AND OVERVIEW

REFERENCES
APPENDICES (VOLUME 2)

A. TABLES OF CHANNEL CAPACITIES A.1

B. SPECTRA OF MAXENTROPIC \((a, \beta, \gamma, \delta)\) SEQUENCES B.2

C. SPECTRA OF THE NEW CODES C.2

D. PROGRAM TO DETERMINE THE CHANNEL CAPACITY OF RLL \((a, \beta, \gamma, \delta)\) SEQUENCES D.1

E. PROGRAM TO DETERMINE THE CHANNEL CAPACITY OF \((a, \beta, \gamma, \delta, C)\) SEQUENCES E.1

F. PROGRAM TO DETERMINE THE TRANSITION MATRIX F.1

G. PROGRAMS TO SEARCH FOR THE APPROXIMATE EIGENVECTOR G.1

H. PROGRAMS TO DETERMINE ERROR BEHAVIOUR H.1

I. PROGRAM TO DETERMINE THE SPECTRA OF MAXENTROPIC \((a, \beta, \gamma, \delta)\) SEQUENCES I.1

J. PROGRAM TO DETERMINE THE SPECTRA OF THE MODULATION CODES J.1

K. PROGRAM TO GENERATE THE \((1, 2, 7, 7)\) CODE K.1

L. LIST OF PROGRAMS ON THE FLOPPY DISK L.1
LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIG. No.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Model of a coding system</td>
<td>2.4</td>
</tr>
<tr>
<td>3.1</td>
<td>Markov model for ((a, \beta, \gamma, \delta)) sequences</td>
<td>3.5</td>
</tr>
<tr>
<td>3.2</td>
<td>General Markov model for ((a, \beta, \gamma, \delta, C)) sequences</td>
<td>3.28</td>
</tr>
<tr>
<td>3.3</td>
<td>Markov model for a ((1, 1, 1, 2, 6)) sequence</td>
<td>3.30</td>
</tr>
<tr>
<td>3.4</td>
<td>Channel capacity for ((1, \beta, \gamma, \delta, 3)) sequences</td>
<td>3.31</td>
</tr>
<tr>
<td>3.5</td>
<td>Channel capacity for ((1, \beta, \gamma, \delta, \infty)) sequences</td>
<td>3.31</td>
</tr>
<tr>
<td>3.6</td>
<td>Channel capacity for ((2, \beta, \gamma, \delta, 3)) sequences</td>
<td>3.32</td>
</tr>
<tr>
<td>3.7</td>
<td>Channel capacity for ((2, \beta, \gamma, \delta, \infty)) sequences</td>
<td>3.32</td>
</tr>
<tr>
<td>3.8</td>
<td>Channel capacity for ((3, \beta, \gamma, \delta, 3)) sequences</td>
<td>3.33</td>
</tr>
<tr>
<td>3.9</td>
<td>Channel capacity for ((3, \beta, \gamma, \delta, \infty)) sequences</td>
<td>3.33</td>
</tr>
<tr>
<td>4.1</td>
<td>Markov model for the ((1, 2, 3, 3)) sequence</td>
<td>4.3</td>
</tr>
<tr>
<td>4.2</td>
<td>The graph (G^2)</td>
<td>4.6</td>
</tr>
<tr>
<td>4.3</td>
<td>Encoder for the ((1, 2, 3, 3)) code</td>
<td>4.7</td>
</tr>
<tr>
<td>4.4</td>
<td>Markov model for the ((1, 2, 7, 7)) sequence</td>
<td>4.9</td>
</tr>
<tr>
<td>4.5</td>
<td>Example of state 1, 8 and 9 before and after splitting</td>
<td>4.13</td>
</tr>
<tr>
<td>4.6</td>
<td>Encoder for the ((1, 2, 7, 7)) code</td>
<td>4.15</td>
</tr>
<tr>
<td>4.7</td>
<td>Markov model for the ((2, 3, 6, 6)) sequence</td>
<td>4.18</td>
</tr>
<tr>
<td>4.8</td>
<td>Encoder for the ((2, 3, 6, 6)) code</td>
<td>4.22</td>
</tr>
<tr>
<td>4.9</td>
<td>Markov model for the ((2, 4, 4, 4)) sequence</td>
<td>4.25</td>
</tr>
<tr>
<td>4.10</td>
<td>Encoder for the ((2, 4, 4, 4)) code</td>
<td>4.29</td>
</tr>
<tr>
<td>4.11</td>
<td>Markov model for the ((1, 2, 5, 5, 5)) sequence</td>
<td>4.33</td>
</tr>
<tr>
<td>4.12</td>
<td>Reduced Markov model for the ((1, 2, 5, 5, 5)) sequence</td>
<td>4.34</td>
</tr>
</tbody>
</table>
List of figures

4.13 Encoder for the (1, 2, 5, 5, 5) code

5.1 Binary symmetrical channel
5.2 Error propagation properties of the (1, 2, 3, 3) code
5.3 Error propagation properties of the (1, 2, 7, 7) code
5.4 Error propagation properties of the (2, 3, 6, 6) code
5.5 Error propagation properties of the (2, 4, 4, 4) code
5.6 Error propagation properties of the (1, 2, 5, 5, 5) code
5.7 BER versus S/N for the (1, 2, 3, 3) code
5.8 BER versus S/N for the (1, 2, 7, 7) code
5.9 BER versus S/N for the (2, 3, 6, 6) code
5.10 BER versus S/N for the (2, 4, 4, 4) code
5.11 BER versus S/N for the (1, 2, 5, 5, 5) code

6.1 Logarithmic power spectrum of maxentropic (1, β, 7, 7)
sequences, for β = 1, 2, 3, 4.
6.2 Linear power spectrum of maxentropic (1, β, 7, 7)
sequences, for β = 1, 2, 3, 4.
6.3 Logarithmic power spectrum of maxentropic (2, β, 4, 4)
sequences, for β = 1, 2, 3, 4.
6.4 Linear power spectrum of maxentropic (2, β, 4, 4)
sequences, for β = 1, 2, 3, 4.
6.5 Logarithmic power spectrum of maxentropic (2, β, 8, 8)
sequences, for β = 2, 3, 4, 5.
6.6 Linear power spectrum of maxentropic (2, β, 8, 8)
sequences, for β = 2, 3, 4, 5.
6.7 Logarithmic power spectrum of maxentropic (3, β, 8, 8)
sequences, for β = 3, 4, 5, 6.
6.8 Linear power spectrum of maxentropic (3, β, 8, 8)
sequences, for β = 3, 4, 5, 6.
6.9 Logarithmic power spectrum of maxentropic (2, 2, 4, δ)
sequences, for δ = 4, 6, 8, 10.
List of figures

6.10 Linear power spectrum of maxentropic (2, 2, 4, $\delta$) sequences, for $\delta = 4, 6, 8, 10$. 6.20
6.11 Logarithmic power spectrum of maxentropic (2, 4, 4, $\delta$) sequences, for $\delta = 4, 6, 8, 10$. 6.21
6.12 Linear power spectrum of maxentropic (2, 4, 4, $\delta$) sequences, for $\delta = 4, 6, 8, 10$. 6.21
6.13 Logarithmic power spectrum of maxentropic (3, 3, 8, $\delta$) sequences, for $\delta = 8, 10, 12, 14$. 6.22
6.14 Linear power spectrum of maxentropic (3, 3, 8, $\delta$) sequences, for $\delta = 8, 10, 12, 14$. 6.22
6.15 Logarithmic power spectrum of maxentropic (3, 5, 8, $\delta$) sequences, for $\delta = 8, 10, 12, 14$. 6.23
6.16 Linear power spectrum of maxentropic (3, 5, 8, $\delta$) sequences, for $\delta = 8, 10, 12, 14$. 6.23
6.17 Logarithmic power spectrum of maxentropic (2, 8, 8, $\delta$) sequences, for $\delta = 8, 10, 12, 14$. 6.24
6.18 Linear power spectrum of maxentropic (2, 8, 8, $\delta$) sequences, for $\delta = 8, 10, 12, 14$. 6.24
6.19 Logarithmic write spectrum of maxentropic (1, 3, 2, 3), (1, 3, 5, 3) and (2, 3, 4, 3) sequences. 6.34
6.20 Linear write spectrum of maxentropic (1, 3, 2, 3), (1, 3, 5, 3) and (2, 3, 4, 3) sequences. 6.34
6.21 Logarithmic write spectrum of maxentropic (1, 5, 2, 5), (1, 5, 6, 5) and (2, 5, 4, 5) sequences. 6.35
6.22 Linear write spectrum of maxentropic (1, 5, 2, 5), (1, 5, 6, 5) and (2, 5, 4, 5) sequences. 6.35

7.1 FSM encoder for the (1, 2, 3, 3) code 7.2
7.2 Linear power spectrum for the (1, 2, 3, 3) code for $p=0.5$ 7.13
7.3 Logarithmic power spectrum for the (1, 2, 3, 3) code 7.15
7.4 Linear power spectrum for the (1, 2, 3, 3) code 7.15
7.5 Logarithmic power spectrum for the (1, 2, 7, 7) code 7.16
7.6 Linear power spectrum for the (1, 2, 7, 7) code 7.16

iii
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.7</td>
<td>Logarithmic power spectrum for the (2, 3, 6, 6) code</td>
</tr>
<tr>
<td>7.8</td>
<td>Linear power spectrum for the (2, 3, 6, 6) code</td>
</tr>
<tr>
<td>7.9</td>
<td>Logarithmic power spectrum for the (2, 4, 4, 4) code</td>
</tr>
<tr>
<td>7.10</td>
<td>Linear power spectrum for the (2, 4, 4, 4) code</td>
</tr>
<tr>
<td>7.11</td>
<td>Logarithmic power spectrum for the (1, 2, 5, 5, 5) code</td>
</tr>
<tr>
<td>7.12</td>
<td>Linear power spectrum for the (1, 2, 5, 5, 5) code</td>
</tr>
<tr>
<td>7.13</td>
<td>Logarithmic $S_W(f)$ for maxentropic (1, 2, 3, 3) sequences, $S_W(f)$ for the (1, 2, 3, 3) code and $S_C(f)$ for the (1, 2, 3, 3) code for $p=0.5$</td>
</tr>
<tr>
<td>7.14</td>
<td>Linear $S_W(f)$ for maxentropic (1, 2, 3, 3) sequences, $S_W(f)$ for the (1, 2, 3, 3) code and $S_C(f)$ for the (1, 2, 3, 3) code for $p=0.5$</td>
</tr>
<tr>
<td>7.15</td>
<td>Logarithmic $S_W(f)$ for maxentropic (1, 2, 7, 7) sequences, $S_W(f)$ for the (1, 2, 7, 7) code and $S_C(f)$ for the (1, 2, 7, 7) code for $p=0.5$</td>
</tr>
<tr>
<td>7.16</td>
<td>Linear $S_W(f)$ for maxentropic (1, 2, 7, 7) sequences, $S_W(f)$ for the (1, 2, 7, 7) code and $S_C(f)$ for the (1, 2, 7, 7) code for $p=0.5$</td>
</tr>
<tr>
<td>7.17</td>
<td>Logarithmic $S_W(f)$ for maxentropic (2, 3, 6, 6) sequences, $S_W(f)$ for the (2, 3, 6, 6) code and $S_C(f)$ for the (2, 3, 6, 6) code for $p=0.5$</td>
</tr>
<tr>
<td>7.18</td>
<td>Linear $S_W(f)$ for maxentropic (2, 3, 6, 6) sequences, $S_W(f)$ for the (2, 3, 6, 6) code and $S_C(f)$ for the (2, 3, 6, 6) code for $p=0.5$</td>
</tr>
<tr>
<td>7.19</td>
<td>Logarithmic $S_W(f)$ for maxentropic (2, 4, 4, 4) sequences, $S_W(f)$ for the (2, 4, 4, 4) code and $S_C(f)$ for the (2, 4, 4, 4) code for $p=0.5$</td>
</tr>
<tr>
<td>7.20</td>
<td>Linear $S_W(f)$ for maxentropic (2, 4, 4, 4) sequences, $S_W(f)$ for the (2, 4, 4, 4) code and $S_C(f)$ for the (2, 4, 4, 4) code for $p=0.5$</td>
</tr>
<tr>
<td>7.21</td>
<td>Logarithmic $S_W(f)$ for the (1, 2, 5, 5, 5) code and $S_C(f)$ for the (1, 2, 5, 5, 5) code for $p=0.5$</td>
</tr>
</tbody>
</table>
List of figures

7.22 Linear $S_W(f)$ for the (1, 2, 5, 5, 5) code and $S_C(f)$ for the (1, 2, 5, 5, 5) code for $p=0.5$ 7.24

7.23 Measured logarithmic write spectrum for the (1, 2, 7, 7) code 7.25
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE No.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Selected values of channel capacity for ( \alpha, \beta, \gamma, \delta, \xi ) sequences</td>
<td>3.34</td>
</tr>
<tr>
<td>4.1</td>
<td>Encoder table for the ((1, 2, 3, 3)) code</td>
<td>4.8</td>
</tr>
<tr>
<td>4.2</td>
<td>Decoder table for the ((1, 2, 3, 3)) code</td>
<td>4.8</td>
</tr>
<tr>
<td>4.3</td>
<td>Summary of states merged and their associated edges</td>
<td>4.14</td>
</tr>
<tr>
<td>4.4</td>
<td>Encoder table for the ((1, 2, 7, 7)) code</td>
<td>4.16</td>
</tr>
<tr>
<td>4.5</td>
<td>Decoder table for the ((1, 2, 7, 7)) code</td>
<td>4.17</td>
</tr>
<tr>
<td>4.6</td>
<td>Hex decoder table for the ((1, 2, 7, 7)) code</td>
<td>4.18</td>
</tr>
<tr>
<td>4.7</td>
<td>Summary of states merged and their associated edges</td>
<td>4.21</td>
</tr>
<tr>
<td>4.8</td>
<td>Encoder table for the ((2, 3, 6, 6)) code</td>
<td>4.23</td>
</tr>
<tr>
<td>4.9</td>
<td>Decoder table for the ((2, 3, 6, 6)) code</td>
<td>4.24</td>
</tr>
<tr>
<td>4.10</td>
<td>Summary of states merged and their associated edges</td>
<td>4.28</td>
</tr>
<tr>
<td>4.11</td>
<td>Encoder table for the ((2, 4, 4, 4)) code</td>
<td>4.30</td>
</tr>
<tr>
<td>4.12</td>
<td>Decoder table for the ((2, 4, 4, 4)) code</td>
<td>4.31</td>
</tr>
<tr>
<td>4.13</td>
<td>Hex decoder table for the ((2, 4, 4, 4)) code</td>
<td>4.32</td>
</tr>
<tr>
<td>4.14</td>
<td>Summary of states merged and their associated edges</td>
<td>4.37</td>
</tr>
<tr>
<td>4.15</td>
<td>Encoder table for the ((1, 2, 5, 5, 5)) code</td>
<td>4.39</td>
</tr>
<tr>
<td>4.16</td>
<td>Decoder table for the ((1, 2, 5, 5, 5)) code</td>
<td>4.40</td>
</tr>
<tr>
<td>5.1</td>
<td>Values of EMF for the new codes</td>
<td>5.7</td>
</tr>
<tr>
<td>7.1</td>
<td>Encoder table for the ((1, 2, 3, 3)) code</td>
<td>7.2</td>
</tr>
<tr>
<td>7.2</td>
<td>Discrete components for some of the new code</td>
<td>7.26</td>
</tr>
<tr>
<td>7.3</td>
<td>Discrete components for the ((1, 2, 7, 7)) code</td>
<td>7.26</td>
</tr>
<tr>
<td>Table</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>7.4</td>
<td>Discrete components for the (2, 4, 4, 4) code</td>
<td>7.26</td>
</tr>
<tr>
<td>8.1</td>
<td>Summary of the new modulation codes</td>
<td>8.2</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Source-symbol sequence.</td>
</tr>
<tr>
<td>A_i</td>
<td>The i-th element of the source-word.</td>
</tr>
<tr>
<td>A</td>
<td>Connection matrix corresponding to a graph.</td>
</tr>
<tr>
<td>a</td>
<td>Minimum runlength of zeros.</td>
</tr>
<tr>
<td>\mathcal{A}</td>
<td>Source Alphabet.</td>
</tr>
<tr>
<td>b</td>
<td>Buffer length.</td>
</tr>
<tr>
<td>b_i</td>
<td>The i-th element of ( \mathcal{A}^m ).</td>
</tr>
<tr>
<td>B</td>
<td>Transition matrix corresponding to a Markov model.</td>
</tr>
<tr>
<td>\beta</td>
<td>Minimum runlength of ones.</td>
</tr>
<tr>
<td>c</td>
<td>Code-symbol sequence.</td>
</tr>
<tr>
<td>c_i</td>
<td>The i-th element of ( \mathcal{S} ).</td>
</tr>
<tr>
<td>C</td>
<td>Maximum absolute value of ( r_\ell ) for any ( \ell ).</td>
</tr>
<tr>
<td>C_i</td>
<td>The i-th correlation factor.</td>
</tr>
<tr>
<td>C'_i</td>
<td>The i-th element of the code-word.</td>
</tr>
<tr>
<td>C_+</td>
<td>Maximum positive charge deviation.</td>
</tr>
<tr>
<td>C_-</td>
<td>Maximum negative charge deviation.</td>
</tr>
<tr>
<td>\mathcal{S}</td>
<td>Code-symbol alphabet.</td>
</tr>
<tr>
<td>d</td>
<td>d-constraint in a ((d, k)) code.</td>
</tr>
<tr>
<td>D</td>
<td>D-transform ((e^{i\omega})).</td>
</tr>
<tr>
<td>\mathcal{D}</td>
<td>The diagonal matrix (\text{diag}{\pi(1), \ldots, \pi(I)}).</td>
</tr>
<tr>
<td>\Delta</td>
<td>Determinant of a matrix.</td>
</tr>
<tr>
<td>\mathcal{D}_s</td>
<td>Source-word dictionary.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$\mathcal{D}_c$</td>
<td>Unconstrained code-word dictionary.</td>
</tr>
<tr>
<td>$\mathcal{D}_h$</td>
<td>Constrained code-word dictionary.</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Maximum runlength of ones.</td>
</tr>
<tr>
<td>$e$</td>
<td>Probability for a channel error to occur.</td>
</tr>
<tr>
<td>$E(T_j)$</td>
<td>Mean value of the runlength process ${T_j}$.</td>
</tr>
<tr>
<td>$E_i$</td>
<td>The $i$-th state-transition matrix.</td>
</tr>
<tr>
<td>$f$</td>
<td>Frequency.</td>
</tr>
<tr>
<td>$f_c$</td>
<td>Code-symbol frequency.</td>
</tr>
<tr>
<td>$f_s$</td>
<td>Data rate.</td>
</tr>
<tr>
<td>$F$</td>
<td>Frequency response of receive filter.</td>
</tr>
<tr>
<td>$F(\omega)$</td>
<td>Fourier transform.</td>
</tr>
<tr>
<td>$g(\beta,u)$</td>
<td>Function which determines the terminal state of the sequence $\beta$ which starts from the state $u$.</td>
</tr>
<tr>
<td>$G$</td>
<td>Graph of Markov model used to construct an encoder.</td>
</tr>
<tr>
<td>$G(D)$</td>
<td>One-step state transition matrix.</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Maximum runlength of zeros.</td>
</tr>
<tr>
<td>$\Gamma_i$</td>
<td>The $i$-th output matrix.</td>
</tr>
<tr>
<td>$H$</td>
<td>Channel capacity.</td>
</tr>
<tr>
<td>$i$</td>
<td>$\sqrt{-1}$.</td>
</tr>
<tr>
<td>$I$</td>
<td>Identity matrix.</td>
</tr>
<tr>
<td>$j$</td>
<td>Dummy index.</td>
</tr>
<tr>
<td>$k$</td>
<td>$k$-constraint in a $(d, k)$ code.</td>
</tr>
<tr>
<td>$K$</td>
<td>Cardinality of $\mathcal{A}$.</td>
</tr>
<tr>
<td>$\chi$</td>
<td>Maximum of $\alpha$ or $\beta$.</td>
</tr>
</tbody>
</table>
### List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l )</td>
<td>Number of symbols in a sequence.</td>
</tr>
<tr>
<td>( \lambda_x )</td>
<td>Largest real root of a polynomial.</td>
</tr>
<tr>
<td>( m )</td>
<td>Dummy index.</td>
</tr>
<tr>
<td>( m_c )</td>
<td>The codeword mean value.</td>
</tr>
<tr>
<td>( M_u )</td>
<td>The number of ones in the source-word ( b_u ).</td>
</tr>
<tr>
<td>( n )</td>
<td>Dummy index.</td>
</tr>
<tr>
<td>( N )</td>
<td>Digital sum variation.</td>
</tr>
<tr>
<td>( N_0 )</td>
<td>Noise power per unit of frequency.</td>
</tr>
<tr>
<td>( N(\ell) )</td>
<td>Number of sequences of length ( \ell ).</td>
</tr>
<tr>
<td>( N_k(\ell) )</td>
<td>Number of sequences of length ( \ell ) with ( r_\ell = k ).</td>
</tr>
<tr>
<td>( p )</td>
<td>Probability for an input data bit to be a one.</td>
</tr>
<tr>
<td>( p' )</td>
<td>Probability of transition between symbols.</td>
</tr>
<tr>
<td>( p_x )</td>
<td>Probability for one channel error to result in ( x ) data errors.</td>
</tr>
<tr>
<td>( p(1) )</td>
<td>Equilibrium probability of a +1 or a -1 in the ( {x(mT_c)} ) process.</td>
</tr>
<tr>
<td>( P_{Tc} )</td>
<td>Modulating pulse function.</td>
</tr>
<tr>
<td>( p^* )</td>
<td>Reciprocal polynomial.</td>
</tr>
<tr>
<td>( \mathcal{P} )</td>
<td>Set of states.</td>
</tr>
<tr>
<td>( \pi )</td>
<td>Row vector containing the first-order state probabilities.</td>
</tr>
<tr>
<td>( \Pi )</td>
<td>Transition probability matrix.</td>
</tr>
<tr>
<td>( \Pi_\infty )</td>
<td>Limiting value of the transition probability matrix.</td>
</tr>
<tr>
<td>( q )</td>
<td>Probability for an input data bit to be a zero.</td>
</tr>
<tr>
<td>( q_i )</td>
<td>The ( i )-th source-word probability.</td>
</tr>
<tr>
<td>( R )</td>
<td>Mapping ratio.</td>
</tr>
<tr>
<td>( \mathbb{R} )</td>
<td>Real numbers.</td>
</tr>
<tr>
<td>( r_\ell )</td>
<td>Running digital sum of sequence of length ( \ell ) symbols.</td>
</tr>
</tbody>
</table>
List of symbols

- $s_i$: The $i$-th information source message.
- $S_Y$: Power spectrum of $c(mT_c)$.
- $S_C$: Cosine-squared pulse power spectrum.
- $S_W$: Write spectrum.
- $S_X$: Power spectrum of $x(mT_c)$.
- $\sigma_i$: The $i$-th element of $\mathcal{R}$.

- $t$: Time variable.
- $\tau$: Time constant for AC-coupling network.
- $T$: Codeword period.
- $T_c$: Code-symbol period.
- $T_j$: The $j$-th set of terminal states.
- $T_s$: Source-symbol period.
- $T(z)$: Generating function.

- $u$: A state in a graph.
- $\mathbf{u}$: Row vector of size $K$ with all 1 elements.
- $v$: Approximate eigenvector.
- $V(f)$: Frequency response of parallel-to-serial converter.
- $V^*(f)$: Conjugate transpose of $V(f)$.

- $w$: Column vector with all entries equal to one.
- $\omega$: Angular frequency variable.
- $W$: Write signal.
- $W^{(c)}(f)$: Continuous code-symbol power spectrum.
- $W^{(c)}_c(f)$: Continuous code-word power spectrum.
- $W^{(d)}(f)$: Discrete code-symbol power spectrum.
- $W^{(d)}_c(f)$: Discrete code-word power spectrum.

- $x$: Modulated signal.
List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>Modulated transmit signal.</td>
</tr>
<tr>
<td>$z$</td>
<td>$\exp(i\omega)$.</td>
</tr>
<tr>
<td>$Z$</td>
<td>Time domain.</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Space-mark-ratio.</td>
</tr>
<tr>
<td>$X^T$</td>
<td>Transpose of a matrix.</td>
</tr>
<tr>
<td>$[x]$</td>
<td>The value of $x$ rounded up to the nearest integer.</td>
</tr>
<tr>
<td>$[x]$</td>
<td>The value of $x$ rounded down to the nearest integer.</td>
</tr>
<tr>
<td>$\max_{\ell}[f(\ell)]$</td>
<td>The maximum value of $f(\ell)$ for all possible values of $\ell$.</td>
</tr>
<tr>
<td>BER</td>
<td>Bit error rate.</td>
</tr>
<tr>
<td>EMF</td>
<td>Error multiplication factor.</td>
</tr>
<tr>
<td>Erfc</td>
<td>Complementary Error function.</td>
</tr>
<tr>
<td>S/N</td>
<td>Signal-to-Noise.</td>
</tr>
</tbody>
</table>
There has been and most probably always will be an abundance of stories of men and women who had dreams of how things should be done for the better. Not all such dreams came to fruition; the cold reality of economic viability, difficulty of implementation and conservatism will sort the weak from the tenacious.

There were those who dreamed that sound could be conveyed in a far better way than permitted by the existing analog technology. This was not achieved by some gradual perfecting of an existing mechanism, but by a totally different approach namely digital recording technology, which would wipe away many existing problems at a stroke. It would require many and diverse skills from fields which had not previously had any relevance to audio: computation, laser optics, mass storage, communications theory and error correction.
CHAPTER 1: Introduction

There are few people in the developed world who have not been exposed to digital audio, either directly by the Compact Disc and LaserVision, or indirectly by listening to material which has been digitally recorded.

The principal advantage that the digital implementation has over the analog implementation is that in a well-engineered digital recording system the sole significant degradation takes place at the initial digitization [1]. However, in an analog system the quality is diminished at each stage of signal processing and the number of recording generations is limited.

Coding techniques are used in communication systems to improve the reliability of the communication channel i.e. to decrease the probability of receiving the wrong information. A code is a set of rules for assigning a sequence known as the code (or recorded) sequence to a source (or input) sequence. For the code-sequence to be compatible with the physical characteristics of the transmission channel the code-sequence should comprise special attributes known as channel constraints. Typical channel constraints are frequent transitions between symbols and freedom from a zero-frequency component.

The object of this study was to investigate sequences and develop modulation (or recording) codes suited for binary asymmetrical optical channels complying to certain channel constraints. Write-once optical recording is an example of a channel for which asymmetrical runlength constraints are required.

This dissertation has been divided into two volumes. Volume 1 contains chapters 1 to 8, while appendices A to L are included in volume 2.

In chapter 2, a short summary of optical channels is given followed by a general model of a coding system. The system diagram and notation were kept very close to that presented by Cariolara et al [2], and this notation will be used throughout the text.
Chapter 3 is devoted to the information theory of binary sequences complying to asymmetrical runlength constraints. A new generating function for these sequences is derived and the characteristic equations for determining the channel capacity are derived from the generating function. Although the characteristic equations have previously been determined, they are derived in this chapter from the new generating function, which has previously not been done. It is then shown that \((d, k)\) sequences are a special case of the sequences considered. Some new relations relating certain channel capacities are also derived and the general Markov model for charge-constrained runlength limited binary asymmetrical sequences is presented.

Chapter 4 presents the synthesis of five new modulation codes, each complying to a different set of channel constraints. The Markov models presented in chapter 3 were used as starting points for the synthesis of the new codes. Different code construction algorithms were then applied to the Markov models to find the codes.

In chapter 5 the error behaviour of the new codes is discussed, and in chapter 6 a new expression for the power spectrum of maxentropic binary asymmetrical sequences is derived. The influence of the different parameters on the power spectrum is investigated. These investigations revealed that certain sequences contained spectral nulls at frequencies other than zero and this new property is also discussed.

Chapter 7 provides an example to illustrate the algorithm of Cariolara et al for calculating the power spectrum of modulation codes. The power spectrum of the new codes are then presented.

An overview and discussion of the first seven chapters is given in chapter 8. In this chapter a short comparison is made between the new codes and existing \((d, k)\) codes, results are discussed, and suggestions for further research are made.

Appendix A contains tables of numerical values of channel capacities, corresponding to sequences which comply to different sets of input restrictions.

Appendix B contains illustrations of maxentropic write spectra and cosine-squared
pulse power spectra for selected runlength constraints as calculated from the formulas derived in chapter 6.

Appendix C contains write spectra and cosine-squared pulse power spectra for the codes developed in chapter 4 as calculated with the algorithm of Cariolara et al.

Appendices D to K provide discussions and listings of computer programs that were used during the course of this study, while Appendix L contains a list of the programs on the included floppy disk.
CHAPTER 2

OPTICAL CHANNELS AND CHANNEL CODING

In this chapter a brief description of optical storage and optical fibre systems is given as examples of important optical communication channels. A general model of an encoder is then presented, and the signals that occur at different points within the system are defined. The notation adopted in this chapter will be used throughout the text.

2.1 OPTICAL CHANNELS

During the past decade the optical disc has become increasingly popular. It is in particular the Compact Disc Digital Audio System that has become commonplace in the home. This is due to it's compact size (12cm in diameter) and a high signal-to-noise ratio of more than 90dB. The larger diameter optical disc is becoming
increasingly popular for the storage of large amounts of data and video information. Together with the advantage of the fast random accessibility inherent to disc systems, the Compact Disc offers the possibility of application as a huge ROM for data distribution in fields other than audio information. Write-once optical recording systems will mainly be used in data storage systems in which archival aspects or mass storage requirements prevail. In the recording of the information of read-only systems, a photosensitive master disc is exposed with light. This is done with an argon-ion laser (wavelength = 457.9 nm) with high numerical aperture (NA \approx 0.65) lenses. This permits the recording of very small depressions, although the light spot used for read-out, is substantially larger. In write-once (PROM) or erasible recording of optical discs, a low-cost semiconducting GaAlAs laser (wavelength = 800 nm) is used both for reading and writing. The advantage of using GaAlAs laser diodes in optical recording systems is that their light intensity can be modulated by modulating the current through the diode structure [3]. The modulated laser light is used to heat a sensitive layer on the disc to form the written mark. Although very small marks can be written, 1 \mu m seems to be the practical lower limit of the mark size [4]. In write-once optical data storage, one is thus faced with an asymmetry between marks and non-marks [4]. This is due to the fact that with increasing information density, the lengths of the non-marks diminish, but the marks cannot be made arbitrarily small. In this dissertation the non-marks will be represented by the runlength of zeros and the marks by the runlength of ones. In write-once optical data storage the minimum non-marks are smaller than the minimum marks and therefore the minimum runlength of zeros will be shorter than the minimum runlength of ones, thus the requirement for asymmetrical runlengths. With the Compact Disc the minimum mark size that is detectable is about 0.5 \mu m [3] and the spot size of the laser 1 \mu m [4]. Thus the ratio of minimum runlength of zeros to ones should be 1:2 in the case of write-once optical recording. Van Uijen et al [4] provided experimental evidence that coding techniques adapted to such an asymmetric channel would successfully combat the effect of intersymbol interference. In this dissertation a specific investigation of coding techniques and codes that are suitable for binary asymmetric channels is presented.

In optical fibre communication, there may also be some asymmetry present in the optical amplifiers. Furthermore, in injection lasers used in optical fibre
communication, spontaneous emission continues even after the applied current pulse falls to zero [5]. This is due to the fact that the injected carriers in the active layer of an injection laser remain for a period comparable with the carrier lifetime. The carrier accumulation effect degrades the modulation characteristics because some carriers generated by the preceding pulse may still remain when successive pulses are applied. It may therefore also be feasible to use the asymmetrical codes, discussed in this dissertation, in optical fibre communication systems.

2.2 MODEL OF A CODING SYSTEM

Two fundamental classes of signals can be found in synchronous data transmission systems [2]:

(i) Continuous-time real-valued signals, where the time-domain variables, i.e. moments in time at which the signals can undergo a change in amplitude, and range of possible signal amplitudes, can be described by the real numbers, \( \mathbb{R} \).

(ii) Discrete-time finite-valued signals, in which the time domain is of the form \( \mathbb{Z}(T) = \{\ldots, -T, 0, T, 2T, \ldots\} \), i.e. the signals can only undergo a change in amplitude at the times which are elements of the set \( \mathbb{Z}(T) \). The possible amplitudes of the signal form a finite set \( \mathcal{A} \), which is referred to as an alphabet when it consists of symbols, and a dictionary when it consists of groups of symbols. The quantity \( T \) represents the spacing between two consecutive values of the signal, and \( f = \frac{1}{T} \) represents the rate of the signal in number of values per second.

The usual notation for the m-th sample of a discrete-time finite-valued signal \( c \) is \( c_m = c(mT_c) \), where \(-\infty < m < +\infty\), which gives the signal value at the time \( mT_c \). The notation \( c(mT_c) \) will be used in subsequent chapters.

Fig. 2.1 shows a model of an encoder for a digital communications system [2]. The decoder is the image of this diagram:
The following signals occur in the system of Fig. 2.1:

(i) The source-symbol sequence, or the data transmitted by the source, which is the signal $a(t)$, $t \in Z(T_s)$, where $T_s$ is the source-symbol period. Each source-symbol is an element of the alphabet $\mathcal{A} = \{b_1, \ldots, b_K\}$ which is usually the binary alphabet, in which case $K = 2$ and $\mathcal{A} = \{0, 1\}$.

(ii) The data is framed into blocks of a fixed length $m$ by the serial-to-parallel converter (S/P) to yield the source-word sequence:

$$A(t) = [A_1(t), \ldots, A_m(t)], \; t \in Z(T).$$

Source words are chosen from the dictionary $\mathcal{D}_s$, which consists of all the possible groups of length $m$ formed by the elements of $\mathcal{A}$, i.e. $D_s$ is the $m$-th extension of $\mathcal{A}$, or $\mathcal{D}_s = \mathcal{A}^m$.

(iii) The word coder maps every source-word onto a code-word of length $n$ to yield the code-word sequence,

$$C(t) = [C_1(t), \ldots, C_n(t)], \; t \in Z(T).$$

Figure 2.1 Model of a coding system
CHAPTER 2: Optical channels

The dictionary of unconstrained code words, \( \mathcal{D}_c \), is in general a subset of the \( n \)-th extension of the code-symbol alphabet, \( \mathcal{G} \), or \( \mathcal{D}_c \subset \mathcal{G}^n \).

(iv) The parallel-to-serial converter deframes the code-word sequence \( \mathcal{C}(t) \) into code symbols. The result is the code-symbol sequence, \( \mathcal{C}(t), t \in \mathbb{Z}(T_c) \). For proper synchronization, \( T = mT_s = nT_c \).

(v) The output of the digital modulator is the modulated signal, \( x(t); t \in \mathbb{Z}(T_c) \), which is the signal that is sent through the channel.

The coding scheme refers to all the processes from the serial-to-parallel converter to the parallel to serial converter in Fig. 2.1. The mapping ratio of a code is defined as

\[
\mathcal{R} = \frac{m}{n}
\]  

(2.3)

If all the code words complying with the channel constraints, \( \mathcal{D}_h \), where \( \mathcal{D}_c \subset \mathcal{D}_h \subset \mathcal{G}^n \) are used by the word coder, i.e. if \( \mathcal{D}_c = \mathcal{D}_h \), then the modulation code is referred to as a saturated code. If the code is not a saturated code, the remaining words can be used to transmit framing information or to do error monitoring [6].

The read waveform is processed through the channel where the presence or absence of a transition within each signalling element is detected by a variable-frequency clock. The available time for detection is called the detection window and is determined by the width of the signalling element in terms of the data bit width \( (T_s) \). Thus, the detection window is completely determined by the rate of the code.
CHAPTER 3

GENERATING FUNCTION, MARKOV MODELS AND CHANNEL CAPACITY

In this chapter information theoretical aspects of sequences that are suitable for write-once optical storage systems is taken under consideration. Paragraph 3.1 describes the input restrictions that is common for such communications channels. The concepts of information and channel capacity of discrete noiseless channels are considered in paragraph 3.2. The generating function, the characteristic equation and some relations for asymmetrical sequences are derived in section 3.3. All results and methods discussed in this section are new unless stated otherwise.

3.1 INPUT RESTRICTIONS

In order to ensure that the discrete communications channel is able to carry the coded
sequence with a low probability of error, the code sequence should generally comply to one or both of the following input restrictions [1]:

(i) Spectral null at zero frequency:

Codes with a spectral null at zero frequency, are called dc-balanced codes, and are employed to counter the effects of the channel's low-frequency cut-off characteristics due to isolating transformers, coupling networks, etc. We would like the low-frequency content to be as small as possible. This is necessary because, firstly, low-frequency signals are used to control the servo systems for track following and focusing [7]. Secondly, fingermarks etc. on disks cause the amplitude and average level of the read signal to fall below the threshold level, causing read errors. These errors can be avoided by using a filter to remove the low-frequency components, provided that the information signal itself contains no low-frequency components. The instantaneous accumulated charge or running digital sum (RDS) of a code sequence of length \( l \) is defined as:

\[
 r_\ell = \sum_{m=1}^{l} c(mT_c),
\]

where \( c(mT_c) \) is the code-symbol sequence. The difference between the maximum and minimum values of \( r_\ell \) for all values of \( l \) is called the digital sum variation (DSV). The parameter C, sometimes called the maximum accumulated "charge", is defined as [8]:

\[
 C = \max_{\ell} | r_\ell |.
\]

If symmetry exists with respect to \( r_\ell = 0 \), then \( C = \frac{\text{DSV}}{2} \).

Pierobon [9] proved that the spectrum of the code-symbol sequence \( \{c(mT_c)\} \) vanishes at zero frequency if and only if the RDS, or C, is bounded.
(ii) Runlength restrictions:

A minimum runlength constraint is used to limit the highest transition frequency and thus has a bearing on the intersymbol interference (ISI) when the sequence is transmitted over a bandwidth-limited channel. The coded sequence should also have frequent transitions between symbols, i.e. a maximum runlength constraint, to enable the read clock to be derived from the coded sequence.

3.2 INFORMATION AND CHANNEL CAPACITY.

The output of a discrete information source can be viewed as consisting of messages, each consisting of a sequence of symbols. Not all messages contain the same amount of information [10]. If the information source can emit one of q possible messages \( s_1, s_2, \ldots, s_q \), with probabilities of occurrence \( p_1, p_2, \ldots, p_q \), then the amount of information conveyed by the message \( s_k \) is defined as:

\[
I(s_k) = \log \left( \frac{1}{p_k} \right).
\]

It is clear from (3.3) that a message with a high probability of occurrence has a small information content, and vice versa. The base of the logarithm in (3.3) determines the unit assigned to the information. If the base is 2, the unit is bit. If two possible messages \( s_1 \) and \( s_2 \) occur with equal probability \( p_1 = p_2 = 0.5 \), then each message conveys \(-\log(0.5) = 1\) bit of information. This property makes the bit a suitable measure of information in cases where the information source is binary, emitting one of two binary digits (0 or 1) with equal probability.

If the information source emits a long sequence of \( \ell \) symbols in a statistically independent manner, then the symbol \( s_k \) will occur approximately \( p_k \ell \) times in the sequence. If we treat symbols as messages of length one, then the information content of the symbol \( s_k \) is \(-\log(p_k)\). Hence the \( p_k \ell \) occurrences of \( s_k \) contributes an
CHAPTER 3: Generating function, Markov Models...

information content of \(-p \ell \log_2 (p)\) bits. If the source emits \(q\) possible symbols, then the total information content of the sequence is:

\[
I_{\text{total}} = \sum_{i=1}^{q} \ell p_i \log_2 \left( \frac{1}{p_i} \right) \quad \text{bits.} \tag{3.4}
\]

The channel capacity of a noiseless, discrete, input restricted communications channel is defined as [11]:

\[
H = \lim_{\ell \to \infty} \log_2 \frac{N(\ell)}{\ell} \quad \text{bits/symbol,} \tag{3.5}
\]

where \(N(\ell)\) denotes the number of possible unique sequences of length \(\ell\) satisfying the input restrictions of the channel. As the unit indicates, \(H\) denotes an upper bound on the average information content of a symbol in long sequences. When employing a code, the amount of information conveyed with each symbol depends on the mapping ratio, \(R\), as defined in (2.3). Thus \(H\) is the highest value of \(R\) that can be achieved by any encoder that generates a sequence suitable for transmission over the channel. The sequence utilization efficiency of a coding strategy is defined accordingly as [8]:

\[
\eta = \frac{R}{H} \times 100\%. \tag{3.6}
\]

3.3 BINARY ASYMMETRICAL RUNLENGTH LIMITED SEQUENCES

Asymmetrical runlength limited sequences have previously been characterised by four parameters \((d_0, k_0)\) and \((d_1, k_1)\), \(d_0, d_1 \geq 0\) and \(k_0 \geq d_0, k_1 \geq d_1\), which describe the constraints on alternate runlengths of 'zeros' and 'ones', in the NRZI format, respectively [1]. We prefer to investigate the class of binary NRZ,
runlength limited channels defined by a 4-tuple \((a, \beta, \gamma, \delta)\), where

(i) \(a \triangleq \) the minimum runlength of 0's between runs of 1's,

(ii) \(\beta \triangleq \) the minimum runlength of 1's between runs of 0's,

(iii) \(\gamma \triangleq \) the maximum runlength of 0's between runs of 1's, and

(iv) \(\delta \triangleq \) the maximum runlength of 1's between runs of 0's.

These parameters are the same as the \((d, k, e, \ell)\)-parameters of Lee [12]. The reason for not using the \((d, k, e, \ell)\)-parameters is to avoid confusion when making comparisons to NRZI \((d, k)\)-parameters.

The general Markov model for \((a, \beta, \gamma, \delta)\) sequences is shown in Fig. 3.1. The state variables are the entry symbol and the current runlength of that symbol.

![Figure 3.1 Markov model for \((a, \beta, \gamma, \delta)\) sequences](image-url)

3.5
We define the space-mark-ratio ($\zeta$) as the ratio of the minimum non-mark to the minimum mark, by:

$$\zeta = \frac{\alpha}{\beta}. \quad (3.7)$$

### 3.3.1 Generating function for $(a, \beta, \gamma, \delta)$ sequences

We start by deriving the generating function for $(a, \beta, \gamma, \delta)$ sequences. Similar techniques were used by Forsberg [13] for deriving the generating function for $(d, k)$ and $(k)$ sequences.

We start by enumerating those $(a, \beta, \gamma, \delta)$ sequences which start and end with zeros. Therefore we start by choosing elements from sets A and B,

$$A = \{ (00\ldots0), (00\ldots0), \ldots, (00\ldots0) \}, \quad (3.8)$$

$$B = \{ (11\ldots1), (11\ldots1), \ldots, (11\ldots1) \}, \quad (3.9)$$

alternately, always starting and ending in A.

Let

$$a = z^\alpha + z^{\alpha+1} + \ldots + z^\gamma, \quad (3.10)$$

$$b = z^\beta + z^{\beta+1} + \ldots + z^\delta. \quad (3.11)$$

Then the enumerating generating function for determining the number of such sequences is:

$$\tilde{T}(z) = a + aba + ababa + ... \quad (3.12)$$

$$= a[1 + ab + a^2 b^2 + ... ]$$

$$= a/(1 - ab) \quad (3.13)$$
where the number of b's in a term of (3.12), is the number of runs of consecutive 1's in the sequence. Substituting for a and b from (3.10) and (3.11) into (3.13):

\[ \tilde{T}(z) = \frac{(z^a + z^{a+1} + \ldots + z^\gamma)}{[1 - (z^a + z^{a+1} + \ldots + z^\gamma)(z^\beta + z^{\beta+1} + \ldots + z^\delta)]}. \]  

(3.14)

After multiplying the denominator and numerator of (3.14) with \((1-z)^2\):

\[ \tilde{T}(z) = \frac{(1-z)(1-z)(z^a + z^{a+1} + \ldots + z^\gamma)(1-z)^2}{(1-z)^2 - [(1-z)(z^a + z^{a+1} + \ldots + z^\gamma)][(1-z)(z^\beta + z^{\beta+1} + \ldots + z^\delta)]} \]  

(3.15)

We are looking for the generating function for the maximum number of \((a, b, \gamma, \delta)\) sequences of length \(l\), for which an arbitrary catenation of code words also satisfy the channel constraints. As described in the optimisation problem by Forsberg [13], the numbers of zeros and ones will be evenly distributed between the beginning and the end of the code words. The sequences for which the generating function, \(\tilde{T}(z)\), was derived, start and end with zeros. We allow the codeword to start with \(b\) ones followed by \(\delta-j\) ones or less and end with \(j = \left\lfloor \frac{\delta-b}{2} \right\rfloor\) or less ones. However, the code-word can also start and end with zeros and to get the maximum set of codewords we also allow
the code words to start with $\alpha$ zeros followed by $i = \left\lceil \frac{i-\alpha}{2} \right\rceil$ or less zeros and end with $\gamma-i$ or less zeros. We build up the generating function by starting with the sequence which begins and ends with zeros denoted by $T(z)$. To the beginning of this sequence we can pad either $\beta$ to $\delta-j$ ones or $\alpha$ to $i$ zeros plus $\beta$ to $\delta$ ones. That gives the factor,

$$
(z^\beta + z^{\beta+1} + \ldots + z^{\delta-j}) + (z^\beta + z^{\beta+1} + \ldots + z^\delta)(z^{\alpha} + z^{\alpha+1} + \ldots + z^1)
$$

$$
= \frac{z^\beta - z^\delta \cdot j + 1}{1-z} + \frac{z^\beta - z^\delta \cdot i + 1}{1-z}.
$$

(3.16)

To the end of the sequence we can pad either $0$ to $j$ ones or $\beta$ to $\delta$ ones plus zero to $\gamma-i$ zeros. That gives the factor,

$$
(z^0 + z^1 + \ldots + z^j) + (z^\beta + z^{\beta+1} + \ldots + z^\delta)(z^0 + z^1 + \ldots + z^{\gamma-i})
$$

$$
= \frac{1-z^j + 1}{1-z} + \frac{z^\beta - z^\delta \cdot i + 1}{1-z}.
$$

(3.17)

We can now obtain the total generating function, $T(z)$, as the product of (3.15), (3.16) and (3.17):

$$
T(z) = \frac{(1-z)(z^{\alpha} - z^{\gamma+1})}{1-2z + z^2 + z^{\alpha+\beta} + z^{\alpha+\delta+1} + z^{\beta+\gamma+\delta+2}}
$$

$$
\times \left( \frac{z^\beta - z^\delta \cdot j + 1}{1-z} + \frac{(z^\beta - z^\delta + 1)(z^{\alpha} - z^{i+1})}{(1-z)^2} \right)
$$

$$
\times \left( \frac{1-z^j + 1}{1-z} + \frac{(z^\beta - z^\delta + 1)(1-z) - z^{\gamma-i+1}}{(1-z)^2} \right),
$$

(3.18)

for sequences with length $\ell$ symbols, where $\ell > \gamma+\delta$. A more complex generating
function for the number of \((d, k, e, l)\) sequences with Hamming weight \(w\) was derived by Lee [12]. The noiseless capacity for a binary \((d, k, e, l)\) input-restricted channel was also derived by Lee [12] using matrix methods.

3.3.2 Generating function - relationship to channel capacity

In this section the relationship between the generating function and the channel capacity of Shannon [11] will be summarised - see eg. Forsberg [13]. To see the relationship of this approach to that of Shannon, observe that

\[
T(z) = \frac{q(z)}{p(z)}
\]  

(3.19)

as the ratio of two polynomials that has a partial fraction expansion. Distinct roots are assumed and that the degree of \(p(z)\) is \(m\). The denominator polynomial \(p(z)\) can be written as a product of factors, one factor for each root, as

\[
p(z) = K \prod_{i=0}^{m} (1-\lambda_i z)
\]  

(3.20)

and the partial fraction of \(T(z)\) becomes

\[
T(z) = \sum_{i=0}^{m} \frac{A_i}{1-\lambda_i z}.
\]  

(3.21)

By observing that \(1/(1-\lambda_i z)\) can be expressed as a geometric series we can write \(T(z)\) as

\[
T(z) = A_1 (1 + \lambda_1 z + \lambda_1^2 z^2 + ...) + A_2 (1 + \lambda_2 z + \lambda_2^2 z^2 + ...) \\
+ ... + A_m (1 + \lambda_m z + \lambda_m^2 z^2 + ...).
\]  

(3.22)

We are interested in the coefficient of \(z^n\), denoted by \(d_n = [z^n]T(z)\), and we therefore write \(T(z)\) as
\[ T(z) = (A_1 + ... + A_m) + (A_1 \lambda_1 + ... + A_m \lambda_m)z + ... + (A_1 \lambda_1^n + ... + A_m \lambda_m^n)z^n + ... \] (3.23)

The coefficient \( d_n \) is now seen to be

\[ d_n = (A_1 \lambda_1^n + ... + A_m \lambda_m^n). \] (3.24)

Assume \( \lambda_1 \) to be the largest of the parameters \( \lambda_i \), and write \( d_n \) as

\[ d_n = A_1 \lambda_1^n \left[ 1 + \frac{A_2}{\lambda_1} \left( \frac{\lambda_2}{\lambda_1} \right)^n + ... + \frac{A_m}{\lambda_1} \left( \frac{\lambda_m}{\lambda_1} \right)^n \right]. \] (3.25)

Since \( \lambda_1 \) is the largest of all \( \lambda_i \), all the factors \( \left( \frac{\lambda_i}{\lambda_1} \right)^n \) will approach zero as \( n \) gets larger. We therefore get an asymptotic relationship as

\[ d_n \sim A_1 \lambda_1^n. \] (3.26)

The channel capacity is then given by [11]

\[ H = \lim \frac{\log_2 (d_n)}{n} = \lim \frac{\log_2 A_1 + n \log_2 \lambda_1}{n} = \log_2 \lambda_1. \] (3.27)

Thus the capacity of the sequences is determined by the largest \( \lambda_1 \), which is the smallest real root of \( p(z) \). That is also the same as the largest real root of the reciprocal polynomial

\[ p^*(z) = z^m p(1/z). \] (3.28)

3.3.3 Channel capacity of \((\alpha, \beta, \gamma, \delta)\) sequences

We will derive the noiseless capacity for the \((\alpha, \beta, \gamma, \delta)\) channel from the generating
function in (3.18).

Theorem 3.1

The noiseless capacity for a binary, \((a, \beta, \gamma, \delta)\), \(\gamma > a \geq 1\) and \(\delta > \beta \geq 1\), NRZ input-restricted channel is given by:

\[ H(a, \beta, \gamma, \delta) = \log_2 \left( \lambda_{a\beta \gamma \delta} \right), \]

where \(\lambda_{a\beta \gamma \delta}\) is the largest real root of the characteristic equation

\[ (z^{2\gamma-\alpha+1}-1)(z^{2\delta-\beta+1}-1) - z^{2\gamma\delta}(z-1)^2 = 0. \]  

Proof:

We observe that \(T(z)\), (3.18), is the ratio of two polynomials. Let \(q(z)\) be the numerator polynomial and \(p(z)\) the denominator polynomial, then the generating function becomes

\[ T(z) = \frac{q(z)}{p(z)}, \]

where, from (3.18),

\[ p(z) = (1-2z+z^{2\gamma-\alpha+1}z^{\alpha+\delta+1}z^{\beta+\gamma+1}z^{2\gamma\delta})(1-z)^3, \]

and

\[ q(z) = (z^{2\gamma-\alpha+1})[(1-z)(z^{\beta-\gamma+1}) + (z^{\beta-\gamma+1})(z^{2\gamma\delta+1})(1-z)^2]. \]

We assume distinct roots and that the degree of \(p(z)\) is \(m\). The reciprocal polynomial for \(\gamma > a \geq 1\) and \(\delta > \beta \geq 1\), is given by:

\[ p^*(z) = z^m p(1/z) \]
As was shown in the previous section, the capacity can be obtained from the reciprocal polynomial's largest real root, say \( \lambda_{a\beta\gamma\delta} \). Thus \( \lambda_{a\beta\gamma\delta} \) is the largest real root of the characteristic equation:

\[
(z^{\gamma+\delta+1}-z^{\gamma+\delta+1})(z^{\gamma+\delta+1}-z^{\gamma+\delta+1}) - z^{\gamma+\delta}(z-1)^2 = 0.
\] (3.35)

The characteristic equation (3.35) was derived by Lee [12] for \((d, k, e, l)\) sequences using matrix methods. Lee [12] presented a few selected values of channel capacity on a graph. For future reference and the sake of completeness, we calculated numerical values of channel capacities for \((a, \beta, \gamma, \delta)\) sequences for an extended range of values of \(a, \beta, \gamma\) and \(\delta\). These values are presented in Table A.1 in Appendix A.

### 3.3.4 Channel capacity of \((d, k)\) sequences

Binary NRZI \((d, k)\) sequences can be viewed as a special case of \((a, \beta, \gamma, \delta)\) sequences, and therefore the characteristic equation of these sequences can be derived from the characteristic equation of \((a, \beta, \gamma, \delta)\) sequences. Next as one test for the correctness of the \((a, \beta, \gamma, \delta)\) sequences' characteristic equation the characteristic equation for \((d, k)\) sequences is derived from it.
Theorem 3.2

The noiseless capacity for a binary \((d, k), k > d \geq 0\), NRZI input-restricted channel is given by:

\[
H_{dk}(d, k) = \log_2 \left( \lambda_{dk} \right),
\]

(3.36)

where \(\lambda_{dk}\) is the largest real root of the characteristic equation

\[
z^{k-d+1} - z^{k+1} = 0.
\]

(3.37)

Proof:

From the characteristic equation of NRZ \((\alpha, \beta, \gamma, \delta)\) sequences, (3.35), given by:

\[
(z^{\gamma-\alpha+1}-1)(z^{\delta-\beta+1}-1)-z^{\gamma+\delta}(z-1)^2 = 0,
\]

(3.38)

we make the following substitution:

\[
\alpha = d+1, \quad \beta = d+1, \quad \gamma = k+1, \text{ and} \quad \delta = k+1.
\]

(3.39) (3.40) (3.41) (3.42)

From (3.38):

\[
(z^{k+1-d-1+1}-1)(z^{k+1-d-1+1}-1)-z^{k+1+k+1}(z-1)^2 = 0,
\]

\[
(z^{k-d+1}-1)^2 z^{2k+2}(z-1)^2 = 0,
\]

\[
(z^{k-d+1}-1)^2 [z^{k+1}(z-1)]^2 = 0,
\]
[z^{k-d+1}-1-z^{k+1}(z-1)][z^{k-d+1}+1+z^{k+1}(z-1)] = 0, and

\[ [z^{k-d+1}-1-z^{k+2}+z^{k+1}]z^{k-d+1}+1+z^{k+2}-z^{k+1}] = 0. \] (3.43)

It will now be shown that the first factor of (3.43) has the largest real root for \( k>d \geq 0 \), and therefore the characteristic equation of a NRZI \((d, k)\) sequence is given by

\[ z^{k-d+1}-1-z^{k+2}+z^{k+1} = 0. \] (3.44)

The proof will be split into two cases namely for \( k>d>0 \) and \( k>d \) with \( d=0 \). From (3.43) let the second factor be given by:

\[ P_2(z) = z^{k-d+1}+1+z^{k+2}-z^{k+1}. \] (3.45)

For \( k > d > 0 \) the polynomial \( P_2(z) \), (3.45), is arranged in descending powers of \( z \):

\[ P_2(z) = z^{k+2}-z^{k+1}+z^{k-d+1}. \] (3.46)

Descartes' [14] rule of signs indicates that the number of positive real roots is either three or one, because there are three sign changes. Synthetic division is an abbreviated form of division that shortens the so-called "long division" process.

We apply synthetic division by dividing \( P_2(z) \) in (3.46), by \((z-1)\):

\[
\begin{array}{cccccccc}
 & z^{k+2} & z^{k+1} & z^k & \ldots & z^{k-d+2} & z^{k-d+1} & z^{k-d} & \ldots & z^1 & z^0 \\
1 & 1 & -1 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & -1 \\
1 & 0 & \ldots & 0 & 0 & 1 & \ldots & 1 & 1 \\
1 & 0 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 & 0 \\
\end{array}
\] (3.47)

Theorem 9 in Leonhardy [14] states that in the synthetic division of the polynomial
P(z) by (z-b), in which b is positive, if all the numbers on the third line of the synthetic division process are positive or zero, then the equation P(z) = 0 has no root greater than b. From the above theorem it is evident that for the synthetic division of P'(z) that all the numbers on the third line of (3.47) are positive or zero, thus the equation P'(z) = 0 has no root greater than one. In fact the largest real root is one, because the remainder is zero.

For k > d and d=0, let the second factor of (3.43) be given by:

\[ P_2''(z) = z^{k+2} - 1. \] (3.48)

Descartes' [14] rule of signs indicates that the number of positive real roots is one. This is obvious as all the roots of P_2''(z) = 0 lie on the circumference of the unit circle. Thus there is only one positive real root, namely z=1.

From (3.43), let the first factor be given by:

\[ P_1'(z) = z^{k-d+1} - z^{k+2} + z^{k+1}. \] (3.49)

For k > d > 0 the polynomial P_1'(z) is arranged in descending powers of z, and a sign change is performed:

\[ P_1'(z) = z^{k+2} - z^{k+1} - z^{k-d+1} + 1. \] (3.50)

Descartes' [14] rule of signs indicates that the number of positive real roots is either two or zero, because there are two sign changes.

We apply synthetic division by dividing P_1'(z) in (3.50), by (z-1):
CHAPTER 3: Generating function, Markov Models...

The numbers on the third line of (3.51) are not all positive or zero, thus the equation $P_1(z) = 0$ has one positive real root that is greater than one. The other positive real root is $z=1$, because the remainder in the third line is zero.

We now apply synthetic division by dividing $P_1(z)$ in (3.50), by $(z-2)$:

\[
\begin{array}{ccccccccc}
1 & 1 & -1 & 0 & \ldots & 0 & -1 & 0 & \ldots & 0 & 1 \\
1 & 0 & \ldots & 0 & 0 & -1 & \ldots & -1 & -1 \\
1 & 0 & 0 & \ldots & 0 & -1 & -1 & \ldots & -1 & 0 \\
\end{array}
\]

(3.51)

The numbers on the third line of (3.52) are all positive if $k > d$, thus the equation $P_1(z) = 0$ has one positive real root for $1 < z < 2$.

For $k > d$ and $d=0$, let the first factor of (3.43) be given by:

\[
P_1^1(z) = z^{k+2} - 2z^{k+1} + 1.
\]

(3.53)

Descartes' rule of signs indicates that the number of positive real roots is either two or zero. We apply synthetic division by dividing $P_1^1(z)$ in (3.53), by $(z-1)$:
The numbers on the third line of (3.54) are not all positive or zero, thus the equation $P_1''(z) = 0$ has one positive real root greater than one. The other positive real root is $z=1$, because the remainder in the third line is zero.

We apply synthetic division by dividing $P_1''(z)$, (3.53), by $(z-2)$:

\[
\begin{array}{c|cccccc}
  & z^k & z^{k+1} & \ldots & z^1 & z^0 \\
 1 & 1 & -2 & 0 & \ldots & 0 & 1 \\
 & 1 & -1 & \ldots & -1 & -1 \\
\hline
1 & 1 & -1 & \ldots & -1 & 0 \\
\end{array}
\]

(3.54)

The numbers on the third line of (3.55) are all positive or zero, thus the equation $P_1''(z) = 0$ has one positive real root for $1 < z < 2$.

Thus for $k > d \geq 0$, the first factor $P_1(z) = 0$ has the largest real root.

This concludes the proof and the result, i.e. the characteristic equation (3.44) is in agreement with Tang and Bahl [15].

Q.E.D.

**EXAMPLE:**

From the equations (3.39) to (3.42) as well as (3.29) and (3.36), the capacity of a $(d, k) = (2, 7)$ code is equivalent to the capacity of a $(a, \beta, \gamma, \delta) = (3, 3, 8, 8)$ code. As previously mentioned, this is the reason for not using the $(d, k, e, l)$-parameters of Lee [12].
3.3.5 Channel capacity of \((\alpha, \beta, \infty, \infty)\) sequences

If, from Theorem 1, we do not restrict the maximum runlength of zeros and ones we obtain the channel capacity of a NRZ \((\alpha, \beta, \infty, \infty)\), \(\alpha \geq 1\) and \(\beta \geq 1\), sequence:

**Theorem 3.3**

The noiseless channel capacity of NRZ \((\alpha, \beta, \infty, \infty)\), \(\alpha \geq 1\) and \(\beta \geq 1\), sequences is given by:

\[
H(\alpha, \beta, \infty, \infty) = \log_2 (\lambda_{\alpha \beta}),
\]

where \(\lambda_{\alpha \beta}\) is the largest real root of the characteristic equation

\[
z^{\alpha+\beta} - 2z^{\alpha+\beta-1} + z^{\alpha+\beta-2} = 0.
\]

**Proof:**

From the generating function \(\tilde{T}(z)\) of \((\alpha, \beta, \gamma, \delta)\) sequences (3.14), the generating function of \((\alpha, \beta, \infty, \infty)\) sequences is given by:

\[
\tilde{T}(z) = \frac{(z^\alpha + z^{\alpha+1} + \ldots)}{[1 - (z^\alpha + z^{\alpha+1} + \ldots)(z^\beta + z^{\beta+1} + \ldots)]}
\]

\[
= \frac{z^\alpha}{1 - (z^\alpha)(z^\beta)}
\]

\[
= \frac{z^\alpha(1-z)}{(1-z)^2 - z^\alpha z^\beta}.
\]
CHAPTER 3: Generating function, Markov Models

According to Shannon [11], when determining the capacity the limit is taken when \( l \) goes to infinity and thus the ends of the sequence do not affect it. As was shown by Forsberg [13] it is only necessary to consider the denominator of \( \tilde{T}(z) \). Let the denominator of \( \tilde{T}(z) \) be given by:

\[
p(z) = 1 - 2z + z^2 - z^{a + \beta}.
\]  

(3.59)

The reciprocal polynomial for \( a \geq 1 \) and \( \beta \geq 1 \), is given by:

\[
p^*(z) = z^{a + \beta} p(1/z)
\]

(3.60)

Thus \( \lambda_{a\beta} \) is the largest real root of

\[
z^{a + \beta} - 2z^{a + \beta - 1} + z^{a + \beta - 2} - 1 = 0.
\]  

(3.61)

Q.E.D.

The characteristic equation (3.61) was derived by Immink [1] for NRZI \((d_0, k_0)\) and \((d_1, k_1)\) sequences. The characteristic equation (3.61) was also derived from recurrence relations, by Van Uijen et al [4] for sequences complying to a D-constraint \((D \geq 2)\) where \( D \) is a run of consecutive zeros between ones. It is quite clear that \( D \) is equivalent to \( a + \beta - 1 \).

3.3.6 Channel capacity of \((d)\) sequences

We will now derive the channel capacity of \((d)\) sequences from Theorem 3.3 as a test for the correctness of the \((a, \beta, \infty, \infty)\) sequences' characteristic equation.
The noiseless channel capacity of NRZI \((d, \infty)\), \(d \geq 0\), sequences is given by:

\[
H_{dk}(d, \infty) = \log_2(\lambda_d),
\]

where \(\lambda_d\) is the largest real root of the characteristic equation

\[
z^{d+1} - z^d - 1 = 0.
\]

Proof:

Let

\[
\alpha = \beta = d+1.
\]

Substitute (3.64) into (3.61):

\[
z^{2d+2} - 2z^{2d+1} + z^{2d} - 1 = 0
\]

\[
(z^{d+1} - z^d - 1)(z^{d+1} - z^d + 1) = 0.
\]

The first factor of (3.65) has the largest real root \(\lambda_d\) for \(d \geq 0\).

Let the second factor be given by:

\[
P_2'(z) = z^{d+1} - z^d + 1.
\]

Descartes' rule of signs indicates that the number of positive real roots is either two or zero.

We apply synthetic division by dividing \(P_2'(z)\) in (3.66) by \(z-1\):

3.20
The numbers on the third line of (3.67) are all positive or zero, thus the equation $P_2(z) = 0$ has no real roots for $z \leq 1$. Let the first factor be given by:

$$P_1(z) = z^{d+1} - z^d - 1.$$  

(3.68)

Descartes' rule of signs indicates that the number of positive real roots of (3.68) is one.

We apply synthetic division by dividing $P_1(z)$ in (3.68) by $(z-1)$:

$$
\begin{array}{cccccc|c}
1 & & & & & & 1 \\
1 & -1 & 0 & \cdots & 0 & -1 \\
1 & 0 & \cdots & 0 & 0 & \\
\hline
1 & 0 & 0 & \cdots & 0 & -1 \\
\end{array}
(3.69)
$$

The numbers on the third line of (3.69) are not all positive or zero, thus the equation $P_1'(z) = 0$ has one real root for $z > 1$.

We now apply synthetic division by dividing $P_1(z)$ in (3.68) by $(z-2)$:

$$
\begin{array}{cccccc|c}
2 & & & & & & 1 \\
1 & -1 & 0 & \cdots & 0 & -1 \\
2 & 2 & 2 & \cdots & 2^{d-1} & 2^d \\
\hline
1 & 1 & 2 & \cdots & 2^{d-1} & 2^{d-1} \\
\end{array}
(3.70)
$$

3.21
The numbers on the third line of (3.70) are all positive or zero for \( d \geq 0 \), thus the equation \( P_1(z) = 0 \) has no real roots for \( z > 2 \). The equation \( P_1(z) = 0 \) has one real root for \( 1 < z \leq 2 \), thus the first factor of (3.70) has the largest real root.

Q.E.D.

3.3.7 Relations between channel capacity for sequences with different values of parameters \((a, \beta, \gamma, \delta)\)

We will now derive some new relations from Theorems 3.1 to 3.4.

Relation 1:

\[
H(a, \beta, \gamma, \delta) = H(a, \beta, \delta-\beta+a, \gamma+\beta-a).
\]  
(3.71)

Proof:

From (3.30), the characteristic equation of \((a, \beta, \gamma, \delta)\) sequences is given by:

\[
(z^{\gamma+\alpha+1-1})(z^{\delta+\beta+1-1}) - z^{\gamma+\delta}(z-1)^2 = 0.
\]  
(3.72)

Substitute:

\[
\gamma \rightarrow \delta-\beta+a, \quad \text{and} \quad \delta \rightarrow \gamma+\beta-a
\]  
(3.73)

(3.74)

into (3.72):

\[
(z^{\delta-\beta+\alpha+a+1-1})(z^{\gamma+\beta+\alpha+1-1}) - z^{\delta-\beta+\alpha+\gamma+\beta-a}(z-1)^2 = 0
\]  
(3.75)

As the two characteristic equations, (3.72) and (3.75), are the same, their largest real
roots are the same, therefore

\[ H(\alpha, \beta, \gamma, \delta) = H(\alpha, \beta, \delta+\alpha, \gamma+\beta-\alpha). \quad (3.76) \]

Q.E.D.

Relation 2:

\[ H(\alpha, \beta, \gamma, \delta) = H(\beta, \alpha, \gamma+\beta-\alpha, \delta-\beta+\alpha). \quad (3.77) \]

Proof:

From (3.30), the characteristic equation of \((\alpha, \beta, \gamma, \delta)\) sequences is given by:

\[ (z^{\gamma-\alpha+1} - 1)(z^{\delta-\beta+1} - 1) - z^{\gamma+\delta}(z-1)^2 = 0. \quad (3.78) \]

Substitute:

\[ \alpha \rightarrow \beta, \quad \beta \rightarrow \alpha, \quad \gamma \rightarrow \gamma+\beta-\alpha, \text{ and} \]
\[ \delta \rightarrow \delta-\beta+\alpha, \]

into (3.78):

\[ (z^{\gamma+\beta-\alpha-\beta+1} - 1)(z^{\delta-\beta+\alpha-1} - 1) - z^{\gamma+\beta-\alpha+\delta-\beta+\alpha}(z-1)^2 = 0 \]

\[ (z^{\gamma+\alpha+1} - 1)(z^{\delta-\beta+1} - 1) - z^{\gamma+\delta}(z-1)^2 = 0 \quad (3.83) \]

As the two characteristic equations, (3.78) and (3.83), are the same, their largest real roots are the same, therefore

\[ H(\alpha, \beta, \gamma, \delta) = H(\beta, \alpha, \gamma+\beta-\alpha, \delta-\beta+\alpha) \quad (3.84) \]

Q.E.D.
Relation 3:

\[ H(\alpha, \beta, \gamma, \delta) = H(\beta, \alpha, \delta, \gamma). \]  \hspace{1cm} (3.85)

**Proof:**

The proof follows from Relation 1 and Relation 2.

**Example:**

From Relations 1, 2 and 3:

\[ H(1, 2, 7, 7) = H(1, 2, 6, 8), \]
\[ H(1, 2, 6, 8) = H(2, 1, 7, 7), \text{ and} \]
\[ H(1, 2, 6, 8) = H(2, 1, 8, 6). \]

The next two relations were given by Immink [1] for \((d_0, k_0)\) and \((d_1, k_1)\) sequences. Due to the difference in notation these relations are given again for the sake of completeness.

**Relation 4:**

\[ H(2d+2, 0, \infty, \infty) = H_{dk}(d, \infty). \]  \hspace{1cm} (3.86)

**Proof:**

We know that

\[ H_{dk}(d, \infty) = \log_2(\lambda_d), \]  \hspace{1cm} (3.87)

where \(\lambda_d\) is the largest real root of

\[ z^{2d+2} - 2z^{2d+1} + z^{2d} - 1 = 0. \]  \hspace{1cm} (3.88)
CHAPTER 3: Generating function, Markov Models ...

From (3.57), the characteristic equation of \((\alpha, \beta, \infty, \infty)\) sequences, let

\[
\alpha = 2d+2, \text{ and} \\
\beta = 0. 
\]

(3.89) \hspace{1cm} (3.90)

Then the characteristic equation (3.57) becomes:

\[
z^{2d+2} - 2z^{2d+1} + z^{2d} - 1 = 0.
\]

(3.91)

As the two characteristic equations, (3.88) and (3.91), are the same, their largest real roots are the same, therefore

\[
H(2d+2, 0, \infty, \infty) = H_{dk}(d, \infty). 
\]

(3.92)

Q.E.D.

Relation 5:

\[
H(\alpha+\beta, 0, \infty, \infty) = H(\alpha, \beta, \infty, \infty).
\]

(3.93)

Proof:

Let

\[
\alpha = x+y, \text{ and} \\
\beta = 0. 
\]

(3.94) \hspace{1cm} (3.95)

Substitute (3.94) and (3.95) into (3.57):

\[
z^{x+y} - 2z^{x+y-1} + z^{x+y-2} - 1 = 0.
\]

(3.96)

Let

\[
\alpha = x, \text{ and} \\
\beta = y. 
\]

(3.97) \hspace{1cm} (3.98)
Substitute (3.97) and (3.98) into (3.57):

\[ z^{x+y-2}z^{x+y-1} + z^{x+y-2} - 1 = 0. \]  

(3.99)

The two characteristic equations, (3.96) and (3.99), are the same, thus the largest real roots are the same, therefore

\[ H(\alpha + \beta, 0, \infty, \infty) = H(\alpha, \beta, \infty, \infty). \]  

(3.100)

Q.E.D.

**EXAMPLE:**

From Relations 4 and 5:

\[ H(4, 0, \infty, \infty) = H_{dk}(1, \infty), \text{ and } \]
\[ H(3, 0, \infty, \infty) = H(1, 2, \infty, \infty). \]

Similar relations between the capacity of \((d, k)\) sequences and \((d)\) sequences were derived by Ashley and Siegel [16] and a relation between \((d)\) sequences was derived by Forsberg and Blake [17].

### 3.4 CHANNEL CAPACITY OF CHARGE-CONSTRAINED RLL SEQUENCES

The capacity of a runlength limited sequence with a bounded running digital sum is dependent on five variables \(\alpha, \beta, \gamma, \delta\) and \(C\), and will be denoted by \(H(\alpha, \beta, \gamma, \delta, C)\). The parameters \(\alpha, \beta, \gamma\) and \(\delta\) are the runlength constraints as mentioned in section 3.3 and \(C\) is the maximum absolute value of \(r_\ell\) for any \(\ell\) as given by (3.2). If symmetry exists with respect to \(r_\ell=0\), then

\[ C = \frac{N}{2}, \]  

(3.101)

where \(N\) denotes the digital sum variation (DSV). The general Markov model for
(\(\alpha, \beta, \gamma, \delta, C\)) sequences is shown in Fig. 3.2. The state variables are the entry symbol, runlength of the current symbol and charge level. Here \(\sigma\) denotes the maximum of \(\alpha\) or \(\beta\).

The capacity of these sequences was determined by taking the Markov model and determining the connection matrix \(A\). The capacity is given by:

\[
H(\alpha, \beta, \gamma, \delta, C) = \log_2(\lambda),
\]

where \(\lambda\) is the largest real eigenvalue of \(A\) [11].

Values of channel capacity for different values of \(\alpha, \beta, \gamma, \delta\) and \(C\) are given in Table A.1 in Appendix A. The values of channel capacity in Table A.1 which correspond to \((d, k, C)\) sequences i.e. \(\alpha=\beta=d+1\) and \(\gamma=\delta=k+1\) are consistent with the values given by Norris and Bloomberg [18].

The general Markov model in Fig. 3.2 can be used for non-symmetrical charge levels as well. Examples of such sequences are given in chapter 4. The charge-constrained codes in chapter 4 will be denoted by the 5-tuple \((\alpha, \beta, \gamma, \delta, N)\). \(N\) is used because non-symmetrical charge levels are used.
Figure 3.2 General Markov model for \((a, \beta, \gamma, \delta, C)\) sequences
At first sight, inspection of the table containing values of channel capacity in Appendix A, there seems to be some discrepancies. Consider for example the channel capacity of a (1, 1, 1, 2, C=3) sequence. In the table there is no value of channel capacity given for C=3, but for C=∞ a value of 0.4057 bits/symbol is possible. The reason that the charge constrained sequence does not have any capacity is because the associated Markov model is not a strongly connected graph. In a graph that is not strongly connected, a state cannot always be reached from any other state. The Markov model for a (1, 1, 1, 2, 6), i.e. C=3, sequence is given in Fig. 3.3. It is quite clear from the graph that if the sequence starts in state 17 and progresses through state 1 to state 3, that once in state 3, it is impossible to reach state 17. In fact, after a number of steps the sequence would reach state 12. Once in state 12 the sequence goes into a limit cycle between state 10 and state 12. It is therefore impossible for such a channel to convey any information and therefore the channel capacity is zero. If there is no limit on the running digital sum, the channel becomes a RLL channel with parameters α, β, γ and δ which does have a non-zero capacity.

Some values of channel capacity for (α, β, γ, δ, C) sequences from Table A.1 are presented graphically as a function of the channel parameters in Fig. 3.4 and Fig. 3.5 for (1, β, γ, δ, 3) and (1, β, γ, δ, ∞) sequences respectively, for γ = δ and where β takes on the values 1, 2, 3 and 4. Values of channel capacity are presented graphically in Fig. 3.6 and Fig. 3.7 for (2, β, γ, δ, 3) and (2, β, γ, δ, ∞) sequences respectively, for γ = δ and where β takes on the values 2, 3, 4 and 5. Figures 3.8 and 3.9 contain channel capacities for (3, β, γ, δ, 3) sequences where β ∈ {3, 4, 5} and (3, β, γ, δ, ∞) sequences where β ∈ {3, 4, 5, 6} for γ = δ.

Table 3.1 contains selected values of channel capacity for (α, β, γ, δ, C) sequences extracted from Table A.1.
Figure 3.3 Markov model for a \((1, 1, 1, 2, 6)\) sequence
CHAPTER 3: Generating function, Markov Models...

Figure 3.4 Channel capacity for \((J, \beta, \gamma, \delta, 3)\) sequences

Figure 3.5 Channel capacity for \((J, \beta, \gamma, \delta, \infty)\) sequences

3.31
### Figure 3.6 Channel capacity for (2, β, γ, δ, 3) sequences

### Figure 3.7 Channel capacity for (2, β, γ, δ, ∞) sequences
Figure 3.8 Channel capacity for \((3, \beta, \gamma, \delta, 3)\) sequences

Figure 3.9 Channel capacity for \((3, \beta, \gamma, \delta, \infty)\) sequences


\[ \begin{array}{cccccccc}
\alpha & \beta & \gamma & \delta & C=2 & C=3 & C=4 & C=5 & C=6 & C=\infty \\
1222 & -- & -- & -- & -- & -- & -- & -- & 0.2878 \\
1233 & 0.4280 & 0.4796 & 0.4965 & 0.5041 & 0.5082 & 0.5995 \\
1244 & 0.5000 & 0.5911 & 0.6218 & 0.6358 & 0.6433 & 0.7112 \\
1255 & 0.5000 & 0.6263 & 0.6681 & 0.6875 & 0.6981 & 0.7603 \\
1266 & 0.5000 & 0.6358 & 0.6860 & 0.7096 & 0.7226 & 0.7842 \\
1277 & 0.5000 & 0.6358 & 0.6925 & 0.7192 & 0.7340 & 0.7966 \\
1288 & 0.5000 & 0.6358 & 0.6942 & 0.7231 & 0.7393 & 0.8032 \\
1333 & -- & -- & -- & -- & -- & -- & 0.3218 \\
1344 & 0.2403 & 0.3319 & 0.3557 & 0.3653 & 0.3701 & 0.5234 \\
1355 & 0.2403 & 0.4147 & 0.4587 & 0.4768 & 0.4860 & 0.6050 \\
1366 & 0.2403 & 0.4396 & 0.4996 & 0.5252 & 0.5383 & 0.6464 \\
1377 & 0.2403 & 0.4396 & 0.5160 & 0.5476 & 0.5641 & 0.6656 \\
1388 & 0.2403 & 0.4396 & 0.5215 & 0.5579 & 0.5771 & 0.6774 \\
1444 & -- & -- & -- & -- & -- & -- & 0.3142 \\
1455 & -- & 0.2325 & 0.2674 & 0.2796 & 0.2853 & 0.4650 \\
1466 & -- & 0.2831 & 0.3477 & 0.3714 & 0.3827 & 0.5303 \\
1477 & -- & 0.2831 & 0.3793 & 0.4132 & 0.4295 & 0.5640 \\
1488 & -- & 0.2831 & 0.3909 & 0.4329 & 0.4537 & 0.5829 \\
2333 & -- & -- & -- & -- & -- & -- & 0.1823 \\
2344 & 0.2403 & 0.3205 & 0.3399 & 0.3476 & 0.3514 & 0.4057 \\
2355 & 0.2403 & 0.3985 & 0.4356 & 0.4505 & 0.4579 & 0.4998 \\
2366 & 0.2403 & 0.4208 & 0.4733 & 0.4949 & 0.5057 & 0.5471 \\
2377 & 0.2403 & 0.4208 & 0.4880 & 0.5151 & 0.5290 & 0.5731 \\
2388 & 0.2403 & 0.4208 & 0.4925 & 0.5242 & 0.5406 & 0.5882 \\
2444 & -- & -- & -- & -- & -- & -- & 0.2281 \\
2455 & -- & 0.2325 & 0.2653 & 0.2764 & 0.2814 & 0.3845 \\
2466 & -- & 0.2831 & 0.3435 & 0.3647 & 0.3746 & 0.4550 \\
2477 & -- & 0.2831 & 0.3736 & 0.4041 & 0.4185 & 0.4925 \\
2488 & -- & 0.2831 & 0.3840 & 0.4224 & 0.4408 & 0.5142 \\
2555 & -- & -- & -- & -- & -- & -- & 0.2382 \\
2566 & -- & 0.1492 & 0.2066 & 0.2225 & 0.2290 & 0.3602 \\
2577 & -- & 0.1492 & 0.2652 & 0.2961 & 0.3092 & 0.4173 \\
2588 & -- & 0.1492 & 0.2854 & 0.3291 & 0.3482 & 0.4489 \\
3444 & -- & -- & -- & -- & -- & -- & 0.1335 \\
3455 & -- & 0.2276 & 0.2552 & 0.2642 & 0.2683 & 0.3076 \\
3466 & -- & 0.2757 & 0.3294 & 0.3474 & 0.3555 & 0.3878 \\
3477 & -- & 0.2757 & 0.3580 & 0.3848 & 0.3970 & 0.4312 \\
3488 & -- & 0.2757 & 0.3675 & 0.4020 & 0.4181 & 0.4568 \\
3555 & -- & -- & -- & -- & -- & -- & 0.1769 \\
3566 & -- & 0.1492 & 0.2058 & 0.2204 & 0.2263 & 0.3048 \\
3577 & -- & 0.1492 & 0.2635 & 0.2919 & 0.3035 & 0.3663 \\
3588 & -- & 0.1492 & 0.2829 & 0.3235 & 0.3407 & 0.4011 \\
\end{array} \]

**TABLE 3.1** Selected values of channel capacity for \((\alpha , \beta , \gamma , \delta , C)\) sequences.

3.34
SYNTHESIS OF THE NEW ASYMMETRICAL BINARY MODULATION CODES

Consider an encoder for a modulation code which maps every group of \( m \) data bits onto \( n \) code bits, generating a code sequence which satisfies a certain set of channel input restrictions. From (2.3) the mapping ratio of the encoder is \( R = \frac{m}{n} \). If the channel capacity of the restricted sequence is \( H \), then the sequence utilization efficiency is

\[ \eta = \frac{R}{H} \times 100\% \text{ from (3.6)}. \]

If the encoder generates all possible sequences complying to the given set of input restrictions, then the encoder is said to be maxentropic, and \( R = H \), so that \( \eta = 100\% \). If the communications channel can transfer a sequence which complies to the given set of input restrictions with a given low probability of error, then the channel is capable of transferring \( H \) bits of information per symbol in long independent sequences with the
same low probability of error. Therefore, if $R$ is smaller than $H$, the full capacity of the channel is not utilized. Thus an important property of a modulation code is that $R$ should be as close to $H$ as possible, or that $\eta$ should be as high as possible. The tables in Appendix A containing values of channel capacity were searched for values that are equal to, or slightly higher than 0.5, 0.667 or 0.75 bits/symbol, as these correspond to mapping ratios of 1/2, 2/3 and 3/4, respectively. Attempts were then made to construct codes for generating these sequences from the Markov model of the sequence, henceforth referred to as the graph $G$.

Communications engineers often select a modulation code on grounds of low complexity rather than any other features. The Markov models corresponding to restricted binary asymmetrical sequences are generally quite large, i.e. they have many states. These graphs typically have 12, 14, or even more states. The main reason for this is that non-return-to-zero (NRZ) notation is used when setting up these Markov models, because a non-return-to-zero-inverse (NRZI) notation for binary asymmetrical sequences is not available. NRZI notation is generally used for binary symmetrical $(d, k)$ sequences, as this notation allows a Markov model with only approximately half of the number of states as the corresponding model when NRZ notation is used. To obtain the largest utilization efficiency it was found that the sliding block code algorithm of Adler et al [21] was the most suitable for the synthesis of asymmetrical binary modulation codes. Most of the codes developed in this chapter are therefore constructed with this coding method.

If the encoder maps every $m$ data bits onto $n$ code symbols, and the decoder has a sliding window of only $n$ code symbols, then a single, isolated channel error can cause at most $m$ data bit errors to occur at the decoder output. A code with a mapping ratio of 1/2 would thus be preferrable to a code with a mapping ratio of 2/4, if both codes generate sequences complying to the same input restrictions.

In sections 4.1 to 4.2, different coding methods for finding a code, given the Markov model, graph $G$, are illustrated [19]. The sequence-state coding methods of Franaszek [20] are discussed in section 4.1 and section 4.2 deals with the sliding block code algorithm of Adler et al [21].
4.1 SEQUENCE-STATE CODING METHODS

The synthesis of a binary modulation code for generating sequences that comply with the \((\alpha, \beta, \gamma, \delta)\) constraint, which was constructed with the sequence-state coding methods of Franaszek [20], is presented in detail in this section.

4.1.1 The \((1, 2, 3, 3)\) modulation code

The Markov model for a binary sequence that complies with the \((\alpha, \beta, \gamma, \delta) = (1, 2, 3, 3)\) constraints is shown in Fig. 4.1. We call this graph \(G\). The state variables are the entry symbol and the current runlength of the symbol, and the states are numbered from 1 to 6.

![Markov model for the \((1, 2, 3, 3)\) sequence](image)

For this sequence \(\zeta=1/2\) and the channel capacity corresponding to this Markov model, is 0.5995 bits/symbol. This means that if an encoder is constructed which generates these sequences with a mapping ratio of \(R = \frac{1}{2}\), then the encoder will have a sequence utilization efficiency of \(\eta = \frac{0.5}{0.5995} = 83.4\%\). The connection matrix \(A\) for \(G\) is:
The matrix $A^n$ can also be set up, such that $a_{ij}$ in $A^n$ denotes the number of edges in the graph $G^n$ from state $i$ to state $j$. Alternatively, $a_{ij}$ in $A^n$ denotes the number of unique paths of length $n$ in $G$ from state $i$ to state $j$. Since a mapping ratio of $R = 1/2$ is desired, the encoder should be constructed from the graph $G^2$. The connection matrix $A^2$ which corresponds to the graph $G^2$ is:

$$A^2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.2)$$

If every data bit is to be mapped onto two code bits ($R = 1/2$) with the graph $G^2$ as encoder, then each state in $G^2$ should have at least $2^n=2$ outgoing edges. However, row 3 in $A^2$ has a row sum of only 1, which means that state 3 in $G^2$ which corresponds to this row, has only one outgoing edge.

The labels of the edges in the graph $G^2$, i.e. the groups of two code bits, can either be obtained by inspection of the graph $G$ or by means of the transition matrix, $B$ [22]. In this matrix, $b_{ij}$ is all the labels of the edges from state $i$ to state $j$ in $G$. The symbol $\phi$ is used when no edges exist from state $i$ to state $j$. The matrix $B^n$ thus contains all the groups of $n$ symbols which are the labels of the edges in the graph $G^n$. The transition matrix $B$ which corresponds to the graph $G$ of Fig. 4.1 is:

$$B = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.4)$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.1)$$
It was found that a set of principal states \[22\], which is a subset of the states of \( G^2 \) exists, \textit{i.e.} a set of states such that each state has at least \( 2^m \) outgoing edges terminating in one of the states in the set. The set of principal states which exists in \( G^2 \) consists of the states 1, 2, 4, 5 and 6. The connection matrix for the graph \( G^2 \) with state 3 deleted, \textit{i.e.} the graph \( G^{2'} \), can be obtained by deleting the corresponding rows and columns of \( A^2 \). The resulting matrix is:

\[
A^{2'} = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0
\end{bmatrix}.
\]

(4.4)

The transition matrix corresponding to \( G^{2'} \), namely \( B^{2'} \) is:

\[
B^{2'} = \begin{bmatrix}
\phi & \phi & 01 & 11 & \phi \\
\phi & \phi & 01 & 11 & \phi \\
10 & \phi & \phi & 11 & \phi \\
10 & 00 & 01 & \phi & \phi \\
\phi & 00 & 01 & \phi & \phi
\end{bmatrix}.
\]

(4.5)

The graph \( G^{2'} \), which corresponds to the connection matrix \( A^{2'} \) and the transition matrix \( B^{2'} \), is shown in Fig. 4.2. If two or more states, henceforth called 'old states', have identical outgoing edges terminating in the same state, then these old states can be merged to a single, new state \[23\]. The incoming edges of all the old states are all redirected to terminate in the new state, while the new state has the set of outgoing edges that was common to the old states.

4.5
From inspection of (4.4) it is clear that row 4 in $B^2$ has three outgoing edges, but only two are needed for the mapping. It is clear from (4.5) that if the entry $b_{41}$ is eliminated from $B^2$, then states 4 and 5, as well as states 1 and 2 can be merged to a single state.

The graph $G^2$ can thus be reduced to a graph consisting of three states. This graph can then be used to construct an encoder with a mapping ratio of $R = 1/2$. Construction of such a modulation code consists of attempting to assign every possible unique group of $m$ information bits to one or more different groups of $n$ code bits, such that no group of code bits has more than one group of information (data) bits assigned to it, and every state has every possible group of information bits assigned to exactly one of its outgoing edges. The process is made easier if the same group of information bits is
assigned to the same group of code bits in cases where such a group corresponds to one of the outgoing edges of more than one state in the graph. If this is successful, a decoder consisting of a serial buffer (sliding block) of length $n$ bits and a simple lookup table can be constructed, limiting the number of data errors due to a single, isolated channel error to at most $m$ bits. Such an assignment of groups of data bits to groups of code bits is not necessarily unique.

The code that resulted from the abovementioned procedure was named $(1, 2, 3, 3)$ and the state machine (Mealy machine) representation of the encoder is shown in graph form in Fig. 4.3.

![Figure 4.3 Encoder for the $(1, 2, 3, 3)$ code](image)

The encoders can also be presented in a finite state machine table form. The encoder table for $(1, 2, 3, 3)$ is shown in Table 4.1. The state-independent sliding block decoder for the $(1, 2, 3, 3)$ code is shown in Table 4.2. In this table all the possible buffer contents are shown, together with the data that it represents. The buffer contents is presented in such a way that the first code bit to be received is written first i.e. code bits are shifted in from the right.
4.2 SLIDING BLOCK CODE ALGORITHM

In this section, the sliding block code algorithm of Adler et al [21] is used to find some new binary modulation codes. Tutorial expositions of the algorithm were given by Siegel [24] and Immink [1]. The synthesis of a number of modulation codes for generating sequences with various mapping ratios and input restrictions, which were constructed with the sliding block code algorithm, also known as the ACH algorithm, is presented in detail in this section.

4.2.1 The (1, 2, 7, 7) modulation code

The Markov model (graph G) for a binary sequence that complies with the \((a, \beta, \gamma, \delta) = (1, 2, 7, 7)\) constraint is shown in Fig. 4.4.
For this sequence $\xi=1/2$ and the channel capacity was calculated as 0.7966 bits/symbol. An encoder which generates this sequence with a mapping ratio of $R = \frac{3}{4}$ would thus have a sequence utilization efficiency of $\eta = 94.15\%$. The connection matrix $A^4$ corresponding to the graph $G^4$ is:

$$A^4 = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.
It is clear from evaluating the row sums in (4.7) that states 5, 6, 7, 8 and 14 in $G^4$ have 7, 6, 4, 7 and 5 outgoing edges, respectively. However, $2^3 = 8$ outgoing edges are needed at each state to construct the desired encoder. According to the sliding block code algorithm, an 'approximate eigenvector', $v$, should be found, such that

$$A^n v \geq 2^m v.$$  \hspace{1cm} (4.8)

The quantity $2^m$ is called the 'approximate eigenvalue'. The vector $v$ is a column matrix of the same dimension as $A$. The $i$-th component of $v$ is called the 'weight' of state $i$ in the graph $G^n$.

For this example, an approximate eigenvector was found through a computer search to be:

$$v = [3 \ 3 \ 2 \ 1 \ 2 \ 1 \ 2 \ 3 \ 3 \ 1 \ 1]^T.$$  \hspace{1cm} (4.9)

The computerised search methods are discussed in more detail in Appendix G.
Equation (4.9) indicates that the states 1, 2, 9, 10 and 11, in $G^4$, should be split into three 'offspring states' and states 3, 5, 6 and 8, in $G^4$, should be split into two 'offspring states', while the remaining states should not be split. The resulting graph $G^4$ will thus consist of 28 states. The transition matrix $B^4$ corresponding to $G^4$ is given in (4.10) and was determined with a computer program which is given in Appendix F.

The splitting rule requires that all the edges in $G^4$ which terminate in states 1, 2, 9, 10 and 11 should be redirected to all three offspring states of each state and all the edges in $G^4$ which terminate in states 3, 5, 6 and 8 should be redirected to both offspring states of each state. The outgoing edges of states 1, 2, 9, 10 and 11 should be partitioned into three groups between the offspring states, such that the sum of the weights of the terminal states of edges in a group is an integer multiple of the approximate eigenvalue $2^3$, with the possible exception of one group. The outgoing edges of states 3, 5, 6 and 8 should be partitioned into two groups between the offspring states, such that the sum of the weights of the terminal states of edges in a group is an integer multiple of the approximate eigenvalue $2^3$, with the possible exception of one group. For example, all the sequences in the first column of $B^4$ terminate in state 1, which has a weight of 3. The three states that state 1 are split into, are named $1^1$, $1^2$ and $1^3$ and the other states are numbered in a similar manner. The sequences in the first row of $B^4$ denote the outgoing edges of state 1, and should thus be partitioned into three groups according to the splitting rule. From inspection of (4.10) we note that states 12, 13 and 14 have a weight of one, thus eight outgoing edges are needed from each state. Fig. 4.5 shows state 1 before the splitting procedure was applied with outgoing edges to states 1, 8 and 9. If we select the edges going to states 1, 8 and 9 with the edge labels 0110, 0001 and 0011, respectively, then we obtain $3+2+3=8$ outgoing edges according to the weight of states 1, 8 and 9. These edges are assigned to state $1^1$ and is illustrated in Fig. 4.5. The reason for choosing these specific edges is because many of the other states have the exact same edges to the exact same states as is evident from (4.10).
\[ B^4 = \]

\[
\begin{array}{cccccccccccccccc}
0110 & 1100 & \phi & \phi & 0000 & \phi & \phi & 0001 & 0011 & 0111 & 1111 & \phi & \phi & \phi \\
1110 & \\
0110 & 1100 & \phi & \phi & \phi & 0000 & \phi & 0001 & 0011 & 0111 & 1111 & \phi & \phi & \\
1110 & \\
0110 & 1100 & \phi & \phi & \phi & \phi & 0000 & 0011 & 0111 & 1111 & \phi & \phi & \\
1110 & \\
0110 & 1100 & \phi & \phi & \phi & \phi & \phi & 0011 & 0111 & 1111 & \phi & \phi & \\
1110 & \\
1110 & 1100 & \phi & \phi & \phi & \phi & \phi & 1101 & 0011 & 0111 & \phi & \phi & \\
1110 & \\
1110 & 1100 & 1000 & \phi & \phi & \phi & \phi & 1001 & 1011 & \phi & \phi & 1111 & \phi & \\
1110 & \\
0110 & 1100 & 1000 & 0000 & \phi & \phi & \phi & 0001 & 0011 & 0111 & \phi & 1111 & \phi & \\
1110 & \\
0110 & 1100 & 1000 & 0000 & \phi & \phi & \phi & 0001 & 0011 & 0111 & \phi & \phi & 1111 & \\
1110 & \\
0110 & 1100 & 1000 & 0000 & \phi & \phi & \phi & 0001 & 0011 & 0111 & \phi & \phi & \phi & \\
1110 & \\
0110 & 1100 & 1000 & 0000 & \phi & \phi & \phi & 0001 & 0011 & 0111 & \phi & \phi & \phi & \\
1110 & \\
0110 & \phi & 1000 & 0000 & \phi & \phi & \phi & 0001 & 0011 & 0111 & \phi & \phi & \phi & \\
1110 & \\
0110 & \phi & \phi & 0000 & \phi & \phi & \phi & 0001 & 0011 & 0111 & \phi & \phi & \phi & \\
1110 & \\
\end{array}
\]

(4.10)
CHAPTER 4: Synthesis of the new asymmetrical...

For example the states $2^1, 3^1, 4^1, 9^1, 10^1, 11^1, 12^1, 13^1$ and $14^1$ can be chosen to have exactly the same edges as state $1^1$ and can thus be reduced to a single state. The abovementioned procedure was done for all the states to ensure an encoder with the minimum number of states. The name of the new state, the states which have been merged to a single state and the outgoing edges associated with these states are given in Table 4.3.
CHAPTER 4: Synthesis of the new asymmetrical...

<table>
<thead>
<tr>
<th>NEW STATE</th>
<th>STATES MERGED</th>
<th>OUTGOING EDGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$1^1, 2^1, 3^1, 4^1, 9^1, 10^1$, $11^1, 12^1, 13^1, 14^1$</td>
<td>0001, 0011, 0110</td>
</tr>
<tr>
<td>B</td>
<td>$1^2, 2^2, 3^2, 4^2, 5^2, 6^2$, $7^2, 8^2, 9^2, 10^2, 11^2$</td>
<td>1100, 1101, 1110</td>
</tr>
<tr>
<td>C</td>
<td>$1^3, 2^3$</td>
<td>0000, 0111, 1111</td>
</tr>
<tr>
<td>D</td>
<td>$5^1, 6^1$</td>
<td>0110, 0111, 1111</td>
</tr>
<tr>
<td>E</td>
<td>$8^1, 9^3, 10^3$</td>
<td>1000, 1001, 1011, 1111</td>
</tr>
<tr>
<td>F</td>
<td>$11^3$</td>
<td>0000, 0111, 1001, 1011</td>
</tr>
</tbody>
</table>

TABLE 4.3 Summary of states merged with their associated edges

By careful application of the abovementioned procedure it was possible to reduce the initial 28 states to 6 states. An encoder and state-independent decoder was found, and the code was named (1, 2, 7, 7). The FSM encoder representation is shown in Fig. 4.6 and the encoder table is shown in Table 4.4.
Figure 4.6 Encoder for the (1, 2, 7, 7) code
A state-independent sliding-block decoder which uses look-ahead decoding of four bits and look-back decoding of one bit was found, and is presented in Table 4.5. The right bit in the table is the most recent bit shifted into the decoder buffer, which is nine bits long. The decoder table is also given in hexadecimal form in Table 4.6.

### Table 4.4 Encoder table for the (1, 2, 7, 7) code

<table>
<thead>
<tr>
<th>DATA</th>
<th>CODE</th>
<th>Next state</th>
<th>CODE</th>
<th>Next state</th>
<th>CODE</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0110</td>
<td>A</td>
<td>1101</td>
<td>B</td>
<td>0000</td>
<td>B</td>
</tr>
<tr>
<td>001</td>
<td>0011</td>
<td>A</td>
<td>1110</td>
<td>B</td>
<td>0111</td>
<td>A</td>
</tr>
<tr>
<td>010</td>
<td>0110</td>
<td>B</td>
<td>1100</td>
<td>B</td>
<td>0000</td>
<td>D</td>
</tr>
<tr>
<td>011</td>
<td>0110</td>
<td>C</td>
<td>1110</td>
<td>C</td>
<td>1111</td>
<td>A</td>
</tr>
<tr>
<td>100</td>
<td>0001</td>
<td>B</td>
<td>1100</td>
<td>C</td>
<td>0111</td>
<td>B</td>
</tr>
<tr>
<td>101</td>
<td>0011</td>
<td>B</td>
<td>1101</td>
<td>E</td>
<td>1111</td>
<td>B</td>
</tr>
<tr>
<td>110</td>
<td>0011</td>
<td>E</td>
<td>1100</td>
<td>A</td>
<td>0111</td>
<td>E</td>
</tr>
<tr>
<td>111</td>
<td>0001</td>
<td>E</td>
<td>1110</td>
<td>A</td>
<td>1111</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA</th>
<th>CODE</th>
<th>Next state</th>
<th>CODE</th>
<th>Next state</th>
<th>CODE</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0110</td>
<td>A</td>
<td>1001</td>
<td>E</td>
<td>1001</td>
<td>E</td>
</tr>
<tr>
<td>001</td>
<td>0111</td>
<td>A</td>
<td>1011</td>
<td>E</td>
<td>1011</td>
<td>E</td>
</tr>
<tr>
<td>010</td>
<td>0110</td>
<td>B</td>
<td>1000</td>
<td>B</td>
<td>1000</td>
<td>B</td>
</tr>
<tr>
<td>011</td>
<td>1111</td>
<td>A</td>
<td>1111</td>
<td>A</td>
<td>0000</td>
<td>A</td>
</tr>
<tr>
<td>100</td>
<td>0111</td>
<td>B</td>
<td>1001</td>
<td>B</td>
<td>1001</td>
<td>B</td>
</tr>
<tr>
<td>101</td>
<td>1111</td>
<td>B</td>
<td>1011</td>
<td>B</td>
<td>1011</td>
<td>B</td>
</tr>
<tr>
<td>110</td>
<td>0111</td>
<td>E</td>
<td>1000</td>
<td>A</td>
<td>0111</td>
<td>E</td>
</tr>
<tr>
<td>111</td>
<td>1111</td>
<td>F</td>
<td>1011</td>
<td>A</td>
<td>1011</td>
<td>A</td>
</tr>
</tbody>
</table>

4.16
### TABLE 4.5 Decoder table for the (1, 2, 7, 7) code

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 0110 0001</td>
<td>000</td>
</tr>
<tr>
<td>X 0110 0011</td>
<td></td>
</tr>
<tr>
<td>X 0110 0110</td>
<td></td>
</tr>
<tr>
<td>X 0000 1100</td>
<td></td>
</tr>
<tr>
<td>X 0000 1110</td>
<td></td>
</tr>
<tr>
<td>X 1101 1100</td>
<td></td>
</tr>
<tr>
<td>X 1101 1101</td>
<td></td>
</tr>
<tr>
<td>X 1101 1110</td>
<td></td>
</tr>
<tr>
<td>X 1001 1000</td>
<td></td>
</tr>
<tr>
<td>X 1001 1001</td>
<td></td>
</tr>
<tr>
<td>X 1001 1011</td>
<td></td>
</tr>
<tr>
<td>X 1001 1111</td>
<td></td>
</tr>
<tr>
<td>X 0110 0000</td>
<td>011</td>
</tr>
<tr>
<td>X 0110 0111</td>
<td></td>
</tr>
<tr>
<td>X 0110 1111</td>
<td></td>
</tr>
<tr>
<td>X 1110 0000</td>
<td></td>
</tr>
<tr>
<td>X 1110 0111</td>
<td></td>
</tr>
<tr>
<td>X 1110 1111</td>
<td></td>
</tr>
<tr>
<td>X 0000 0001</td>
<td></td>
</tr>
<tr>
<td>X 0000 0011</td>
<td></td>
</tr>
<tr>
<td>X 1000 0110</td>
<td></td>
</tr>
<tr>
<td>X 1111 0001</td>
<td></td>
</tr>
<tr>
<td>X 1111 0011</td>
<td></td>
</tr>
<tr>
<td>X 1111 0111</td>
<td></td>
</tr>
<tr>
<td>X 1111 1011</td>
<td></td>
</tr>
<tr>
<td>X 1111 1111</td>
<td></td>
</tr>
<tr>
<td>X 0011 1100</td>
<td>101</td>
</tr>
<tr>
<td>X 0011 1101</td>
<td></td>
</tr>
<tr>
<td>X 0011 1110</td>
<td></td>
</tr>
<tr>
<td>X 1011 1100</td>
<td></td>
</tr>
<tr>
<td>X 1011 1101</td>
<td></td>
</tr>
<tr>
<td>X 1011 1111</td>
<td></td>
</tr>
<tr>
<td>X 1111 1100</td>
<td></td>
</tr>
<tr>
<td>X 1111 1110</td>
<td></td>
</tr>
<tr>
<td>X 1011 1000</td>
<td></td>
</tr>
<tr>
<td>X 1011 1001</td>
<td></td>
</tr>
<tr>
<td>X 1011 1011</td>
<td></td>
</tr>
<tr>
<td>X 1011 1110</td>
<td></td>
</tr>
<tr>
<td>X 1100 0001</td>
<td></td>
</tr>
<tr>
<td>X 1100 0011</td>
<td></td>
</tr>
<tr>
<td>X 1100 0110</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 0001 1100</td>
<td>010</td>
</tr>
<tr>
<td>X 0001 1101</td>
<td></td>
</tr>
<tr>
<td>X 0001 1111</td>
<td></td>
</tr>
<tr>
<td>X 0110 0001</td>
<td></td>
</tr>
<tr>
<td>X 0110 0110</td>
<td></td>
</tr>
<tr>
<td>X 0110 1110</td>
<td></td>
</tr>
<tr>
<td>X 1110 0001</td>
<td></td>
</tr>
<tr>
<td>X 1110 0110</td>
<td></td>
</tr>
<tr>
<td>X 1110 1110</td>
<td></td>
</tr>
<tr>
<td>X 1001 1000</td>
<td></td>
</tr>
<tr>
<td>X 1001 1001</td>
<td></td>
</tr>
<tr>
<td>X 1001 1011</td>
<td></td>
</tr>
<tr>
<td>X 1001 1111</td>
<td></td>
</tr>
<tr>
<td>X 0100 0011</td>
<td></td>
</tr>
<tr>
<td>X 0100 0110</td>
<td></td>
</tr>
<tr>
<td>X 1100 0001</td>
<td></td>
</tr>
<tr>
<td>X 1100 0011</td>
<td></td>
</tr>
<tr>
<td>X 1100 0110</td>
<td></td>
</tr>
<tr>
<td>X 1111 1011</td>
<td></td>
</tr>
<tr>
<td>X 1111 1000</td>
<td></td>
</tr>
<tr>
<td>X 1111 1001</td>
<td></td>
</tr>
<tr>
<td>X 1111 1011</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 0001 1100</td>
<td>111</td>
</tr>
<tr>
<td>X 0001 1101</td>
<td></td>
</tr>
<tr>
<td>X 0001 1111</td>
<td></td>
</tr>
<tr>
<td>X 0110 0001</td>
<td></td>
</tr>
<tr>
<td>X 0110 0110</td>
<td></td>
</tr>
<tr>
<td>X 0110 1110</td>
<td></td>
</tr>
<tr>
<td>X 1110 0001</td>
<td></td>
</tr>
<tr>
<td>X 1110 0110</td>
<td></td>
</tr>
<tr>
<td>X 1110 1110</td>
<td></td>
</tr>
<tr>
<td>X 1001 1000</td>
<td></td>
</tr>
<tr>
<td>X 1001 1001</td>
<td></td>
</tr>
<tr>
<td>X 1001 1011</td>
<td></td>
</tr>
<tr>
<td>X 1001 1111</td>
<td></td>
</tr>
<tr>
<td>X 0100 0011</td>
<td></td>
</tr>
<tr>
<td>X 0100 0110</td>
<td></td>
</tr>
<tr>
<td>X 1100 0001</td>
<td></td>
</tr>
<tr>
<td>X 1100 0011</td>
<td></td>
</tr>
<tr>
<td>X 1100 0110</td>
<td></td>
</tr>
<tr>
<td>X 1111 1011</td>
<td></td>
</tr>
<tr>
<td>X 1111 1000</td>
<td></td>
</tr>
<tr>
<td>X 1111 1001</td>
<td></td>
</tr>
<tr>
<td>X 1111 1011</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 4: Synthesis of the new asymmetrical...

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>61 63 66 0C 0D 0E DC DD DE 98 99 9B 9F</td>
<td>000</td>
</tr>
<tr>
<td>60 67 6F E0 E7 EF 01 03 F1 F3 F6 06+</td>
<td>011</td>
</tr>
<tr>
<td>3C 3D 3E BC BD BE D8 D9 DB DF FC FD FE</td>
<td>101</td>
</tr>
<tr>
<td>38 39 3B 3F 78 79 7B 7F 81 83 86 C1 C3 C6</td>
<td>110</td>
</tr>
</tbody>
</table>

\[+\text{ Preceded by a binary 1}\]

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 33 36 71 73 76 EC ED EE B8 B9 BB BF</td>
<td>001</td>
</tr>
<tr>
<td>07 0F 6C 6D 6E 8C 8D 8E CC CD CE 06\times</td>
<td>010</td>
</tr>
<tr>
<td>1C 1D 1E 7C 7D 7E 9C 9D 9E C0 C7 CF</td>
<td>100</td>
</tr>
<tr>
<td>18 19 1B 1F B1 B3 B6 E1 E3 E6 F0 F7 F8 P9 FB</td>
<td>111</td>
</tr>
</tbody>
</table>

\[\times\text{ Preceded by a binary 0}\]

TABLE 4.6 Hex decoder table for the (1, 2, 7, 7) code

4.2.2 The (2, 3, 6, 6) modulation code

The Markov model (graph G) for a binary sequence that complies with the $(\alpha, \beta, \gamma, \delta) = (2, 3, 6, 6)$ constraint is shown in Fig. 4.7.

![Image of Markov model for the (2, 3, 6, 6) sequence](image)

**Figure 4.7 Markov model for the (2, 3, 6, 6) sequence**
The connection matrix corresponding to the graph G is:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$ (4.11)

For this sequence $\xi=2/3$ and the channel capacity was calculated as 0.5471 bits/symbol. An encoder which generates this sequence with a mapping ratio of $R = \frac{1}{2}$ would thus have a sequence utilization efficiency of $\eta = 91.4\%$. The connection matrix $A^2$ corresponding to the graph $G^2$ is:

$$A^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$ (4.12)

It is clear from evaluating the row sums in (4.12) that states 6, 7 and 12 in $G^4$ each have only one outgoing edge. However, $2^1 = 2$ outgoing edges are needed at each state to construct the desired encoder. According to the sliding block code algorithm, an approximate eigenvector, $v$, was found to be:

4.19
\[ v = [2 \ 4 \ 2 \ 3 \ 2 \ 1 \ 2 \ 3 \ 4 \ 4 \ 2 \ 2]^T. \] (4.13)

Equation (4.13) indicates that the states 2, 9 and 10, in \( G^2 \), should each be split into four offspring states, states 4 and 8 should each be split into three offspring states and states 1, 3, 5, 7, 11 and 12 should each be split into two offspring states, while the remaining state should not be split. The resulting graph \( G^2 \) will thus consist of 31 states. The transition matrix \( B^2 \) corresponding to \( G^2 \) is:

\[
B^2 = \begin{bmatrix}
\phi & \phi & 00 & \phi & \phi & 01 & \phi & \phi & \phi & \phi \\
\phi & \phi & \phi & 00 & \phi & 01 & 11 & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & 00 & 01 & 11 & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi & 01 & 11 & \phi & \phi & \phi \\
10 & 00 & \phi & \phi & \phi & \phi & \phi & \phi & \phi & 11 \\
10 & 00 & \phi & \phi & \phi & \phi & \phi & \phi & \phi & 11 \\
10 & 00 & \phi & \phi & \phi & \phi & \phi & \phi & \phi & 11 \\
10 & 00 & \phi & \phi & \phi & \phi & \phi & \phi & \phi & 11 \\
\phi & 00 & \phi & \phi & \phi & \phi & \phi & \phi & \phi & \phi \\
\phi & 00 & \phi & \phi & \phi & \phi & \phi & \phi & \phi & \phi
\end{bmatrix}. \] (4.14)

The splitting rule requires that all the edges in \( G^2 \) which terminate in states 2, 9 and 10 should be redirected to all four offspring states of each state, that all the edges in \( G^2 \) which terminate in states 4 and 8 should be redirected to all three offspring states of each state and all the edges in \( G^2 \) which terminate in states 1, 3, 5, 7, 11 and 12 should be redirected to both offspring states of each state. The outgoing edges of states 2, 9 and 10 should be partitioned into four groups between the offspring states, such that the sum of the weights of the terminal states of edges in a group is an integer multiple of the approximate eigenvalue \( 2^1 \), with the possible exception of one group. The outgoing edges of states 4 and 8 should be partitioned into three groups between the offspring states, such that the sum of the weights of the terminal states of edges in a group is an integer multiple of the approximate eigenvalue \( 2^1 \), with the possible exception of one.
group. The outgoing edges of states 1, 3, 5, 7, 11 and 12 should be partitioned into two groups between the offspring states, such that the sum of the weights of the terminal states of edges in a group is an integer multiple of the approximate eigenvalue $2^1$, with the possible exception of one group. For example, all the sequences in the first column of $B^2$ terminate in state 1, which has a weight of 2 meaning that state one is split into two states namely $1^1$ and $1^2$. The other states which were split are numbered in a similar manner. The sequences in the first row of $B^2$ denote the outgoing edges of state 1, and should thus be partitioned into two groups according to the splitting rule. The same procedure as discussed for the $(1, 2, 7, 7)$ code was applied to reduce the encoder to a minimum number of states. The name of the new state, the states which have been merged to a single state and the outgoing edges associated with these states are given in Table 4.7.

<table>
<thead>
<tr>
<th>NEW STATE</th>
<th>STATES MERGED</th>
<th>OUTGOING EDGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$8^2$</td>
<td>11</td>
</tr>
<tr>
<td>B</td>
<td>$9^3, 10^3, 11^3, 12^2$</td>
<td>00</td>
</tr>
<tr>
<td>C</td>
<td>$10^4$</td>
<td>11</td>
</tr>
<tr>
<td>D</td>
<td>$3^1$</td>
<td>00</td>
</tr>
<tr>
<td>E</td>
<td>$2^3, 4^3, 5^2, 6^1$</td>
<td>11</td>
</tr>
<tr>
<td>F</td>
<td>$7^2$</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>$1^1$</td>
<td>00</td>
</tr>
<tr>
<td>H</td>
<td>$2^2, 3^2, 4^2, 5^1$</td>
<td>01</td>
</tr>
<tr>
<td>I</td>
<td>$7^1, 8^1, 9^4$</td>
<td>11</td>
</tr>
<tr>
<td>J</td>
<td>$2^4, 4^1$</td>
<td>00, 11</td>
</tr>
<tr>
<td>K</td>
<td>$9^2, 10^2, 11^2, 12^1$</td>
<td>00</td>
</tr>
<tr>
<td>L</td>
<td>$8^3, 9^1, 10^1, 11^1$</td>
<td>10</td>
</tr>
<tr>
<td>M</td>
<td>$2^1$</td>
<td>00</td>
</tr>
</tbody>
</table>

**Table 4.7** Summary of states merged with their associated edges

4.21
By careful application of the abovementioned procedure it was possible to reduce the initial 31 states to 13 states. An encoder and state-independent decoder was found. The FSM encoder of the code, that was named (2, 3, 6, 6), is shown in Fig. 4.8 and the encoder table is shown in Table 4.8. A state-independent sliding-block decoder which uses look-back decoding of 2 bits and look-ahead decoding of 10 bits was found, and is presented in Table 4.9. In this table 'X' denotes either a zero or a one. The right bit in the table is the most recent bit shifted into the decoder buffer.

![Diagram of the encoder for the (2, 3, 6, 6) code]

*Figure 4.8. Encoder for the (2, 3, 6, 6) code*
**TABLE 4.8 Encoder table for the (2, 3, 6, 6) code**

<table>
<thead>
<tr>
<th>STATE</th>
<th>CODE</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>11 11</td>
<td>0 1</td>
</tr>
<tr>
<td>B</td>
<td>00 00</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>11 11</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>00 00</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>11 11</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>11 11</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>00 00</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>01 01</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>11 11</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>00 11</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>00 00</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>10 10</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>00 00</td>
<td></td>
</tr>
</tbody>
</table>

**Next state**

- A: B C
- B: J E
- C: B K
- D: E H
- E: I A
- F: I B
- G: H D
- H: I F
- I: L K
- J: E L
- K: M H
- L: G H
- M: J H
### TABLE 4.9 Decoder table for the (2, 3, 6, 6) code

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 00 00 11 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 11 10 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 11 11 01 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 11 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 11 11 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 10 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 10 01 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 11 00 00 11</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 11 00 01 11</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 11 00 11 10</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 11 10 00 XX</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 11 10 01 XX</td>
<td>0</td>
</tr>
<tr>
<td>00 01 11 11 11 00 XX</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 00 01 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 00 00 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 00 11 00 11</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 00 11 11 00</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 10 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 10 01 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 11 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 11 11 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 11 00 00 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 11 00 01 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 11 01 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 00 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 00 01 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 01 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 11 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 00 01 11 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 00 01 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 10 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 10 01 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 11 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 00 00 11 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 00 11 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 11 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>00 00 11 11 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 00 11 10 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 00 00 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 00 01 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 11 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>01 11 10 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 00 01 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 10 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 10 01 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 11 00 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>10 01 11 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>11 00 00 00 11 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>11 00 00 11 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>11 00 11 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>11 00 11 11 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>11 10 00 01 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>11 11 00 00 11 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>11 11 00 01 11 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>11 11 11 00 00 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>11 11 11 00 01 11 XX</td>
<td>0</td>
</tr>
<tr>
<td>11 11 11 11 10 XX XX</td>
<td>0</td>
</tr>
<tr>
<td>11 11 11 11 11 XX XX</td>
<td>0</td>
</tr>
</tbody>
</table>
4.2.3 The (2, 4, 4, 4) modulation code

The Markov model (graph G) for a binary sequence that complies with the \((a, \beta, \gamma, \delta) = (2, 4, 4, 4)\) constraint is shown in Fig. 4.9.

The connection matrix corresponding to the graph G is:

\[
A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]  \hspace{1cm} (4.15)

For this sequence \(\zeta=2/4\) and the channel capacity was calculated as 0.2281 bits/symbol. An encoder which generates this sequence with a mapping ratio of \(R = \frac{1}{5}\) would thus have a sequence utilization efficiency of \(\eta = 87.7\%\). The connection matrix \(A^5\) corresponding to the graph \(G^5\) is:
It is clear from evaluating the row sums in (4.16) that states 4 and 5 in $G^5$ each have only one outgoing edge. However, $2^1 = 2$ outgoing edges are needed at each state to construct the desired encoder. According to the sliding block code algorithm, an approximate eigenvector, $v$, was found to be:

$$v = [3 \ 4 \ 2 \ 1 \ 2 \ 2 \ 2 \ 3]^T.$$  \hspace{1cm} (4.17)

Equation (4.17) indicates that the state 2, in $G^5$, should be split into four offspring states, states 1 and 8 should be split into three offspring states and states 3, 5, 6 and 7 should be split into two offspring states, while the remaining state 4 should not be split. The resulting graph $G^5$ will thus consist of 19 states. The transition matrix $B^5$ corresponding to $G^5$ is:

$$B^5 = \begin{bmatrix} \phi & \phi & \phi & \phi & \phi & 0011 & 0011 & 0111 \ 1110 & \phi & \phi & \phi & \phi & 0011 & 0111 \ 1110 & \phi & \phi & \phi & \phi & \phi & 0111 \ 1110 & \phi & \phi & \phi & \phi & \phi & \phi & \phi \ \phi & 1100 & \phi & \phi & 1100 & \phi & \phi & \phi \ \phi & \phi & 1000 & \phi & 1000 & \phi & \phi & \phi \ \phi & \phi & \phi & \phi & 0001 & \phi & \phi & \phi \ \phi & \phi & \phi & \phi & \phi & 0011 & 0111 & \phi \end{bmatrix}. \hspace{1cm} (4.18)$$

4.26
The splitting rule requires that all the edges in $G^5$ which terminate in state 2 should be redirected to all four offspring states of each state, that all the edges in $G^5$ which terminate in states 1 and 8 should be redirected to all three offspring states of each state and all the edges in $G^5$ which terminate in states 3, 5, 6 and 7 should be redirected to both offspring states of each state. The outgoing edges of state 2 should be partitioned into four groups between the offspring states, such that the sum of the weights of the terminal states of edges in a group is an integer multiple of the approximate eigenvalue $2^1$, with the possible exception of one group. The outgoing edges of states 1 and 8 should be partitioned into three groups between the offspring states and the outgoing edges of states 3, 5, 6 and 7 should be partitioned into two groups between the offspring states, such that the sum of the weights of the terminal states of edges in a group is an integer multiple of the approximate eigenvalue $2^1$, with the possible exception of one group.

The same procedure as discussed for the (1, 2, 7, 7) code was applied to reduce the encoder to a minimum number of states. The name of the new state, the states which have been merged to a single state and the outgoing edges associated with these states are given in Table 4.10.
TABLE 4.10 Summary of states merged with their associated edges

<table>
<thead>
<tr>
<th>NEW STATE</th>
<th>STATES MERGED</th>
<th>OUTGOING EDGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6^2</td>
<td>11001</td>
</tr>
<tr>
<td>B</td>
<td>5^2</td>
<td>11100</td>
</tr>
<tr>
<td>C</td>
<td>8^1</td>
<td>00001</td>
</tr>
<tr>
<td>D</td>
<td>7^2</td>
<td>10011</td>
</tr>
<tr>
<td>E</td>
<td>5^1</td>
<td>11100</td>
</tr>
<tr>
<td>F</td>
<td>7^1</td>
<td>10001</td>
</tr>
<tr>
<td>G</td>
<td>6^1</td>
<td>11000</td>
</tr>
<tr>
<td>H</td>
<td>2^1, 3^1, 4^1</td>
<td>11110</td>
</tr>
<tr>
<td>I</td>
<td>1^2, 2^2, 8^3</td>
<td>00111</td>
</tr>
<tr>
<td>J</td>
<td>1^1, 8^2</td>
<td>00011</td>
</tr>
<tr>
<td>K</td>
<td>1^3, 2^3, 3^2</td>
<td>01111</td>
</tr>
<tr>
<td>L</td>
<td>2^4</td>
<td>01111, 01111</td>
</tr>
</tbody>
</table>

By careful application of the abovementioned procedure it was possible to reduce the initial 19 states to 12 states. An encoder and state-independent decoder was found, and the code was named (2, 4, 4, 4). The FSM encoder representation is shown Fig. 4.10 and encoder table is shown in Table 4.11.

A state-independent sliding-block decoder which uses look-ahead decoding of 15 bits was found, and is presented in Table 4.12. The right bit in the table is the most recent bit shifted into the decoder buffer, which is twenty bits long.
Figure 4.10 Encoder for the (2, 4, 4, 4) code
TABLE 4.11 Encoder table for the (2, 4, 4, 4) code

<table>
<thead>
<tr>
<th>STATE</th>
<th>CODE</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>11001</td>
<td>E B</td>
</tr>
<tr>
<td>B</td>
<td>11100</td>
<td>L K</td>
</tr>
<tr>
<td>C</td>
<td>00001</td>
<td>E B</td>
</tr>
<tr>
<td>D</td>
<td>10011</td>
<td>A G</td>
</tr>
<tr>
<td>E</td>
<td>11100</td>
<td>H I</td>
</tr>
<tr>
<td>F</td>
<td>10001</td>
<td>E B</td>
</tr>
<tr>
<td>G</td>
<td>11000</td>
<td>H K</td>
</tr>
<tr>
<td>H</td>
<td>11110</td>
<td>I J</td>
</tr>
<tr>
<td>I</td>
<td>00111</td>
<td>F D</td>
</tr>
<tr>
<td>J</td>
<td>00011</td>
<td>A G</td>
</tr>
<tr>
<td>K</td>
<td>01111</td>
<td>C J</td>
</tr>
<tr>
<td>L</td>
<td>11110</td>
<td>K I</td>
</tr>
</tbody>
</table>

CHAPTER 4: Synthesis of the new asymmetrical ...
### TABLE 4.12 Decoder table for the (2, 4, 4, 4) code

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001 11100 00111 10001</td>
<td>0</td>
</tr>
<tr>
<td>00001 11100 00111 10011</td>
<td>0</td>
</tr>
<tr>
<td>00011 11001 11100 00111</td>
<td>0</td>
</tr>
<tr>
<td>00011 11001 11100 01111</td>
<td>0</td>
</tr>
<tr>
<td>00111 11000 11110 00111</td>
<td>0</td>
</tr>
<tr>
<td>00111 11000 11110 01111</td>
<td>0</td>
</tr>
<tr>
<td>10011 11001 11111 10001</td>
<td>0</td>
</tr>
<tr>
<td>10011 11001 11111 10011</td>
<td>0</td>
</tr>
<tr>
<td>10111 11000 11110 00111</td>
<td>0</td>
</tr>
<tr>
<td>10111 11000 11110 01111</td>
<td>0</td>
</tr>
<tr>
<td>11110 00111 10001 11100</td>
<td>0</td>
</tr>
<tr>
<td>11110 00111 10001 11110</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001 11100 00111 00001</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 00111 00011</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 00111 00111</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 01111 00001</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 01111 00011</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 01111 00111</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 11110 00111</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 11110 01111</td>
<td>1</td>
</tr>
<tr>
<td>00001 11100 11110 11111</td>
<td>1</td>
</tr>
</tbody>
</table>

4.31
The decoder for the (2, 4, 4, 4) code is also given in hexadecimal form in Table 4.13.

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F0F1 0F0F3 0F3C3 0F3C7 0F3C7</td>
<td>0</td>
</tr>
<tr>
<td>1E787 1E78F 1E79E 3C787 3C787</td>
<td>0</td>
</tr>
<tr>
<td>3C78F 3C79E 78787 7878F 7878F</td>
<td>0</td>
</tr>
<tr>
<td>7879E 8F0F1 8F0F3 8F3C3 8F3C3</td>
<td>0</td>
</tr>
<tr>
<td>8F3C7 9E787 9E78F 9E79E 9E79E</td>
<td>0</td>
</tr>
<tr>
<td>C7878 C7879 C78F1 C78F3 C78F3</td>
<td>0</td>
</tr>
<tr>
<td>CF0F1 CF0F3 CF3C3 CF3C7 CF3C7</td>
<td>0</td>
</tr>
<tr>
<td>E3CF1 E3CF3 E7878 E7879 E7879</td>
<td>0</td>
</tr>
<tr>
<td>E78F1 E78F3 E79E1 E79E3 E79E3</td>
<td>0</td>
</tr>
<tr>
<td>F1E38 F1E78 F1E79 F3C3C F3C78</td>
<td>0</td>
</tr>
<tr>
<td>F3C78 F3C79 F3C79 F3C79</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F1E1 0F1E3 0F1E7 0F3CF 0F3CF</td>
<td>1</td>
</tr>
<tr>
<td>1E1E1 1E1E3 1E3C3 1E3C7 1E3C7</td>
<td>1</td>
</tr>
<tr>
<td>3CF0F 3CF1E 3CF3C 78F0F 78F0F</td>
<td>1</td>
</tr>
<tr>
<td>78F1E 78F3C 79E3C 79E78 79E78</td>
<td>1</td>
</tr>
<tr>
<td>79E79 8F1E1 8F1E3 8F1E7 8F1E7</td>
<td>1</td>
</tr>
<tr>
<td>8F3CF 9E1E1 9E1E3 9E3C3 9E3C3</td>
<td>1</td>
</tr>
<tr>
<td>9E3C7 C3C3C C3C78 C3C79 C3C79</td>
<td>1</td>
</tr>
<tr>
<td>CF1E1 CF1E3 CF1E7 CF3CF CF3CF</td>
<td>1</td>
</tr>
<tr>
<td>E1E3C E1E78 E1E79 E3C3C E3C3C</td>
<td>1</td>
</tr>
<tr>
<td>E3C78 E3C79 F0F0F F0F1E F0F3C</td>
<td>1</td>
</tr>
</tbody>
</table>

**TABLE 4.13 Hex decoder table for the (2, 4, 4, 4) code**

4.2.4 The (1, 2, 5, 5, 5) modulation code

The Markov model for a binary sequence that complies with the 
\((a, \beta, \gamma, \delta, N) = (1, 2, 5, 5, 5)\) i.e. \(C_+ = 3\) and \(C_- = 2\) constraint can be obtained from the general Markov model, shown in chapter 3, with some slight modification and is shown in Fig. 4.11. \(C_+\) is the maximum positive charge deviation and \(C_-\) is the maximum negative charge deviation. If two or more states, called *old states*, have identical outgoing edges terminating in the same state, then these *old states* can be merged to a single new state [23], thus reducing the number of states. The reduced Markov model (graph G) for this sequence is shown in Fig. 4.12.
Figure 4.11 Markov model for the (1, 2, 5, 5, 5) sequence
Figure 4.12 Reduced Markov model for the (1, 2, 5, 5, 5) sequence
The connection matrix corresponding to the graph G is:

\[ A = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}. \tag{4.19}

For this sequence \( \xi = 1/2 \) and the channel capacity was calculated as 0.5850 bits/symbol. An encoder which generates this sequence with a mapping ratio of \( R = \frac{1}{2} \) would thus have a sequence utilization efficiency of \( \eta = 85.6\% \). The connection matrix \( A^2 \) corresponding to the graph \( G^2 \) is:

\[ A^2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}. \tag{4.20}

It is interesting to note that graph \( G^2 \) consists of two sets of states, with no edges between the states of the two sets. The one set consists of all the states on the odd charge levels, namely states 1, 4, 5, 8, 10 and 12, and the other set consists of all the
states on the even charge levels, namely states 2, 3, 6, 7, 9, 11 and 13. Neither of the
two sets of states have sufficient outgoing edges from each state, namely 2. The set
consisting of the states on the odd charge levels, namely states 1, 4, 5, 8, 10 and 12,
was chosen for the construction of the code. The graph consisting of the states 1, 4, 5,
8, 10 and 12 is named $G_2'$. It is clear from evaluating the row sums in (4.20) that state
8 in $G_2'$ only has one outgoing edge. However, $2^1 = 2$ outgoing edges are needed at
each state to construct the desired encoder. According to the sliding block code
algorithm, an approximate eigenvector, $v$, was found to be:

$$v = [2 1 2 1 2 2]^T.$$  \hspace{1cm} (4.21)

Equation (4.21) indicates that the states 1, 5, 10 and 12, in $G_2'$, should be split into two
offspring states, while the remaining states should not be split. The resulting graph $G_2'$
will thus consist of 10 states. The transition matrix $B_2'$ corresponding to $G_2'$ is:

$$B_2' = \begin{bmatrix}
1 & 4 & 5 & 8 & 10 & 12 \\
\phi & \phi & ++ & \phi & \phi & ++ \\
\phi & \phi & ++ & \phi & ++ & ++ \\
\phi & \phi & ++ & \phi & ++ & \phi \\
++ & \phi & ++ & \phi & ++ & \phi
\end{bmatrix}. \hspace{1cm} (4.22)$$

The splitting rule requires that all the edges in $G_2'$ which terminate in states 1, 5, 10
and 12 should be redirected to both offspring states of each state. The outgoing edges
of states 1, 5, 10 and 12 should be partitioned into two groups between the offspring
states, such that the sum of the weights of the terminal states of edges in a group is an
integer multiple of the approximate eigenvalue $2^1$, with the possible exception of one
group. For example, all the sequences in the first column of $B_2$ terminate in state 1,
which has a weight of 2. The sequences in the first row of $B_2$ denote the outgoing

4.36
edges of state 1, and should thus be partitioned into two groups according to the splitting rule. It is clear that there are two edges going to state 5 and two edges going to state 12, because both states have a weight of two. The two splitted states of state 1 are named $1^1$ and $1^2$ and the other states are numbered in a similar manner.

The same procedure as discussed for the (1, 2, 7, 7) code was applied to reduce the encoder to a minimum number of states. The name of the new state, the states which have been merged to a single state and the outgoing edges associated with these states are given in Table 4.14.

<table>
<thead>
<tr>
<th>NEW STATE</th>
<th>STATES MERGED</th>
<th>OUTGOING EDGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$1^1, 12^1$</td>
<td>++</td>
</tr>
<tr>
<td>B</td>
<td>$1^2$</td>
<td>+</td>
</tr>
<tr>
<td>C</td>
<td>$12^2$</td>
<td>±</td>
</tr>
<tr>
<td>D</td>
<td>$4, 5^2$</td>
<td>+</td>
</tr>
<tr>
<td>E</td>
<td>$5^1, 10^2$</td>
<td>+, ++</td>
</tr>
<tr>
<td>F</td>
<td>$10^1$</td>
<td>--</td>
</tr>
<tr>
<td>G</td>
<td>8</td>
<td>--</td>
</tr>
</tbody>
</table>

**Table 4.14 Summary of states merged with their associated edges**

By careful application of the abovementioned procedure it was possible to reduce the initial 10 states to 7 states. An encoder and state-independent decoder was found and the code was named (1, 2, 5, 5, 5). The FSM encoder of the (1, 2, 5, 5, 5) code is shown in Fig. 4.13 and the encoder table is shown in Table 4.15.
Figure 4.13. Encoder for the (1, 2, 5, 5, 5) code
### TABLE 4.15 Encoder table for the (1, 2, 5, 5, 5) code

<table>
<thead>
<tr>
<th>STATE</th>
<th>CODE</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>B</td>
<td>+-</td>
<td>+-</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>+-</td>
<td>+-</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>+-</td>
<td>+-</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>E</td>
</tr>
<tr>
<td>E</td>
<td>+-</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>G</td>
</tr>
<tr>
<td>F</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>G</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
A state-independent sliding-block decoder which uses look-ahead decoding of 4 bits was found, and is presented in Table 4.16. The right bit in the table is the most recent bit shifted into the decoder buffer, which is six bits long.

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>++ ++ --</td>
<td>0</td>
</tr>
<tr>
<td>++ + +</td>
<td>0</td>
</tr>
<tr>
<td>++ ++ ++</td>
<td>0</td>
</tr>
<tr>
<td>++ ++ -</td>
<td>0</td>
</tr>
<tr>
<td>++ ++ +</td>
<td>0</td>
</tr>
<tr>
<td>+ + ++ ++</td>
<td>0</td>
</tr>
<tr>
<td>+ + ++ -</td>
<td>0</td>
</tr>
<tr>
<td>+ + ++ +</td>
<td>0</td>
</tr>
<tr>
<td>+ -- ++</td>
<td>0</td>
</tr>
<tr>
<td>+ -- + +</td>
<td>0</td>
</tr>
<tr>
<td>+ -- +</td>
<td>0</td>
</tr>
<tr>
<td>-- + ++</td>
<td>0</td>
</tr>
<tr>
<td>-- ++ --</td>
<td>0</td>
</tr>
<tr>
<td>-- ++ +</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BUFFER CONTENTS</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>++ ++</td>
<td>1</td>
</tr>
<tr>
<td>++ +</td>
<td>1</td>
</tr>
<tr>
<td>++ + ++</td>
<td>1</td>
</tr>
<tr>
<td>+ -- ++</td>
<td>1</td>
</tr>
<tr>
<td>+ ++ ++</td>
<td>1</td>
</tr>
<tr>
<td>+ ++ +</td>
<td>1</td>
</tr>
<tr>
<td>+ ++ --</td>
<td>1</td>
</tr>
<tr>
<td>+ -- ++</td>
<td>1</td>
</tr>
<tr>
<td>-- + ++</td>
<td>1</td>
</tr>
<tr>
<td>-- ++ --</td>
<td>1</td>
</tr>
<tr>
<td>-- ++ +</td>
<td>1</td>
</tr>
</tbody>
</table>

TABLE 4.16 Decoder table for the (1, 2, 5, 5, 5) code

4.2.5 Other dc-free modulation codes

Attempts were made to find a charge-constrained modulation code with rate=2/3. However, the number of states in the Markov model are so vast that it made finding a code very difficult. For example, for a code complying with the (1, 2, 9, 9, 9) constraint, the channel capacity was calculated as 0.7118 bits/symbol. After application of the splitting procedure the graph $G^3$ consisted of 180 states. After merging states with the same outgoing edges the graph $G^3$ was reduced to 92 states.
The complexity of the graph $G^3$ made it impossible to develop the encoder and decoder for this code. Furthermore, an encoder of 92 states is not practical. It was found that in most cases that sequences having practical significance in terms of channel constraints had Markov models consisting of a large number of states. For a high utilization efficiency it was necessary to apply the sliding-block code algorithm which meant splitting states leading to an increase in the number of states. A coding algorithm specifically adapted to the synthesis of charge-constrained codes may solve the problem of the large number of encoder states required in these codes.
CHAPTER 5

ERROR BEHAVIOUR OF THE NEW CODES

The error propagation properties of a communications system which uses a specific modulation code can be determined by simulation on a computer. The error propagation properties of the new modulation codes which were presented in chapter 4 were determined, and are presented here. The channel model that was used in the simulations is described in paragraph 5.1, while the results are presented in paragraph 5.2. The computer programs used for the simulations are described in more detail and listed in Appendix H.

5.1 CHANNEL MODEL

The noise on the channel was assumed to be additive Gaussian noise. It was assumed
that the digital signal processing (DSP) modules in the communications system are ideal, i.e. the number of data errors caused by nonideal operation of the DSP units causing effects like timing jitter [26] were assumed to be negligible in comparison to that caused by the additive Gaussian noise on the channel. It was also assumed that the signal-to-noise ratio is large. If the probability for an error to occur is e, then the probability for two code symbols within the sliding window to be in error is \( \binom{a}{2} e^2 \), where a is the memory of the sliding window, since errors are statistically independent.

In view of the assumed high signal-to-noise ratio and corresponding small value of e, it would suffice only to consider isolated errors, since the probability of i errors to occur is of the order \( e^i \), i.e. \( e >> e^2 >> e^3 \), etc. A hard-decision decoder was assumed.

Figure 5.1 shows the transition diagram for a discrete binary symmetrical channel (BSC) as described above if the code sequence consists of symbols \( s_1 \) and \( s_2 \). A binary symmetrical channel is characterised by one parameter only, namely \( p' \), where \( p' \) denotes the probability of transition between two different symbols. Berger [27] investigated so called asymmetrical channels. The asymmetry referred to by Berger relates to channels where the probability of a 0 being converted to a 1 differs from the probability of a 1 being converted to a 0. This must not be confused with the asymmetrical channels investigated in this dissertation which deals with asymmetries in runlengths.

![Binary symmetrical channel](image)

*Figure 5.1 Binary symmetrical channel*
5.2 RESULTS ON ERROR PROPAGATION PROPERTIES

For a specific modulation code, let a single, isolated channel error have a probability \( p_0 \) of causing no data errors at the decoder output, a probability \( p_1 \) of causing one error, a probability \( p_2 \) of causing 2 errors, etc. If the mapping ratio of the modulation code is \( R = \frac{m}{n} \), and the sliding block decoder has a serial buffer of length \( b \geq n \) code bits, then a single, isolated channel error cannot cause more than \( \left\lfloor \frac{b}{n} \right\rfloor \times m \) data errors to occur at the decoder output, i.e. \( p_x = 0 \ \forall \ x > \left\lfloor \frac{b}{n} \right\rfloor \times m \). This means that

\[
\sum_{x=0}^{\left\lfloor \frac{b}{n} \right\rfloor \times m} p_x = 1. \tag{5.1}
\]

Figure 5.2 shows a graph of probability \( (p_x) \) versus number of errors \( (x) \) for the \((1, 2, 3, 3)\) code for different values of \( p \), where \( p \) denotes the probability for an input data bit to be a one. Figures 5.3 and 5.4 show probability \( (p_x) \) versus number of errors \( (x) \) for the \((1, 2, 7, 7)\) code and the \((2, 3, 6, 6)\) code, respectively, for various values of \( p \). Figures 5.5 and 5.6 show \( p_x \) against \( x \) for the \((2, 4, 4, 4)\) code and the \((1, 2, 5, 5, 5)\) code, respectively, for various values of \( p \).
CHAPTER 5: Error behaviour of the new codes

Figure 5.2 Error propagation properties of the (1, 2, 3, 3) code

Figure 5.3 Error propagation properties for the (1, 2, 7, 7) code
CHAPTER 5: Error behaviour of the new codes

Figure 5.4 Error propagation properties of the \((2, 3, 6, 6)\) code

Figure 5.5 Error propagation properties of the \((2, 4, 4, 4)\) code
The bit error rate (BER) of a data communications system at any instant is the probability for a data bit at the decoder output to be in error. If the probability for a channel error to occur is $e$, $e$ small, and the channel model of paragraph 5.1 is again used, then the BER can be approximated by:

$$\text{BER} = e \sum_{i=0}^{b} i p_i, \quad e \text{ small.}$$

(5.2)

The quantity $\sum_{i=0}^{b} i p_i$ in (5.2) is called the error multiplication factor (EMF).

Table 5.1 shows values of EMF for the modulation codes investigated. These values were verified with computer simulations.
The probability of error, $e$, for the polar binary case is given by [28]:

$$ e = \text{Erfc} \sqrt{\frac{S}{N}}. $$

From (5.2) and (5.3) the BER is given by:

$$ \text{BER} = \text{EMF} \times \text{Erfc} \sqrt{\frac{S}{N}}. $$

The BER versus Signal-to-Noise (S/N) ratio for the (1, 2, 3, 3) code and the (1, 2, 7, 7) code is given in Figures 5.7 and 5.8, respectively. Figures 5.9 and 5.10 contain the BER versus Signal-to-Noise (S/N) ratio for the (2, 3, 6, 6) code and the (2, 4, 4, 4) code, respectively. The BER versus Signal-to-Noise (S/N) ratio for the (1, 2, 5, 5, 5) code is given in Figure 5.11.
CHAPTER 5: Error behaviour of the new codes

Figure 5.7 BER versus S/N for the (1, 2, 3, 3) code

Figure 5.8 BER versus S/N for the (1, 2, 7, 7) code
CHAPTER 5: Error behaviour of the new codes

Figure 5.9 BER versus SIN for the (2, 3, 6, 6) code

Figure 5.10 BER versus SIN for the (2, 4, 4, 4) code
From the figures above it is quite clear that the (1, 2, 3, 3) code has the best error propagation properties of all the codes considered. This is logical as this code was developed with the sequence-state coding methods of Franaszek [20] and thus no look-back and look-ahead decoding is necessary in the decoder. For a single isolated channel error the (1, 2, 7, 7) code and the (2, 3, 6, 6) code will at most cause 6 and 7 errors, respectively. This is due to the look-back and look-ahead decoding used in the decoder. For the (2, 4, 4, 4) code and the (1, 2, 5, 5, 5) code only 3 and 2 steps of look-ahead decoding, respectively, are necessary for decoding. Therefore, the number of errors propagated are less than in the case of the (1, 2, 7, 7) code and the (2, 3, 6, 6) code.
CHAPTER 6

POWER SPECTRA OF MAXENTROPIC RLL BINARY ASYMMETRICAL SEQUENCES

In this chapter the power spectrum, write spectrum and power spectrum for a cosine-squared pulse of maxentropic, i.e. the maximum information rate, runlength limited (RLL) sequences will be considered. An expression for the power spectrum of maxentropic \((a, \beta, \gamma, \delta)\) sequences is derived in section 6.1. From this expression an expression for the power spectrum of maxentropic \((d, k)\) sequences is derived in section 6.2. The influence of the different parameters on the power spectrum is investigated and sequences that have spectral nulls at frequencies other than zero are also investigated in section 6.3. Power spectrum refers to the continues power spectrum unless stated otherwise. The results obtained in sections 6.1 and 6.3 are new results.

6.1
6.1 POWER SPECTRUM OF MAXENTROPIC \((a, \beta, \gamma, \delta)\) SEQUENCES

A two-level (or saturation) waveform, \(W(t) \in \{-1, +1\}\), called the write (or transmission) signal, is used to store (or transmit) information. In practise, the write signal is synchronously generated from a binary modulation sequence \(\{c(mT_c)\}\) [29]:

\[
W(t) = \sum_{m=-\infty}^{\infty} c(mT_c) P_{T_c}(t-mT_c),
\]

(6.1)

with clock period \(T_c\), where the modulating pulse function is given by:

\[
P_{T_c}(t) = \begin{cases} 1, & \text{if } 0 \leq t \leq T_c; \\ 0, & \text{otherwise}. \end{cases}
\]

(6.2)

The modulation sequence is the input to the modulator, and the output of the \((a, \beta, \gamma, \delta)\) encoder.

Gallopoulos et al [29] presented the one-step state-transition matrix \(G(D)\) for \((d_1, k_1) - (d_2, k_2)\) sequences which describe the constraints on alternate runlengths of zeros and ones, respectively. From this the power spectrum expression for maxentropic \((a, \beta, \gamma, \delta)\) sequences is derived.

**Theorem 6.1**

The power spectrum of maxentropic \((a, \beta, \gamma, \delta)\) sequences is given by:

\[
S_Y(\omega) = \frac{1}{L \sin^2(\omega/2)} \ast \frac{2a - (a^2 + b^2) - ac + bd}{a^2 + b^2},
\]

(6.3)

where

\[
L = \frac{1}{2} \left\{ \frac{a\lambda^{-\alpha+1} - (\alpha-1)\lambda^{-\alpha} - (\gamma+1)\lambda^{-\gamma} + \gamma\lambda^{-\gamma+1}}{\lambda(1 - \lambda^{-1}) (\lambda^{-\alpha} - \lambda^{-\gamma+1})} + \frac{\beta\lambda^{-\beta+1} - (\beta-1)\lambda^{-\beta} - (\delta+1)\lambda^{-\delta} + \delta\lambda^{-\delta+1}}{\lambda(1 - \lambda^{-1}) (\lambda^{-\beta} - \lambda^{-\delta+1})} \right\},
\]

(6.4)

6.2
a = 1 - su + tv, \hspace{1cm} (6.5)\\
b = sv + tu, \hspace{1cm} (6.6)\\
c = s + u, \hspace{1cm} (6.7)\\
d = t + v, \hspace{1cm} (6.8)\\

and where

\[ s = \frac{1 - \lambda^{-1}}{\lambda^{-\alpha} - \lambda^{-\gamma - 1}} \sum_{n=\alpha}^{\gamma} \lambda^{-n} \cos(\omega n), \] (6.9)\\
\[ t = \frac{1 - \lambda^{-1}}{\lambda^{-\alpha} - \lambda^{-\gamma - 1}} \sum_{n=\alpha}^{\gamma} \lambda^{-n} \sin(\omega n), \] (6.10)\\
\[ u = \frac{1 - \lambda^{-1}}{\lambda^{-\beta} - \lambda^{-\delta - 1}} \sum_{m=\beta}^{\delta} \lambda^{-m} \cos(\omega m), \] (6.11)\\
\[ v = \frac{1 - \lambda^{-1}}{\lambda^{-\beta} - \lambda^{-\delta - 1}} \sum_{m=\beta}^{\delta} \lambda^{-m} \sin(\omega m), \] (6.12)

with \( \lambda \) the largest real root of:

\[ (z^{\gamma - \alpha + 1} - 1)(z^{\delta - \beta + 1} - 1) - z^{\gamma + \delta}(z - 1)^2 = 0. \] (6.13)

Proof:

Let \( \{c(mT_c')\}, \{c(mT_c')\} \in \{-1, 1\} \), be a runlength limited sequence. We define the sequence \( \{x(mT_c')\} \) as:

6.3
\[ x(mT_c) = \frac{1}{2}[c(mT_c) - c((m-1)T_c)], \]

where \( x(mT_c) \in \{-1, 0, +1\} \). Note that \( \{x(mT_c)\} \) is not a \((a, \beta, \gamma, \delta)\) sequence, but that the \((a, \beta, \gamma, \delta)\) sequence can be obtained by the process \( |x(mT_c)| \). Let \( S_x(\omega) \) and \( S_y(\omega) \) denote the power spectrum of \( \{x(mT_c)\} \) and \( \{c(mT_c)\} \), respectively. It has been shown that [29]:

\[ S_y(\omega) = \frac{1}{\sin^2(\omega/2)} S_x(\omega). \]  

Gallopoulos et al [29] showed that in terms of "D-transforms" that the power spectrum \( S_x(D) \) is given by:

\[ S_x(D) = p(1) x ((I+G(D))^{-1} + (I+G(D^{-1}))^{-1}) u^T, \]  

where \( D = e^{i\omega} \).

From the examples in Gallopoulos et al [29] it follows that for \((a, \beta, \gamma, \delta)\) sequences:

\[ x = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}, \]

\[ u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \]

\[ G(D) = \begin{bmatrix} 0 & \sum_{n=\alpha}^{\gamma} \lambda^{-n}D^{-n} \\ \sum_{m=\beta}^{\delta} \lambda^{-m}D^{-m} & \sum_{n=\alpha}^{\gamma} \lambda^{-n} \end{bmatrix}, \]  

\[ 6.4 \]
CHAPTER 6: Power Spectra of Maxentropic RLL ...

\( p(1) = \frac{2}{E_{T_1} + E_{T_2}}, \quad (6.21) \)

where

\[
E_{T_1} = \frac{\sum_{n=a}^{\gamma} n\lambda^{-n}}{\sum_{n=a}^{\gamma} \lambda^{-n}}, \quad (6.22)
\]

and

\[
E_{T_2} = \frac{\sum_{m=\beta}^{\delta} m\lambda^{-m}}{\sum_{m=\beta}^{\delta} \lambda^{-m}}, \quad (6.23)
\]

Let

\( L = \frac{1}{p(1)}, \quad (6.24) \)

then from (6.21), (6.22) and (6.23):

\[
L = \frac{1}{2} \left\{ \sum_{n=a}^{\gamma} n\lambda^{-n} + \sum_{m=\beta}^{\delta} m\lambda^{-m} \right\} \quad (6.25)
\]

We know that

\[
\sum_{n=a}^{\gamma} \lambda^{-n} = \lambda^{-a} + \lambda^{-(a+1)} + ... + \lambda^{-\gamma} = \frac{\lambda^{-a} - \lambda^{-\gamma} - 1}{1 - \lambda^{-1}}. \quad (6.26)
\]

6.5
\[ \sum_{m=\beta}^{\gamma} \lambda^{-m} = \lambda^{-\beta} + \lambda^{-(\beta+1)} + \ldots + \lambda^{-\delta} = \frac{\lambda^{-\beta} - \lambda^{-\delta}}{1 - \lambda^{-1}}, \quad (6.27) \]

\[ \sum_{n=\alpha}^{\gamma} n\lambda^{-n} = \lambda^{-1} \left\{ a\lambda^{-\alpha+1} - (\alpha - 1)\lambda^{-\alpha} - (\gamma + 1)\lambda^{-\gamma} + \gamma\lambda^{-\gamma} \right\} \left(1 - \lambda^{-1}\right)^2, \quad \text{and} \quad (6.28) \]

\[ \sum_{m=\beta}^{\delta} m\lambda^{-m} = \lambda^{-1} \left\{ b\lambda^{-\beta+1} - (\beta - 1)\lambda^{-\beta} - (\delta + 1)\lambda^{-\delta} + \delta\lambda^{-\delta} \right\} \left(1 - \lambda^{-1}\right)^2. \quad (6.29) \]

Substituting (6.26) to (6.29) into (6.25):

\[ L = \frac{1}{2} \left\{ \frac{a\lambda^{-\alpha+1} - (\alpha - 1)\lambda^{-\alpha} - (\gamma + 1)\lambda^{-\gamma} + \gamma\lambda^{-\gamma}}{\lambda(1 - \lambda^{-1})(\lambda^{-\alpha} - \lambda^{-\gamma})} + \frac{b\lambda^{-\beta+1} - (\beta - 1)\lambda^{-\beta} - (\delta + 1)\lambda^{-\delta} + \delta\lambda^{-\delta}}{\lambda(1 - \lambda^{-1})(\lambda^{-\beta} - \lambda^{-\delta})} \right\} \quad (6.30) \]

Let

\[ a_{12} = \frac{\sum_{n=\alpha}^{\gamma} \lambda^{-n}D^{-n}}{\lambda^{-\alpha} - \lambda^{-\gamma}} = \frac{1 - \lambda^{-1}}{\lambda^{-\alpha} - \lambda^{-\gamma}} \sum_{n=\alpha}^{\gamma} \lambda^{-n}D^{-n}, \quad (6.31) \]

and

\[ a_{21} = \frac{\sum_{m=\beta}^{\delta} \lambda^{-m}D^{-m}}{\lambda^{-\beta} - \lambda^{-\delta}} = \frac{1 - \lambda^{-1}}{\lambda^{-\beta} - \lambda^{-\delta}} \sum_{m=\beta}^{\delta} \lambda^{-m}D^{-m}. \quad (6.32) \]
Then from (6.20), (6.31) and (6.32):

\[ G(D) = \begin{bmatrix} 0 & a_{12} \\ a_{21} & 0 \end{bmatrix}. \]  \hfill (6.33)

Then

\[ I + G(D) = \begin{bmatrix} 1 & a_{12} \\ a_{21} & 1 \end{bmatrix}, \]  \hfill (6.34)

and the inverse of (6.34) is given by:

\[ (I + G(D))^{-1} = \frac{1}{1 - a_{12} a_{21}} \begin{bmatrix} 1 & -a_{12} \\ -a_{21} & 1 \end{bmatrix}. \]  \hfill (6.35)

Let

\[ K_a = \frac{1}{1 - a_{12} a_{21}}, \]  \hfill (6.36)

then from (6.35) and (6.36):

\[ (I + G(D))^{-1} = \begin{bmatrix} K_a & -K_a a_{12} \\ -K_a a_{21} & K_a \end{bmatrix}. \]  \hfill (6.37)

From (6.20):
CHAPTER 6 : Power Spectra of Maxentropic RIL ...

\[ G(D^{-1}) = \begin{bmatrix}
0 & \sum_{n=\alpha}^{\gamma} \lambda^{-n}D^n \\
\sum_{m=\beta}^{\delta} \lambda^{-m}D^m & \sum_{n=\alpha}^{\gamma} \lambda^{-n}
\end{bmatrix}. \quad (6.38) \]

Let

\[ b_{12} = \frac{\sum_{n=\alpha}^{\gamma} \lambda^{-n}D^n}{\sum_{n=\alpha}^{\gamma} \lambda^{-n}} = \frac{1 - \lambda^{-1}}{\lambda^{-\alpha - \gamma - 1}} \sum_{n=\alpha}^{\gamma} \lambda^{-n}D^n, \quad (6.39) \]

and

\[ b_{21} = \frac{\sum_{m=\beta}^{\delta} \lambda^{-m}D^m}{\sum_{m=\beta}^{\delta} \lambda^{-m}} = \frac{1 - \lambda^{-1}}{\lambda^{-\beta - \delta - 1}} \sum_{m=\beta}^{\delta} \lambda^{-m}D^m. \quad (6.40) \]

Then from (6.38), (6.39) and (6.40):

\[ G(D^{-1}) = \begin{bmatrix}
0 & b_{12} \\
b_{21} & 0
\end{bmatrix}. \quad (6.41) \]

Then

\[ I + G(D^{-1}) = \begin{bmatrix}
1 & b_{12} \\
b_{21} & 1
\end{bmatrix}. \quad (6.42) \]

6.8
and the inverse of (6.42) is given by:

$$(I+G(D^{-1}))^{-1} = \frac{1}{1-b_{12}b_{21}} \begin{bmatrix} 1 & -b_{12} \\ -b_{21} & 1 \end{bmatrix}. \quad (6.43)$$

Let

$$K_b = \frac{1}{1-b_{12}b_{21}}, \quad (6.44)$$

then from (6.43) and (6.44):

$$(I+G(D^{-1}))^{-1} = \begin{bmatrix} K_b & -K_b b_{12} \\ -K_b b_{21} & K_b \end{bmatrix}. \quad (6.45)$$

Then from (6.37) and (6.45):

$$(I+G(D))^{-1} + (I+G(D^{-1}))^{-1} - I = \begin{bmatrix} K_a + K_b - 1 & -K_a a_{12} - K_b b_{12} \\ -K_a a_{21} - K_b b_{21} & K_a + K_b - 1 \end{bmatrix}. \quad (6.46)$$

From (6.18) and (6.46), follows:

$$x [(I+G(D))^{-1} + (I+G(D^{-1}))^{-1} - I] = \left[ \frac{1}{2}(K_a + K_b - 1) + \frac{1}{2}(-K_a a_{21} - K_b b_{21}) \right] \left[ \frac{1}{2}(K_a + K_b - 1) + \frac{1}{2}(-K_a a_{12} - K_b b_{12}) \right]. \quad (6.47)$$

From (6.16), (6.24), (6.47) and (6.19):

$$LS_x(D) = x[(I+G(D))^{-1} + (I+G(D^{-1}))^{-1} - I]u^T$$
Substituting for $K_a$ from (6.36) and $K_b$ from (6.44) into (6.48):

$$L_{S_\chi}(D) = \left[ \frac{1}{2}(K_a + K_b - 1) + \frac{1}{2}(-K_a a_{21} - K_b b_{21}) \right] * \left[ \frac{1}{2}(K_a + K_b - 1) + \frac{1}{2}(-K_a a_{12} - K_b b_{12}) \right]$$

$$= \frac{1}{2}(K_a + K_b - 1) + \frac{1}{2}(-K_a a_{21} - K_b b_{21}) + \frac{1}{2}(K_a + K_b - 1) + \frac{1}{2}(-K_a a_{12} - K_b b_{12})$$

$$= K_a + K_b - 1 - \frac{1}{2}(K_a a_{21} + K_b b_{21} + K_a a_{12} + K_b b_{12}) \quad (6.48)$$

Substituting for $a_{12}$ from (6.31), (6.32), (6.39) and (6.40), respectively, we obtain:

$$a_{12} = \frac{1 - \lambda^{-1}}{\lambda^{-\alpha} - \lambda^{-\gamma} - 1} \sum_{n=\alpha}^{\gamma} \lambda^{-n} \cos(n\omega) - i \frac{1 - \lambda^{-1}}{\lambda^{-\alpha} - \lambda^{-\gamma} - 1} \sum_{n=\alpha}^{\gamma} \lambda^{-n} \sin(n\omega), \quad (6.50)$$

$$a_{21} = \frac{1 - \lambda^{-1}}{\lambda^{-\beta} - \lambda^{-\delta} - 1} \sum_{m=\beta}^{\delta} \lambda^{-m} \cos(m\omega) - i \frac{1 - \lambda^{-1}}{\lambda^{-\beta} - \lambda^{-\delta} - 1} \sum_{m=\beta}^{\delta} \lambda^{-m} \sin(m\omega), \quad (6.51)$$

$$b_{12} = \frac{1 - \lambda^{-1}}{\lambda^{-\alpha} - \lambda^{-\gamma} - 1} \sum_{n=\alpha}^{\gamma} \lambda^{-n} \cos(n\omega) + i \frac{1 - \lambda^{-1}}{\lambda^{-\alpha} - \lambda^{-\gamma} - 1} \sum_{n=\alpha}^{\gamma} \lambda^{-n} \sin(n\omega), \quad (6.52)$$
Let

\[ s = \frac{1 - \lambda^{-1}}{\lambda^{-a} - \lambda^{-\gamma - 1}} \sum_{n=a}^{\gamma} \lambda^{-n} \cos(\omega n), \]  

\[ t = \frac{1 - \lambda^{-1}}{\lambda^{-a} - \lambda^{-\gamma - 1}} \sum_{n=a}^{\gamma} \lambda^{-n} \sin(\omega n), \]  

\[ u = \frac{1 - \lambda^{-1}}{\lambda^{-\beta} - \lambda^{-\delta - 1}} \sum_{m=\beta}^{\delta} \lambda^{-m} \cos(\omega m), \]  

\[ v = \frac{1 - \lambda^{-1}}{\lambda^{-\beta} - \lambda^{-\delta - 1}} \sum_{m=\beta}^{\delta} \lambda^{-m} \sin(\omega m). \]  

If we substitute (6.54) to (6.57) into (6.50) to (6.53), then:

\[ a_{12} = s - it, \]  

\[ a_{21} = u - iv, \]  

\[ b_{12} = s + it, \]  

\[ b_{21} = u + iv. \]  

From (6.49), if we substitute for \( a_{12}, a_{21}, b_{12} \) and \( b_{21} \) from (6.58) to (6.61), then:

6.11
\[ \begin{align*}
    Ls_\chi(\omega) &= \frac{1}{1-su+tv+i(sv+tu)} + \frac{1}{1-su+tv-i(sv+tu)} - 1 - \frac{1}{2} \left[ \frac{s+u-i(t+v)}{1-su+tv+i(sv+tu)} ight. \\
    &\quad \left. + \frac{s+u+i(t+v)}{1-su+tv-i(sv+tu)} \right].
\end{align*} \tag{6.62}

Let
\begin{align*}
    a &= 1 - su + tv, \quad \tag{6.63} \\
    b &= sv + tu, \quad \tag{6.64} \\
    c &= s + u, \text{ and} \quad \tag{6.65} \\
    d &= t + v. \quad \tag{6.66}
\end{align*}

Then if we substitute (6.63) to (6.66) into (6.62):
\begin{align*}
    Ls_\chi(\omega) &= \frac{1}{a+ib} + \frac{1}{a-ib} - 1 - \frac{1}{2} \left[ \frac{c-id}{a+ib} + \frac{c+id}{a-ib} \right] \\
    &= \frac{2a}{a^2+b^2} - 1 - \frac{ac-bd}{a^2+b^2} \\
    &= \frac{2a-(a^2+b^2)-ac+bd}{a^2+b^2}. \tag{6.67}
\end{align*}

From (6.15) and (6.67)
\begin{align*}
    S_\chi(\omega) &= \frac{1}{\sin^2(\omega/2)} S_\chi(\omega) \\
    &= \frac{1}{Lsin^2(\omega/2)} * \frac{2a-(a^2+b^2)-ac+bd}{a^2+b^2}. \tag{6.68}
\end{align*}

This concludes the proof. \hfill \text{Q.E.D.}
The transmitted waveform $y(t)$ is a series of pulses of duration $T_c$ with alternating polarity. We make $y(t)$ a stationary process by adding a random phase which is uniformly distributed over the interval $[0, T_c]$. Therefore, the power spectrum of $y(t)$ is obtained by multiplying $S_Y(\omega T)$ by the squared Fourier transform of a pulse of duration $T_c$ [29]. The write (or transmission) spectrum of maxentropic $(\alpha, \beta, \gamma, \delta)$ sequences is given by:

$$S_W(\omega T_c) = \frac{\sin^2(\omega T_c/2)}{(\omega T_c/2)^2} T_c^2 S_Y(\omega T_c). \quad (6.69)$$

For $T_c=1$, (6.69) becomes:

$$S_W(\omega) = \frac{\sin^2(\omega/2)}{(\omega/2)^2} S_Y(\omega) = \frac{1}{L(\omega/2)^2} \frac{2a - (a^2 + b^2) - ac + bd}{a^2 + b^2}. \quad (6.70)$$

The waveform used for transmission is not always a square pulse of duration $T_c$. In some cases a cosine-squared waveform is used [30]. This gives us an indication of how the spectrum changes if the pulses are rounded rather than square. We will derive an expression for the cosine-squared power spectrum of maxentropic $(\alpha, \beta, \gamma, \delta)$ sequences. We will start by deriving the Fourier transform of the cosine-squared waveform. The cosine-squared time function is given by:

$$f(t) = \begin{cases} \cos^2(\omega t), & \text{if } T/2 \leq t \leq T/2; \\ 0, & \text{otherwise}. \end{cases} \quad (6.71)$$

The Fourier transform is given by [28]:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$$

$$= \int_{-T/2}^{T/2} \cos^2(\omega t)e^{-j\omega t}dt$$

6.13
CHAPTER 6: Power Spectra of Maxentropic RLL...

\[ F(\omega) = \int_{-T/2}^{T/2} [\cos^3(\omega t) - j\cos^2(\omega t)\sin(\omega t)] dt \]

\[ = \int_{-T/2}^{T/2} \cos^3(\omega t) dt - j\int_{-T/2}^{T/2} \cos^2(\omega t)\sin(\omega t) dt \]

\[ = \frac{\sin(\omega T/2)}{\omega} \cdot \frac{\sin(-\omega T/2)}{\omega} \cdot \frac{\sin^3(\omega T/2)}{3\omega} + \frac{\sin^3(-\omega T/2)}{3\omega} + \]

\[ j \left( \frac{\cos^3(\omega T/2)}{3\omega} - \frac{\cos^3(-\omega T/2)}{3\omega} \right) \]

\[ = \frac{2\sin(\omega T/2)}{\omega} \cdot \frac{2\sin^3(\omega T/2)}{3\omega} + j0 \]

\[ = \frac{6\sin(\omega T/2) - 2\sin^3(\omega T/2)}{3\omega}. \]  

(6.72)

We multiply the expression of the power spectrum (6.3) by the square of the Fourier transform of the pulse shape [29]. Therefore, the expression for the power spectrum for a cosine-squared pulse is given by:

\[ S_C(\omega T_c) = \left( \frac{6\sin(\omega T_c/2) - 2\sin^3(\omega T_c/2)}{3\omega} \right)^2 S_Y(\omega T_c) \]  

(6.73)

For \( T_c = 1 \), (6.73) becomes:

\[ S_C(\omega) = \left( \frac{6\sin(\omega/2) - 2\sin^3(\omega/2)}{3\omega} \right)^2 S_Y(\omega) \]  

(6.74)

where \( S_Y(\omega) \) is given by (6.3).

The power spectrum, \( S_Y(f) \), for maxentropic (\( \alpha, \beta, \gamma, \delta \)) sequences have been plotted for different values of \( \alpha, \beta, \gamma, \delta \), and are shown in Fig. 6.1 to Fig. 6.18.
The write spectrum, $S_w(f)$, for $T_c=1$ and the cosine-squared pulse power spectrum, $S_c(f)$, for $T_c=1$, for maxentropic $(a, \beta, \gamma, \delta)$ sequences have been plotted for different values of $a$, $\beta$, $\gamma$ and $\delta$, and are shown in Fig. B.1 to Fig. B.36 in Appendix B.

Firstly, the parameters $a$, $\gamma$ and $\delta$ were kept constant in each case, while $\beta$ was varied in order to examine the influence of $\beta$ on the power spectrum. From these figures we observe that the spectra exhibit a more pronounced peak that shifts to lower frequencies with increasing $\beta$, which is a similar occurrence when increasing $d$ in $(d, k)$ sequences \[1\]. Furthermore, as $\beta$ increases, the power spectrum at zero frequency decreases.

From Fig. 6.3 and 6.4 it is seen that the (2, 4, 4, 4) sequence contains spectral nulls at $f=1/4, 1/2$ and $3/4$. The maxentropic spectrum for the (2, 8, 8, 8) sequence is shown in Fig. 6.17 and 6.18, and this sequence contains spectral nulls at $1/8, 1/4, 3/8, 1/2, 5/8, 3/4$ and $7/8$. Sequences containing spectral nulls at frequencies other than zero, like the abovementioned cases, will be investigated in section 6.3.

Secondly, the parameters $a$, $\beta$ and $\gamma$ were kept constant in each case, with $a = \beta$, while $\delta$ was varied in order to examine the influence of $\delta$ on the power spectrum. From these figures we observe that the spectra exhibit a less pronounced peak with increasing $\delta$, and furthermore, as $\delta$ increases, the power spectrum at zero frequency increases. In the mid-band the changes in the power spectrum for increasing $\delta$ are negligible.

Thirdly, the parameters $a$, $\beta$ and $\gamma$ were kept constant in each case, with $\beta > a$, while $\delta$ was varied in order to examine the influence of $\delta$ on the power spectrum. From these figures we observe that the spectra exhibit a less pronounced peak that shifts slightly to lower frequencies with increasing $\delta$, and furthermore, as $\delta$ increases, the power spectrum at zero frequency increases. In the mid-band the changes in the power spectrum for increasing $\delta$ are larger than the previous case, because the spectrum is more oscillatory for $\beta > a$.

A program to calculate the maxentropic power spectrum of $(a, \beta, \gamma, \delta)$ sequences is given in Appendix I.

6.15
Figure 6.1 Logarithmic power spectrum of maxentropic $(1, \beta, 7, 7)$ sequences, for $\beta = 1, 2, 3$ and 4.

Figure 6.2 Linear power spectrum of maxentropic $(1, \beta, 7, 7)$ sequences, for $\beta = 1, 2, 3$ and 4.
Figure 6.3  Logarithmic power spectrum of maxentropic \((2, \beta, 4, 4)\) sequences, for 
\(\beta = 1, 2, 3\) and 4.

Figure 6.4  Linear power spectrum of maxentropic \((2, \beta, 4, 4)\) sequences, for 
\(\beta = 1, 2, 3\) and 4.
Figure 6.5 Logarithmic power spectrum of maxentropic \((2, \beta, 8, 8)\) sequences, for \(\beta = 2, 3, 4\) and 5.

Figure 6.6 Linear power spectrum of maxentropic \((2, \beta, 8, 8)\) sequences, for \(\beta = 2, 3, 4\) and 5.
Figure 6.7  Logarithmic power spectrum of maxentropic (3, β, 8, 8) sequences, for β = 3, 4, 5 and 6.

Figure 6.8  Linear power spectrum of maxentropic (3, β, 8, 8) sequences, for β = 3, 4, 5 and 6.
CHAPTER 6: Power Spectra of Maxentropic RLL...

Figure 6.9 Logarithmic power spectrum of maxentropic \((2, 2, 4, \delta)\) sequences, for \(\delta = 4, 6, 8\) and 10.

Figure 6.10 Linear power spectrum of maxentropic \((2, 2, 4, \delta)\) sequences, for \(\delta = 4, 6, 8\) and 10.
CHAPTER 6: Power Spectra of Maxentropic RLL...

Figure 6.11  Logarithmic power spectrum of maxentropic \((2, 4, 4, \delta)\) sequences, for \(\delta = 4, 6, 8\) and 10.

Figure 6.12  Linear power spectrum of maxentropic \((2, 4, 4, \delta)\) sequences, for \(\delta = 4, 6, 8\) and 10.
CHAPTER 6: Power Spectra of Maxentropic RLL...

Figure 6.13 Logarithmic power spectrum of maxentropic (3, 3, 8, δ) sequences, for δ = 8, 10, 12 and 14.

Figure 6.14 Linear power spectrum of maxentropic (3, 3, 8, δ) sequences, for δ = 8, 10, 12 and 14.
CHAPTER 6: Power Spectra of Maxentropic RLL...

Figure 6.15 Logarithmic power spectrum of maxentropic (3, 5, 8, δ) sequences, for δ = 8, 10, 12 and 14.

Figure 6.16 Linear power spectrum of maxentropic (3, 5, 8, δ) sequences, for δ = 8, 10, 12 and 14.
Figure 6.17 Logarithmic power spectrum of maxentropic \((2, 8, 8, \delta)\) sequences, for \(\delta = 8, 10, 12\) and 14.

Figure 6.18 Linear power spectrum of maxentropic \((2, 8, 8, \delta)\) sequences, for \(\delta = 8, 10, 12\) and 14.
6.2 POWER SPECTRUM OF MAXENTROPIC \((d, k)\) SEQUENCES

An expression for the power spectrum of maxentropic \((d, k)\) sequences was derived by Gallopoulos et al \([29]\) and is derived here as a special case of the power spectrum of maxentropic \((a, \beta, \gamma, \delta)\) sequences as a test for the correctness of the expression of the power spectrum of maxentropic \((a, \beta, \gamma, \delta)\) sequences.

**Theorem 6.2**

The power spectrum of maxentropic \((d, k)\) sequences is given by:

\[
S_y(\omega) = \frac{1}{L \sin^2(\omega/2)} \cdot \frac{1 - |G(\omega)|^2}{|1 + G(\omega)|^2},
\]

where

\[
L = \sum_{n=d+1}^{k+1} n \lambda^{-n},
\]

and

\[
G(\omega) = \sum_{n=d+1}^{k+1} \lambda^{-n} e^{i\omega n},
\]

where \(\lambda\) is the largest real root of:

\[
z^{k+2} - z^{k+1} - z^{-d+1} + 1 = 0.
\]

**Proof:**

Let

\[
a = d+1, \quad \beta = d+1, \quad \gamma = k+1, \text{ and } \delta = k+1.
\]
We substitute (6.79) to (6.82) into (6.9) to (6.12):

\[
\begin{align*}
S &= u = \frac{1 - \lambda^{-1}}{\lambda^{-1}} \sum_{n=d+1}^{k+1} \lambda^{-n} \cos(\omega n), \quad \text{and} \\
t &= v = \frac{1 - \lambda^{-1}}{\lambda^{-1}} \sum_{n=d+1}^{k+1} \lambda^{-n} \sin(\omega n). 
\end{align*}
\]

(6.83) (6.84)

We substitute \( s = u \) and \( t = v \) into (6.5) to (6.8):

\[
\begin{align*}
a &= 1 - su + tv = 1 - s^2 + t^2, \\
b &= sv + tu = st + ts = 2st, \\
c &= s + u = s + s = 2s, \quad \text{and} \\
d &= t + v = t + t = 2t. 
\end{align*}
\]

(6.85) (6.86) (6.87) (6.88)

Substituting (6.85) to (6.88) into (6.3):

\[
S_Y(\omega) = \frac{1}{\text{L} \sin^2(\omega/2)} * \frac{2(1-s^2+t^2) - (1-s^2+t^2) - 4s^2t^2 - 2s(1-s^2+t^2)}{(1-s^2+t^2)^2 + 4s^2t^2}
\]

\[
= \frac{1}{\text{L} \sin^2(\omega/2)} * \frac{2-2s^2 + 2t^2 - 1 + 2s^2 - 2t^2 + 2s^2t^2 + 4s^2t^2 - 2s^2 + 2st^2 + 4st^2}{1 - 2s^2 + 2t^2 + 2s^2t^2 + 4s^2t^2 + 4s^2t^2}
\]

\[
= \frac{1}{\text{L} \sin^2(\omega/2)} * \frac{1 - 2s^2t^2 - s^4 - t^4 - 2s^2 + 2st^2}{1 - 2s^2 + 2t^2 + 2s^2t^2 + 4s^2t^2 + 4st^2}.
\]

(6.89)

Factoring the denominator and the numerator of (6.89):

6.26
\[ S_k(\omega) = \frac{1}{\sin^2(\omega/2)} \left[ 1 - s^2 t^2 \right] \left[ (1+s^2+t^2) - 2s(1-s^2-t^2) \right] \]
\[ = \frac{1}{\sin^2(\omega/2)} \left[ 1 - s^2 t^2 \right] \left[ 1 - 2s + s^2 + t^2 \right] \left[ 1 + 2s + s^2 + t^2 \right] \]
\[ = \frac{1}{\sin^2(\omega/2)} \left[ 1 - s^2 t^2 \right] \left[ 1 + 2s + s^2 + t^2 \right]. \] (6.90)

Let
\[ s = cx, \quad \text{and} \]
\[ t = cy, \] (6.91)
\[ \text{where from (6.83) and (6.84):} \]
\[ c = \frac{1 - \lambda^{-1}}{\lambda^{-(d+1)} - \lambda^{-(k+2)}}, \] (6.93)
\[ x = \sum_{n=d+1}^{k+1} \lambda^{-n} \cos(\omega n), \quad \text{and} \] (6.94)
\[ y = \sum_{n=d+1}^{k+1} \lambda^{-n} \sin(\omega n). \] (6.95)

From the characteristic equation of \((d, k)\) sequences
\[ z^{k+2} - z^{k+1} - z^{k-d+1} + 1 = 0, \] (6.96)
we know that \(\lambda\) is the largest real root. Thus we can see from (6.96) that
\[ \lambda^{k+2} = \lambda^{k+1} + \lambda^{k-d+1} - 1. \] (6.97)
From (6.93),

\[ c = \frac{1 - \lambda^{-1}}{\lambda^{-(d+1)} - \lambda^{-(k+2)^{\prime}}}, \quad (6.98) \]

we multiply the denominator and the numerator of (6.98) by \( \lambda^{k+2} \):

\[ c = \frac{\lambda^{k+2} - \lambda^{k+1}}{\lambda^{k-d+1} - 1}. \quad (6.99) \]

If we substitute (6.97) into (6.99):

\[ c = \frac{\lambda^{k-d+1} - 1}{\lambda^{k-d+1} - 1} = 1. \quad (6.100) \]

Then from (6.100), (6.91) and (6.92):

\[ s = x, \quad \text{and} \quad t = y. \quad (6.101, 6.102) \]

From (6.90), (6.101) and (6.102):

\[ S_y(\omega) = \frac{1}{\sin^2(\omega/2)} \ast \frac{1-x^2-y^2}{1+2x+x^2+y^2}. \quad (6.103) \]

Let

\[ G(\omega) = x + iy. \quad (6.104) \]

Then

\[ 1 - |G(\omega)|^2 = 1 - x^2 - y^2, \quad (6.105) \]

and

\[ |1+G(\omega)|^2 = (1+x)^2 + y^2 = 1 + 2x + x^2 + y^2. \quad (6.106) \]
CHAPTER 6: Power Spectra of Maxentropic RLL...

Form (6.105), (6.106) and (6.103):

\[ S_v(\omega) = \frac{1}{L \sin^2(\omega/2)} \left| \frac{1 - |G(\omega)|^2}{1 + |G(\omega)|^2} \right|^2. \]  \hspace{1cm} (6.107)

From (6.4):

\[ L = \frac{1}{2} \left[ \frac{\alpha \lambda^{-\alpha+1} - (\alpha-1) \lambda^{-\alpha} - (\gamma+1) \lambda^{-\gamma} + \gamma \lambda^{-\gamma-1}}{\lambda(1-\lambda^{-1}) (\lambda^{-\alpha} - \lambda^{-\gamma} - \gamma)} \right] + \frac{\beta \lambda^{-\beta+1} - (\beta-1) \lambda^{-\beta} - (\delta+1) \lambda^{-\delta} + \delta \lambda^{-\delta-1}}{\lambda(1-\lambda^{-1}) (\lambda^{-\beta} - \lambda^{-\delta} - \delta)}. \]  \hspace{1cm} (6.108)

Substitute (6.79) to (6.82) into (6.108):

\[ L = \frac{(d+1) \lambda^{-d} - d \lambda^{-1} - (k+2) \lambda^{-1} + (k+1) \lambda^{-(k+2)}}{\lambda(1-\lambda^{-1}) (\lambda^{-d+1} - \lambda^{-(k+2)})}. \]  \hspace{1cm} (6.109)

Multiply the denominator and the numerator of (6.109) by (1-\lambda^{-1}):

\[ L = \frac{((d+1) \lambda^{-d} - d \lambda^{-1} - (k+2) \lambda^{-1} + (k+1) \lambda^{-(k+2)}) (1-\lambda^{-1})}{\lambda(1-\lambda^{-1}) (\lambda^{-d+1} - \lambda^{-(k+2)})(1-\lambda^{-1})}. \]  \hspace{1cm} (6.110)

We know from (6.98) and (6.100) that

\[ c = \frac{1-\lambda^{-1}}{\lambda^{-d+1} - \lambda^{-(k+2)}} = 1. \]  \hspace{1cm} (6.111)

From (6.110) and (6.111):

\[ L = \frac{(d+1) \lambda^{-d} - d \lambda^{-1} - (k+2) \lambda^{-1} + (k+1) \lambda^{-(k+2)}}{\lambda(1-\lambda^{-1})^2}. \]  \hspace{1cm} (6.112)
We know that

\[
\sum_{n=d+1}^{k+1} n \lambda^{-n} = \frac{(d+1) \lambda^{-d} \lambda^{-1} - (d+1) \lambda^{-2} - (k+1) \lambda^{-1}}{\lambda (1 - \lambda^{-1})^2}.
\]  

(6.113)

From (6.112) and (6.113):

\[
L = \sum_{n=d+1}^{k+1} n \lambda^{-n}.
\]  

(6.114)

From (6.104), (6.94) and (6.95) follows

\[
G(\omega) = x + iy = \sum_{n=d+1}^{k+1} \lambda^{-n} e^{i \omega n}.
\]  

(6.115)

This concludes the proof and the result is in agreement with Gallopoulos [29]. Compare notation with Immink [1].

Q.E.D.

The influence of \(d\) and \(k\) on the power spectrum can be found in Immink [1] and Zehavi and Wolf [31].

6.3 SPECTRAL NULLS AT FREQUENCIES OTHER THAN ZERO

Information regarding the servo position of magnetic tape or disc recorders is often recorded as low-frequency components, usually called pilot tracking tones [1]. To provide space for auxiliary pilot tones it is necessary to use codes that give rise to a spectral null at an arbitrary frequency. Carasso and Huijser [32] reported the utilization of spectral null codes in write-once optical recording systems. A timing signal at 1/2
the code clock frequency was implemented in the form of a groove depth modulation, leading to a spectral null requirement at the Nyquist frequency. Marcus and Siegel investigated codes containing spectral nulls at rational submultiples of the symbol frequency [33].

The next Theorem will consider the \((a, \beta, \gamma, \delta)\) sequences containing spectral nulls.

**Theorem 6.3**

A \((a, \xi, \gamma, \xi)\) sequence, \(\gamma > a \geq 1\) and \(\xi\) an integer, will contain spectral nulls at the frequency \(f = r/\xi\), \(r < \xi\) and \(r\) an integer.

**Proof:**

From Theorem 6.1 the power spectrum of a maxentropic \((a, \beta, \gamma, \delta)\) sequence is given by:

\[
S_y(\omega) = \frac{1}{L \sin^2(\omega/2)} \frac{2a-(a^2+b^2)-ac+bd}{a^2+b^2},
\]

where \(L, a, b, c, d, s, t, u\) and \(v\) are given by (6.4) to (6.12). Substitute (6.5) to (6.8) into the numerator of (6.116):

\[
2a-(a^2+b^2)-ac+bd
\]

\[
= 2 - 2su + 2tv - 1 + 2su - 2tv - s^2u^2 - t^2v^2 - s^2v^2 - t^2u^2 - s + s^2u - stv - u + s^2u^2 - t^2u + sv^2 + tuv
\]

\[
= 1 - s^2u^2 - t^2v^2 - s^2v^2 - t^2u^2 - s + s^2u - u + s^2u^2 + t^2u + sv^2.
\]

We set
\[ \beta = \delta = \xi. \tag{6.118} \]

From (6.11) and (6.12)
\[
\begin{aligned}
u &= \frac{1 - \lambda^{-1}}{\lambda^{-\xi} - \lambda^{-\xi - 1}} \sum_{m=\xi}^{\xi} \lambda^{-m} \cos(\omega m) = \cos(\xi \omega), \quad \text{and} \\
v &= \frac{1 - \lambda^{-1}}{\lambda^{-\xi} - \lambda^{-\xi - 1}} \sum_{m=\xi}^{\xi} \lambda^{-m} \sin(\omega m) = \sin(\xi \omega). \tag{6.119} \tag{6.120}
\end{aligned}
\]

If we take the limit when \( f \to r/\xi, \ r<\xi \) and \( r \) an integer, then from (6.119) and (6.120)
\[
\begin{aligned}
u &\to \cos(2\pi r) = 1, \quad \text{and} \\
v &\to \sin(2\pi r) = 0. \tag{6.121} \tag{6.122}
\end{aligned}
\]

If we substitute (6.121) and (6.122) into (6.117) we see that the numerator:
\[
1 - s^2 u^2 - t^2 v^2 - s^2 v^2 - t^2 u^2 - s + s^2 u - u + s^2 u + t^2 u + sv^2
\]
\[
= 1 - s^2 - t^2 - s + s^2 - 1 + s + t^2 = 0, \tag{6.123}
\]
for any \( s \) and \( t \).

From the denominator of (6.116), \( L \) is a constant and \( \sin(\omega/2) \) is not zero for \( f \to r/\xi, \ r<\xi, \xi \) and \( r \) an integer. Thus it is only necessary to consider:
\[
a^2 + b^2 = 1 - 2su + 2tv + s^2 u^2 + t^2 v^2 + s^2 v^2 + t^2 u^2, \tag{6.124}
\]
where \( a \) and \( b \) are given by (6.5) and (6.6). If we substitute (6.121) and (6.122) into (6.124) we obtain:
\[ a^2 + b^2 = 1 - 2s + s^2 + t^2. \]  

(6.125)

The value in (6.125) will only be zero when \( t=0 \) and \( s=1 \) which is only true for the special case when \( \alpha = \gamma = \xi \). Thus (6.116) goes to zero if \( f \rightarrow r/\xi \) for \((\alpha, \xi, \gamma, \xi)\) sequences.

Q.E.D.

Figures 6.3 and 6.17 are examples of spectra of sequences with spectral nulls at \( f = r/\xi \). As a further example the write spectrum for maxentropic \((1, 3, 2, 3), (1, 3, 5, 3)\) and \((2, 3, 4, 3)\) sequences is given in Figures 6.19 and 6.20 and the write spectrum for maxentropic \((1, 5, 2, 5), (1, 5, 6, 5)\) and \((2, 5, 4, 5)\) sequences is given in Figures 6.21 and 6.22.

The complex running sum of a sequence with a null at frequencies other than zero is defined as [1]:

\[ z_{m,f} = \sum_{m=-\infty}^{\ell} c(mT_c) e^{-i2\pi mf}. \]  

(6.126)

A necessary and sufficient condition for a spectral null at frequency \( f=r/\xi \) is that the sequence \( \{c(mT_c)\} \) takes a finite range of values of \( z_{m,f} \). We assume that \( r \) and \( \xi \) are relatively prime and that \( r<\xi \). Thus, there is a constant \( c_i > 0 \) for which

\[ \left| \sum_{m=-\infty}^{\ell} c(mT_c) e^{-i2\pi mf} \right| < c_i \]  

(6.127)

for all sequences \( c(mT_c) \).

For the \((2, 4, 4, 4)\) code it is easy to verify that \( c_i = 3+3i \).
CHAPTER 6: Power Spectra of Maxentropic RLL ...

Figure 6.19 Logarithmic write spectrum of maxentropic \((1, 3, 2, 3), (1, 3, 5, 3)\) and \((2, 3, 4, 3)\) sequences.

Figure 6.20 Linear write spectrum of maxentropic \((1, 3, 2, 3), (1, 3, 5, 3)\) and \((2, 3, 4, 3)\) sequences.
Figure 6.21  Logarithmic write spectrum of maxentropic (1, 5, 2, 5), (1, 5, 6, 5) and (2, 5, 4, 5) sequences.

Figure 6.22  Linear write spectrum of maxentropic (1, 5, 2, 5), (1, 5, 6, 5) and (2, 5, 4, 5) sequences.
In this chapter the calculated power spectra of the new modulation codes are presented. The spectral lines, or discrete parts of the power spectrum, should be as small as possible in order to avoid the problem of phase-locked-loops in the timing recovery circuits locking onto the frequency of the spectral line.

The method of Cariolaro, Pierobon and Tronca [2] was used for calculating the power spectra of the codes. A worked example to illustrate the method is presented in section 7.1, while the power spectra for the other codes are presented in 7.2.
7.1 POWER SPECTRUM OF THE (1, 2, 3, 3) CODE

The (1, 2, 3, 3) modulation code maps every one data bit onto two code bits (see chapter 4). This code was chosen for illustration purposes because it is relatively simple (the encoder has three states, each with two outgoing edges), and the method of Cariolaro et al entails very laborious calculations, increasing rapidly in complexity as the code increases in number of states and number of outgoing edges per state.

Using the notation of Cariolaro et al, the three encoder states are called $\sigma_1$, $\sigma_2$ and $\sigma_3$. The encoder table for the (1, 2, 3, 3) code is shown in Table 7.1. An index variable, $u$, which corresponds to the row number, is also shown in the table. The finite-state machine (FSM) representation of the (1, 2, 3, 3) code is given in Fig. 7.1.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
\text{DATA} & \text{CODE} & \text{Next state} & \text{CODE} & \text{Next state} & \text{CODE} & \text{Next state} & \text{u} \\
\hline
0 & + & $\sigma_2$ & - & $\sigma_1$ & + & $\sigma_2$ & 1 \\
1 & ++ & $\sigma_3$ & ++ & $\sigma_3$ & -- & $\sigma_1$ & 2 \\
\hline
\end{tabular}
\caption{Encoder table for the (1, 2, 3, 3) code}
\end{table}

\begin{figure}[h]
\centering
\includegraphics{fig71}
\caption{FSM encoder for the (1, 2, 3, 3) code}
\end{figure}
A finite-state machine is defined in this context as a 5-tuple $\mathcal{M} = (\mathcal{D}_s, \mathcal{D}_c, \mathcal{P}, g, h)$, where:

(i) The input set is the source dictionary $\mathcal{D}_s = \mathcal{A}^m = \{b_1, \ldots, b_K\}$, where

$$K = |\mathcal{A}|^m, \quad (7.1)$$

and $|\mathcal{A}|$ denotes the cardinality of the source alphabet, $\mathcal{A}$. For the state machine of Table 7.1,

$$\mathcal{A} = \{0,1\}, \quad (7.2)$$

$$\mathcal{D}_s = \{0,1\}. \quad (7.3)$$

(ii) The output set is the code dictionary $\mathcal{D}_c = \{c_1, \ldots, c_J\}$, where

$$J \leq |\mathcal{S}^n|. \quad (7.4)$$

For the FSM of Fig 7.1,

$$\mathcal{S} = \{-1, +1\}, \quad (7.5)$$

$$\mathcal{D}_c = \{-+, +-, ++, --\}. \quad (7.6)$$

(iii) The state set $\mathcal{P} = \{\sigma_1, \ldots, \sigma_I\}$ collects the possible values of the state sequence $s(t), t \in \mathbb{Z}(T)$. For the example,

$$\mathcal{P} = \{\sigma_1, \sigma_2\}. \quad (7.7)$$

(iv) The state transition function $g : \mathcal{P} \times \mathcal{D}_s \rightarrow \mathcal{P}$ determines the next state as a function of the present state and the input source-word:
\[ s(t + T) = g\{ s(t), A(t) \}. \quad (7.8) \]

(v) The output function \( h : \mathcal{P} \times \mathcal{S}_s \rightarrow \mathcal{S}_c \) determines the output code-word in terms of state and input source-word:

\[ C(t) = h\{ s(t), A(t) \}. \quad (7.9) \]

For each source-word \( b_u \in \mathcal{S} \), an output matrix \( \Gamma_u \), and a state transition matrix \( \mathbf{E}_u \) are defined:

(i) The output matrix \( \Gamma_u \) is of dimension \( U \times n \), where \( U \) is the number of states of the encoder and \( n \) is the codeword length, as in chapter 2. The \( i \)-th row of \( \Gamma_u \) is the code-word

\[ c_{iu} = h(\sigma_i, b_u). \]

(ii) The state transition matrix \( \mathbf{E}_u \) is an \( U \times U \) binary matrix with \((i, j)\)th entry

\[ E_u(i, j) = \begin{cases} 1, & \text{if } g(\sigma_i, b_u) = \sigma_j, \\ 0, & \text{otherwise} \end{cases} \quad (7.10) \]

For the \((1, 2, 3, 3)\) encoder:

\[ \Gamma_1 = \begin{bmatrix} -1 & +1 \\ +1 & -1 \\ -1 & +1 \end{bmatrix}, \quad \Gamma_2 = \begin{bmatrix} +1 & +1 \\ +1 & +1 \\ -1 & -1 \end{bmatrix} \quad (7.11) \]

and

\[ \mathbf{E}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{E}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}. \quad (7.12) \]

If the probability for a data bit at the input of the encoder to be a '1' is \( p \), and the

7.4
probability for the bit to be a '0' is \( q \), then \( q = 1 - p \), and the source-word probabilities are given by:

\[
q_u = p_u q_u^{m-M_u},
\]

where \( M_u \) denotes the number of ones in the source-word \( b_u \).

For the (1, 2, 3, 3) code,

\[
q_1 = q, \quad q_2 = p.
\]

The \((i,j)\)th entry of the transition probability matrix \( \Pi \) is the conditional probability

\[
P( s(t + T) = \sigma_j \mid s(t) = i ) .
\]

The matrix \( \Pi \) can be calculated by

\[
\Pi = \sum_{u = 1}^{K} q_u E_u .
\]

Where \( K \) is the number of data words, \( K = 2^m \). For the (1, 2, 3, 3) code, from (7.14) and (7.16), the matrix \( \Pi \) was found to be

\[
\Pi = \begin{bmatrix}
0 & q & p \\
q & 0 & p \\
p & q & 0
\end{bmatrix} .
\]

If the data source is random, then \( p = q = 0.5 \), and
The row vector $\mathbf{\pi}$ of size $U$ contains the first-order state probabilities:

$$\mathbf{\pi}(i) = P(s(t) = \sigma_i).$$  \hfill (7.19) 

The vector $\mathbf{\pi}$ is the normalized solution of the matrix equation

$$\mathbf{\pi} = \mathbf{\pi} \mathbf{\Pi}.$$  \hfill (7.20) 

For the $(1, 2, 3, 3)$ code with $\mathbf{\Pi}$ as in (7.18), $\mathbf{\pi}$ was determined:

$$\mathbf{\pi} = \left[ \begin{array}{ccc} 1/3 & 1/3 & 1/3 \end{array} \right].$$  \hfill (7.21) 

The $k$-step transition probabilities collected in the matrix $\mathbf{\Pi}_k$ are defined as:

$$\mathbf{\Pi}_k(i,j) = P(s(t + kT) = \sigma_j | s(t) = \sigma_i).$$  \hfill (7.22) 

The matrix $\mathbf{\Pi}_k$ can be calculated by

$$\mathbf{\Pi}_k = \mathbf{\Pi}^k.$$  \hfill (7.23) 

The limiting value of $\mathbf{\Pi}_k$, i.e. $\lim_{k \to \infty} \mathbf{\Pi}_k$, can be calculated by:

$$\mathbf{\Pi}_\infty = \mathbf{w} \mathbf{\pi},$$  \hfill (7.24) 

where $\mathbf{w}$ is a column vector of size $U$ with all the entries equal to one. The rows of $\mathbf{\Pi}_\infty$ are therefore equal to $\mathbf{\pi}$. 

7.6
For the $(1, 2, 3, 3)$ code,

$$\Pi_\infty = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}. \quad (7.25)$$

The matrix $D$ is defined as the diagonal matrix $D = \text{diag}(\pi(1), \ldots, \pi(U))$. For the $(1, 2, 3, 3)$ code:

$$D = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}. \quad (7.26)$$

The matrix $B(z)$ can be calculated by:

$$B(z) = \{ zI - (\Pi - \Pi_\infty) \}^{-1}, \quad (7.27)$$

where $I$ is the identity matrix. For the $(1, 2, 3, 3)$ code:

$$B(z) = \begin{bmatrix} z^\frac{1}{3} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & z^\frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & z^\frac{1}{3} \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} x & y & y \\ y & x & y \\ y & y & x \end{bmatrix}. \quad (7.28)$$

where
The matrix $Y(z)$ can be calculated by

$$Y(z) = \frac{1}{2} \left( C_0 - m_c^\mathsf{T} m_c \right) + C_1 B(z) C_2,$$

where $m_c$ is the code-word mean value:

$$m_c = \sum_{u=1}^{K} q_u x \Gamma_u,$$

and $C_0$, $C_1$, and $C_2$ can be determined by:

$$C_0 = \sum_{u=1}^{K} q_u \Gamma_u^\mathsf{T} D \Gamma_u,$$

$$C_1 = \sum_{u=1}^{K} q_u \Gamma_u^\mathsf{T} D E_u (I - \Pi_u),$$

and

$$C_2 = \sum_{u=1}^{K} q_u \Gamma_u.$$
It is clear from (7.13) and (7.14) that when $p = \frac{1}{2}$ for the (1, 2, 3, 3) code, then

$$q_u = \frac{1}{2} \quad \forall \ 1 \leq u \leq 2. \quad (7.36)$$

Substituting these values for $q_u$ together with $\pi$ from (7.21) and $\Gamma_u$ from (7.11) into (7.32):

$$m_c = \frac{1}{2} \left[ \begin{array}{ccc} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array} \right] \sum_{u=1}^{K} \Gamma_u$$

$$m_c = \left[ \begin{array}{cc} 0 & \frac{1}{3} \end{array} \right]. \quad (7.37)$$

If values for $q_u$, $\Gamma_u$, and $D$ are obtained from (7.36), (7.11), and (7.26) respectively and substituted into (7.33), then $C_0$ is obtained for the (1, 2, 3, 3) code:

$$C_0 = \left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]. \quad (7.38)$$

When values for $q_u$, $\Gamma_u$, $D$, $E_u$, and $\Pi_\infty$ is substituted from (7.36), (7.11), (7.26) (7.12), and (7.25) respectively into (7.34), then $C_1$ is obtained for the (1, 2, 3, 3) code:

$$C_1 = \left[ \begin{array}{ccc} 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{4}{9} & \frac{2}{9} & \frac{2}{9} \end{array} \right]. \quad (7.39)$$

Similarly, if $q_u$ from (7.36) and $\Gamma_u$ from (7.11) is substituted into (7.35), then $C_2$ for the (1, 2, 3, 3) code is obtained:

$$C_2 = \left[ \begin{array}{cc} 0 & 1 \\ 1 & 0 \\ -1 & 0 \end{array} \right]. \quad (7.40)$$
CHAPTER 7: Power spectra of the new codes

Values for $C_0$, $C_1$, $C_2$, and $B(z)$ for the $(1, 2, 3, 3)$ code can be obtained from (7.38), (7.39), (7.40) and (7.28) respectively. These values were substituted into (7.29) to obtain $Y(z)$ for the $(1, 2, 3, 3)$ code:

$$Y(z) = \begin{bmatrix} \frac{1}{2} + \frac{2}{3}z & 0 \\ 0 & \frac{4}{9} + \frac{4}{9}z \end{bmatrix}.$$  
(7.41)

where

$$\varphi = \frac{-z}{z(z+1)/2}.$$  
(7.42)

Once $Y(z)$ is known, the continuous part of the code-word power spectrum can be calculated by:

$$W_c^{(c)}(f) = T [ Y(z) + Y^T(z^{-1}) ], \ z = \exp(i2\pi/T).$$  
(7.43)

If $Y(z)$ from (7.41) is substituted into (7.43), then $W_c^{(c)}(f)$ is obtained for the $(1, 2, 3, 3)$ code:

$$W_c^{(c)}(f) = T \begin{bmatrix} 1 + \frac{2}{3}z & \frac{2}{3}z' \\ 0 & \frac{8}{9} + \frac{4}{9}z + \frac{4}{9}z' \end{bmatrix}.$$  
(7.44)

where

$$z' = \frac{-2z}{(z+2)}.$$  
(7.45)

The parallel-to-serial converter in Fig 2.1 is an $n$-input, one-output interpolating filter with frequency response

$$V(f) = \frac{1}{n} [1, \exp(-i2\pi fT_c), \ldots, \exp(-i2\pi f(n-1)T_c)]^T.$$  
(7.46)
The continuous part of the code-symbol power spectrum is given by:

\[ W^{(c)}(f) = V^*(f) W^{(c)}(f) V(f), \]  

(7.47)

where \( V^*(f) \) is the conjugate transpose of \( V(f) \). For the (1, 2, 3, 3) code, \( n = 2 \), and \( V(f) \) is therefore:

\[
V(f) = \frac{1}{2} \begin{bmatrix}
1 \\
\cos(2\pi f T_c) - i \sin(2\pi f T_c)
\end{bmatrix},
\]

(7.48)

because \( \exp(-i2\pi f T_c) = \cos(2\pi f T_c) - i \sin(2\pi f T_c) \).

(7.49)

If \( W^{(c)}(f) \) from (7.44), \( V(f) \) from (7.48) and \( V^*(f) \) are substituted into (7.47), then \( W^{(c)}(f) \), or the code-symbol power spectrum for the (1, 2, 3, 3) code is obtained:

\[ W^{(c)}(f) = \frac{T}{36} (17 + 10\varphi + 10\varphi'). \]

(7.50)

After substituting for \( \varphi \) from (7.42) and \( \varphi' \) from (7.45) and some manipulation the following is obtained:

\[
W^{(c)}(f) = \frac{T}{12} \left[ \frac{-z^2 + 7.5z - 1}{z^2 + 2.5z + 1} \right].
\]

(7.51)

If \( \omega_T = 2\pi f T \), then

\[ z = \exp(i\omega_T) = \cos(\omega_T) + i \sin(\omega_T). \]

(7.52)

It also follows that for the (1, 2, 3, 3) code:
\[ \omega_T = n \omega_c = 2 \omega_c, \quad \text{and} \]
\[ T = n T_c = 2 T_c. \]  

(7.53)  
(7.54)  

After substituting for \( \omega_T \) from (7.53) and \( T \) from (7.54), into (7.52) the following expression for \( W^{(c)}(f) \) is obtained:

\[
W^{(c)}(f) = \frac{T_c}{6} \left[ -e^{\frac{i4\omega_c}{T_c}} +7.5e^{\frac{i2\omega_c}{T_c}} -1 \right].
\]  

(7.55)  

Substituting for \( \omega_c = 2\pi f T_c \), the following expression for \( W^{(c)}(f) \) is obtained:

\[
W^{(c)}(f) = \frac{T_c}{6} \left[ -\cos 8\pi f T_c +7.5 \cos 4\pi f T_c \cdot 1 + i(-\sin 8\pi f T_c +7.5 \sin 4\pi f T_c) \right].
\]  

(7.56)  

Therefore, if the code-symbol period \( T_c = 1 \) is substituted into (7.56), then the continuous code-symbol power spectrum for the \( (1, 2, 3, 3) \) code is obtained, with the frequency axis normalized to \( f_c = \frac{1}{T_c} = 1 \):

\[
W^{(c)}(f) = \frac{1}{6} \left[ -\cos 8\pi f +7.5 \cos 4\pi f \cdot 1 + i(-\sin 8\pi f +7.5 \sin 4\pi f) \right].
\]  

(7.57)  

The discrete part of the code-word power spectrum is given by:

\[
W^{(d)}(f) = m_c^{T} m_c \sum_{k=\infty}^{\infty} \delta(f - kF), \quad F = \frac{1}{T}.
\]  

(7.58)  

The discrete part of the code-symbol power spectrum can then be calculated by:

\[
W^{(d)}(f) = V^{*}(f) W^{(d)}_{c}(f) V(f).
\]  

(7.59)
However, from (7.37) \( m_c = [0 \ 1/3] \) for the (1, 2, 3, 3) code if \( p=0.5 \). From (7.58) and (7.59), it follows that \( W^{(d)}(f) \) for the (1, 2, 3, 3) code is given by

\[
W^{(d)}(f) = \begin{bmatrix}
\frac{1}{36} & \frac{1}{36}
\end{bmatrix},
\]

(7.60)

at the code-symbol frequency \( \frac{1}{2} f_c \) and \( f_c \), respectively. The continuous power spectrum for the (1, 2, 3, 3) code is the magnitude of (7.57) and the discrete components are given by (7.60). The linear continuous power spectrum for the (1, 2, 3, 3) code, for \( p=0.5 \), is shown in Fig. 7.2. The logarithmic and linear power spectrum for the (1, 2, 3, 3) code for \( p=0.3 \) and \( p=0.7 \) is given in the next section.

![Figure 7.2 Linear power spectrum for the (1, 2, 3, 3) code for \( p=0.5 \)](image-url)
7.2 POWER SPECTRA FOR THE NEW CODES

The power spectra for the new codes that were presented in chapter 4 are shown in this section. The continuous part of the power spectrum is shown as a graph for each code, whereas the discrete part or spectral line amplitudes are presented in table form. A computer program which implements the method described in section 7.1 was used for the calculations. The program was first written by Whalley [34] and modified to have more output formats. It is clear from Fig 7.2 that the power spectrum for the (1, 2, 3, 3) code is symmetrical around \( \frac{1}{2} f_c \). Since this is true for all the codes considered, only one half of the symmetrical power spectrum is presented in each case. The power spectrum is also presented for \( p = 0.3, p = 0.5, \) and \( p = 0.7 \) in each case, in order to get an idea of how the power spectrum fluctuates as \( p \) changes, as in a practical communications system. Both the logarithmic and linear power spectra are given for completeness. The power spectrum, \( S_Y(f) \), for the (1, 2, 3, 3), (1, 2, 7, 7), (2, 3, 6, 6), (2, 4, 4, 4) and (1, 2, 5, 5, 5) modulation codes are presented in Fig 7.3 to Fig 7.12. The \( write \) spectrum, \( S_w(f) \), and the cosine-squared pulse power spectrum, \( S_C(f) \), for the (1, 2, 3, 3), (1, 2, 7, 7), (2, 3, 6, 6), (2, 4, 4, 4) and (1, 2, 5, 5, 5) modulation codes are presented in Fig C.1 to Fig C.20 in Appendix C.

The \( write \) spectrum, \( S_w(f) \), for the maxentropic sequence, the \( write \) spectrum, \( S_W(f) \), for the specific modulation code and the cosine-squared pulse power spectrum, \( S_C(f) \), for the specific modulation code are presented on a single graph for \( p=0.5 \). These spectra are given for the (1, 2, 3, 3), (1, 2, 7, 7), (2, 3, 6, 6) and (2, 4, 4, 4) modulation codes in Fig. 7.13 to Fig. 7.20, respectively. The \( write \) spectrum, \( S_w(f) \), for the (1, 2, 5, 5, 5) modulation code and the cosine-squared pulse power spectrum, \( S_C(f) \), for the (1, 2, 5, 5, 5) modulation code are presented on a single graph for \( p=0.5 \) in Figures 7.21 and 7.22.

The \( write \) spectrum, \( S_w(f) \), for the maxentropic sequence, the \( write \) spectrum, \( S_W(f) \), for the specific modulation code and the cosine-squared pulse power spectrum, \( S_C(f) \), for the specific modulation code are presented on a single graph for \( p=0.3 \) and \( p=0.7 \). These spectra are given for the (1, 2, 3, 3), (1, 2, 7, 7), (2, 3, 6, 6) and (2, 4, 4, 4) modulation codes in Fig. C.21 to Fig. C.36, Appendix C.
CHAPTER 7: Power spectra of the new codes

**Figure 7.3** Logarithmic power spectrum for the (1, 2, 3, 3) code

**Figure 7.4** Linear power spectrum for the (1, 2, 3, 3) code
CHAPTER 7: Power spectra of the new codes

Figure 7.5  Logarithmic power spectrum for the (1, 2, 7, 7) code

Figure 7.6  Linear power spectrum for the (1, 2, 7, 7) code

7.16
CHAPTER 7: Power spectra of the new codes

Figure 7.7 Logarithmic power spectrum for the (2, 3, 6, 6) code

Figure 7.8 Linear power spectrum for the (2, 3, 6, 6) code
CHAPTER 7: Power spectra of the new codes

Figure 7.9 Logarithmic power spectrum for the (2, 4, 4, 4) code

Figure 7.10 Linear power spectrum for the (2, 4, 4, 4) code
CHAPTER 7: Power spectra of the new codes

Figure 7.11 Logarithmic power spectrum for the \((1, 2, 5, 5, 5)\) code

Figure 7.12 Linear power spectrum for the \((1, 2, 5, 5, 5)\) code
CHAPTER 7: Power spectra of the new codes

Figure 7.13 Logarithmic $S_W(f)$ for maxentropic $(1, 2, 3, 3)$ sequences, $S_W(f)$ for the $(1, 2, 3, 3)$ code and $S_C(f)$ for the $(1, 2, 3, 3)$ code for $p=0.5$

Figure 7.14 Linear $S_W(f)$ for maxentropic $(1, 2, 3, 3)$ sequences, $S_W(f)$ for the $(1, 2, 3, 3)$ code and $S_C(f)$ for the $(1, 2, 3, 3)$ code for $p=0.5$
CHAPTER 7: Power spectra of the new codes

Figure 7.15 Logarithmic $S_W(f)$ for maxentropic $(1, 2, 7, 7)$ sequences, $S_W(f)$ for the $(1, 2, 7, 7)$ code and $S_C(f)$ for the $(1, 2, 7, 7)$ code for $p=0.5$

Figure 7.16 Linear $S_W(f)$ for maxentropic $(1, 2, 7, 7)$ sequences, $S_W(f)$ for the $(1, 2, 7, 7)$ code and $S_C(f)$ for the $(1, 2, 7, 7)$ code for $p=0.5$
CHAPTER 7: Power spectra of the new codes

Figure 7.17 Logarithmic $S_w(f)$ for maxentropic (2, 3, 6, 6) sequences, $S_w(f)$ for the (2, 3, 6, 6) code and $S_c(f)$ for the (2, 3, 6, 6) code for $p=0.5$

Figure 7.18 Linear $S_w(f)$ for maxentropic (2, 3, 6, 6) sequences, $S_w(f)$ for the (2, 3, 6, 6) code and $S_c(f)$ for the (2, 3, 6, 6) code for $p=0.5$
CHAPTER 7: Power spectra of the new codes

Figure 7.19 Logarithmic \( S_w(f) \) for maxentropic (2, 4, 4, 4) sequences, \( S_w(f) \) for the (2, 4, 4, 4) code and \( S_c(f) \) for the (2, 4, 4, 4) code for \( p=0.5 \)

Figure 7.20 Linear \( S_w(f) \) for maxentropic (2, 4, 4, 4) sequences, \( S_w(f) \) for the (2, 4, 4, 4) code and \( S_c(f) \) for the (2, 4, 4, 4) code for \( p=0.5 \)
CHAPTER 7: Power spectra of the new codes

Figure 7.21 Logarithmic $S_W(f)$ for the (1, 2, 5, 5, 5) code and $S_C(f)$ for the (1, 2, 5, 5, 5) code for $p=0.5$

Figure 7.22 Linear $S_W(f)$ for the (1, 2, 5, 5, 5) code and $S_C(f)$ for the (1, 2, 5, 5, 5) code for $p=0.5$
The *write* spectrum, $S_W(f)$, for the $(1, 2, 5, 5, 5)$ modulation code and the cosine-squared pulse power spectrum, $S_C(f)$, for the $(1, 2, 5, 5, 5)$ modulation code are presented on a single graph for $p=0.3$ and $p=0.7$ in Fig. C.37 to Fig. C.40, Appendix C.

The measured logarithmic *write* spectrum, $S_W(f)$, for the $(1, 2, 7, 7)$ code is shown in Fig. 7.23. Details of the method of measurement are given in Appendix K.

![Figure 7.23 Measured logarithmic write spectrum for the $(1, 2, 7, 7)$ code](image)

Spectral lines are present at multiples of the code-word rate $f = \frac{1}{n} = f'_n$. Table 7.2 contains values of spectral line amplitudes rounded to 3 digits for the $(1, 2, 3, 3)$, $(2, 3, 6, 6)$ and $(1, 2, 5, 5, 5)$ modulation codes for different values of $p$. Table 7.3 contains values of spectral line amplitudes rounded to 3 digits for the $(1, 2, 7, 7)$ modulation code for different values of $p$ and Table 7.4 contains values of spectral line amplitudes for the $(2, 4, 4, 4)$ modulation code for different values of $p$. 

7.25
### TABLE 7.2 Discrete components for some of the new codes

<table>
<thead>
<tr>
<th>CODE</th>
<th>( \frac{1}{2} f_c )</th>
<th>( f_c )</th>
<th>( \frac{1}{2} f_c )</th>
<th>( f_c )</th>
<th>( \frac{1}{2} f_c )</th>
<th>( f_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 2, 3, 3)</td>
<td>0.026</td>
<td>0.015</td>
<td>0.028</td>
<td>0.028</td>
<td>0.015</td>
<td>0.026</td>
</tr>
<tr>
<td>(2, 3, 6, 6)</td>
<td>0.002</td>
<td>0</td>
<td>0.008</td>
<td>0.003</td>
<td>0.027</td>
<td>0.01</td>
</tr>
<tr>
<td>(1, 2, 5, 5, 5)</td>
<td>0</td>
<td>0.005</td>
<td>0</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### TABLE 7.3 Discrete components for the (1, 2, 7, 7) code

<table>
<thead>
<tr>
<th>PROBABILITY</th>
<th>( \frac{1}{4} f_c )</th>
<th>( \frac{1}{2} f_c )</th>
<th>( \frac{3}{4} f_c )</th>
<th>( f_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 0.3</td>
<td>0.022</td>
<td>0.018</td>
<td>0.004</td>
<td>0.018</td>
</tr>
<tr>
<td>P = 0.5</td>
<td>0.028</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>P = 0.7</td>
<td>0.032</td>
<td>0.001</td>
<td>0.004</td>
<td>0.001</td>
</tr>
</tbody>
</table>

### TABLE 7.4 Discrete components for the (2, 4, 4, 4) code

<table>
<thead>
<tr>
<th>PROBABILITY</th>
<th>( \frac{1}{5} f_c )</th>
<th>( \frac{2}{5} f_c )</th>
<th>( \frac{3}{5} f_c )</th>
<th>( \frac{4}{5} f_c )</th>
<th>( f_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 0.3</td>
<td>0.033</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>P = 0.5</td>
<td>0.025</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>P = 0.7</td>
<td>0.016</td>
<td>0</td>
<td>0.007</td>
<td>0.007</td>
<td>0</td>
</tr>
</tbody>
</table>

7.26
The aim of this study was to use information theoretical methods to find values of channel capacity for sequences complying with asymmetrical runlength constraints and to develop modulation codes for generating these sequences. In write-once and erasible optical recording there is an asymmetry between marks and non-marks and thus the need for a modulation code with asymmetrical runlength constraints.

The study started with the investigation of runlength limited binary asymmetrical sequences which led to the derivation of a generating function. From this the characteristic equation was derived leading to the channel capacity. Some relations relating the capacity of different asymmetrical sequences were derived as well as
relations relating these sequences to \((d, k)\) sequences. Binary asymmetrical sequences of which the running digital sum is limited were investigated, leading to a general Markov model and thus channel capacities. Several tables containing values of channel capacity were presented for these sequences.

The detailed synthesis of several new modulation codes satisfying various input restrictions were presented. Table 8.1 is a summary of the new modulation codes.

<table>
<thead>
<tr>
<th>CODE</th>
<th>R</th>
<th>(\eta)</th>
<th>(\alpha)</th>
<th>(\beta)</th>
<th>(\gamma)</th>
<th>(\delta)</th>
<th>(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 2, 3, 3)</td>
<td>1/2</td>
<td>83.4%</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(1, 2, 7, 7)</td>
<td>3/4</td>
<td>94.2%</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(2, 3, 6, 6)</td>
<td>1/2</td>
<td>91.4%</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(2, 4, 4, 4)</td>
<td>1/5</td>
<td>87.7%</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(1, 2, 5, 5, 5)</td>
<td>1/2</td>
<td>85.6%</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

TABLE 8.1 Summary of the new modulation codes

Consider a channel for which the minimum runlength of ones required is two and the maximum runlength of zeros and ones does not exceed 4. For this channel the \((d, k) = (1, 3)\) Miller code [35] can be used with rate 1/2. This code has 4 encoder states (NRZ format) and the error propagation is limited to at most 1 bit. However, for this channel the \((1, 2, 3, 3)\) code can also be used with a rate 1/2. For this code the maximum runlength of ones and zeros is limited to 3 as opposed to the 4 of the \((1, 3)\) code. The encoder for the \((1, 2, 3, 3)\) code consists of 3 states and the error propagation is limited to at most one bit.

Secondly, consider a channel for which the minimum runlength of ones required is two and the maximum runlength of zeros and ones does not exceed 8. For this specific channel the \((d, k) = (1, 7)\) code can be used with rate 2/3. This code has 5 encoder states (NRZI format) and the error propagation is limited to at most 5 bits [35].

8.2
However, for this channel the \((1, 2, 7, 7)\) code can be used with a rate 3/4. For this code the maximum runlength of ones and zeros is limited to 7 as opposed to the 8 of the \((1, 7)\) code. The encoder for the \((1, 2, 7, 7)\) code consists of 6 states and the error propagation is limited to at most six bits. Thus the new code achieves an increase of 12\% in the data rate and a 12\% increase in detection window over the \((1, 7)\) code.

Thirdly, consider a channel for which the minimum runlength of ones required is three and the maximum runlength of zeros and ones must not exceed 8. For this specific channel the \((d, k) = (2, 7)\) code can be used with rate 1/2. However, for this channel the \((2, 3, 6, 6)\) code can also be used with a rate 1/2. For this code the maximum runlength of ones and zeros is limited to 6 as opposed to the 8 of the \((2, 7)\) code.

The \((1, 2, 5, 5, 5)\) code with a rate 1/2 has 7 encoder states and the error propagation is limited to at most three bits. The Modified Squared \((M^2)\) code of Mallinson and Miller [36] has a digital sum variation of 6 and the encoder consists of 10 states. Ferreira [37] modified the \(M^2\) code to a \((1, 4, 6)\), \textit{i.e.} \(C=3\), code consisting of 8 encoder states and the error propagation is limited to at most three bits. Thus the \((1, 2, 5, 5, 5)\) code has a tighter charge constraint.

A new expression for the power spectrum of maxentropic runlength limited binary asymmetrical sequences was derived. Several spectra were plotted to investigate the influence of the different parameters on the spectrum. It was also found that asymmetrical sequences complying to very specific parameters contained spectral nulls at frequencies other than zero.

The power spectra for the new codes were also presented. The code spectra are in good agreement with the spectra of their maxentropic counterparts. It was also found that the rounded pulse shape had little influence on the shape of the power spectrum, except that it contained less energy. It was also found that as the probability for an input data bit to be a one decreased, the peaks of the spectrum were accentuated. When the probability for an input data bit to be a one increased, the valleys of the spectrum were accentuated.
Using information theoretical methods, values of channel capacity for sequences complying with asymmetrical runlength constraints were found. Five new modulation codes were developed for generating asymmetrical sequences complying to different runlength constraints. Thus the aim of this study was achieved.

Topics for further research could include deriving a generating function for binary asymmetrical sequences of which the running digital sum is limited, a better representation i.e. Markov model with less states for these sequences and a new coding algorithm specifically aimed at the synthesis of codes for generating these sequences. Furthermore, the synthesis of error-correcting codes satisfying the input restrictions considered in this study and the synthesis of charge constrained binary asymmetrical codes with specific spectral shaping properties such as spectral nulls at frequencies other than zero can be investigated. Another topic of interest could be the derivation of an expression for the power spectrum of maxentropic binary asymmetrical sequences of which the running digital sum is limited and an investigation of the usefulness of these sequences on optic fibre channels.
REFERENCES


R.2


R.3
References


