

---

---

**COMMUNICATIONS AND  
CRYPTOGRAPHY**

*Two Sides of One Tapestry*

---

---

# **THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE**

## **COMMUNICATIONS AND INFORMATION THEORY**

*Consulting Editor*  
**Robert Gallager**

### *Other books in the series:*

- WIRELESS AND MOBILE COMMUNICATIONS**, Jack M. Holtzman and David J. Goodman  
ISBN: 0-7923-9464-X
- INTRODUCTION TO CONVOLUTIONAL CODES WITH APPLICATIONS**, Ajay Dholakia  
ISBN: 0-7923-9467-4
- CODED-MODULATION TECHNIQUES FOR FADING CHANNELS**, S. Hamidreza Jamali,  
and Tho Le-Ngoc  
ISBN: 0-7923-9421-6
- ELLIPTIC CURVE PUBLIC KEY CYRPTOSYSTEMS**, Alfred Menezes  
ISBN: 0-7923-9368-6
- SATELLITE COMMUNICATIONS: Mobile and Fixed Services**, Michael Miller, Branka Vucetic  
and Les Berry  
ISBN: 0-7923-9333-3
- WIRELESS COMMUNICATIONS: Future Directions**, Jack M. Holtzman and David J. Goodman  
ISBN: 0-7923-9316-3
- DISCRETE-TIME MODELS FOR COMMUNICATION SYSTEMS INCLUDING ATM**,  
Herwig Bruneel and Byung G. Kim  
ISBN: 0-7923-9292-2
- APPLICATIONS OF FINITE FIELDS**, Alfred J. Menezes, Ian F. Blake, XuHong Gao, Ronald  
C. Mullin, Scott A. Vanstone, Tomik Yaghoobian  
ISBN: 0-7923-9282-5
- WIRELESS PERSONAL COMMUNICATIONS**, Martin J. Feuerstein, Theodore S. Rappaport  
ISBN: 0-7923-9280-9
- SEQUENCE DETECTION FOR HIGH-DENSITY STORAGE CHANNEL**, Jaekyun Moon, L.  
Richard Carley  
ISBN: 0-7923-9264-7
- DIGITAL SATELLITE COMMUNICATIONS SYSTEMS AND TECHNOLOGIES: Military  
and Civil Applications**, A. Nejat Ince  
ISBN: 0-7923-9254-X
- IMAGE AND TEXT COMPRESSION**, James A. Storer  
ISBN: 0-7923-9243-4
- VECTOR QUANTIZATION AND SIGNAL COMPRESSION**, Allen Gersho, Robert M. Gray  
ISBN: 0-7923-9181-0
- THIRD GENERATION WIRELESS INFORMATION NETWORKS**, Sanjiv Nanda, David J.  
Goodman  
ISBN: 0-7923-9128-3
- SOURCE AND CHANNEL CODING: An Algorithmic Approach**, John B. Anderson, Seshadri  
Mohan  
ISBN: 0-7923-9210-8
- ADVANCES IN SPEECH CODING**, Bishnu Atal, Vladimir Cuperman, Allen Gersho  
ISBN: 0-7923-9091-1
- SWITCHING AND TRAFFIC THEORY FOR INTEGRATED BROADBAND NETWORKS**,  
Joseph Y. Hui  
ISBN: 0-7923-9061-X
- ADAPTIVE DATA COMPRESSION**, Ross N. Williams  
ISBN: 0-7923-9085
- SOURCE CODING THEORY**, Robert M. Gray  
ISBN: 0-7923-9048-2

---

---

**COMMUNICATIONS AND  
CRYPTOGRAPHY**  
*Two Sides of One Tapestry*

*edited by*

**Richard E. Blahut**  
*University of Illinois*

**Daniel J. Costello, Jr.**  
*University of Notre Dame*

**Ueli Maurer**  
*ETH Zurich*

**Thomas Mittelholzer**  
*University of California, San Diego*



SPRINGER SCIENCE+BUSINESS MEDIA, LLC

ISBN 978-1-4613-6159-6      ISBN 978-1-4615-2694-0 (eBook)  
DOI 10.1007/978-1-4615-2694-0

---

**Library of Congress Cataloging-in-Publication Data**

A C.I.P. Catalogue record for this book is available  
from the Library of Congress.

---

**Copyright** © 1994 by Springer Science+Business Media New York  
Originally published by Kluwer Academic Publishers, New York in 1994  
Softcover reprint of the hardcover 1st edition 1994

All rights reserved. No part of this publication may be reproduced, stored in  
a retrieval system or transmitted in any form or by any means, mechanical,  
photo-copying, recording, or otherwise, without the prior written permission of  
the publisher, Springer Science+Business Media, LLC.

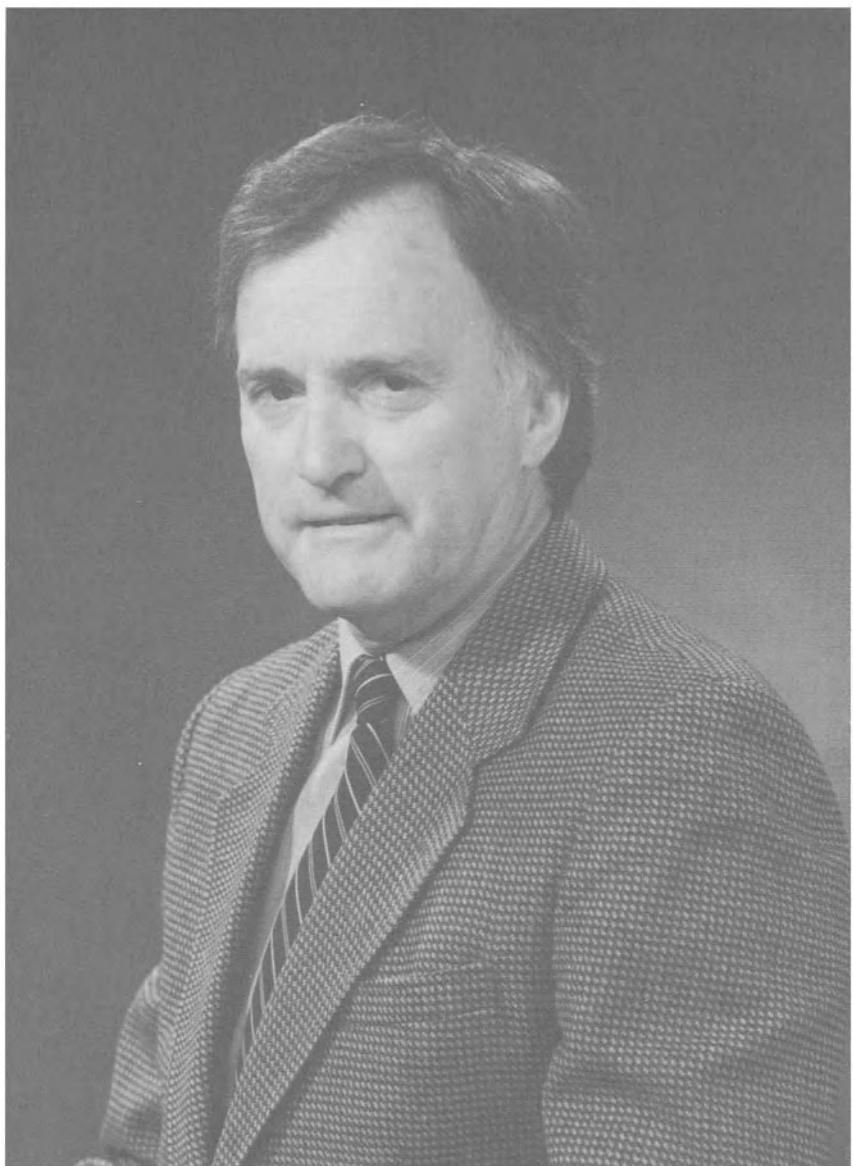
---

*Printed on acid-free paper.*

---

**Manuuscript Coordination and Production**

John L. Ott  
Clarice Staunton  
*University of Notre Dame*



**James L. Massey**

**Talks presented at the Symposium on  
“Communications, Coding, and Cryptography”  
in honor of James L. Massey on the occasion of  
his 60th birthday Centro Stefano Franscini,  
Ascona, Switzerland, February 10-14, 1994**

**Block Coding**

- **On a Problem of Persi Diaconis**  
E. Berlekamp
- **A Finite Fourier Transform for Vectors of Arbitrary Length**  
C.G. Günther
- **Massey’s Theorem and the Golay Codes**  
R.E. Blahut
- **Spherical Codes from the Hexagonal Lattice**  
T. Ericson and V. Zinoviev
- **On Group Codes Generated by Finite Reflection Groups**  
T. Mittelholzer
- **Using Redundancy to Speed up Disk Arrays**  
D.L. Cohn and R.L. Stevenson
- **A Comparison of Error Patterns Corrected by Block Codes and Convolutional Codes**  
J. Justesen
- **Coded MPSK Modulation for the AWGN and Rayleigh Fading Channels**  
S. Lin, S. Rajpal, and D.J. Rhee
- **On the Basic Averaging Arguments for Linear Codes**  
H.A. Loeliger
- **Coding and Multiplexing**  
H.J. Matt

## **Convolutional Coding**

- **Duality of Linear Input-Output Maps**  
S.K. Mitter
- **Inverses of Linear Sequential Circuits: On Beyond Poles and Zeros**  
M.K. Sain
- **Trellises Old and New**  
G.D. Forney Jr.
- **On Canonical Encoding Matrices and the Generalized Constraint Lengths of Convolutional Codes**  
R. Johannesson and Z. Wan
- **On Code Linearity and Rotational Invariance for a Class of Trellis Codes for M-PSK**  
L.H. Zetterberg
- **Progress Towards Achieving Channel Capacity**  
D.J. Costello and L. Perez
- **Soft is Better than Hard**  
J. Hagenauer
- **Charge Constrained Convolutional Codes**  
M.A. Herro, R.H. Deng, and Y.X. Li
- **Five Views of Differential MSK: A Unified Approach**  
B. Rimoldi
- **Binary Convolutional Codes Revisited**  
G. Ungerboeck

## **Cryptography**

- **Development of Fast Multiplier Structures with Cryptographic Applications**  
G. Agnew
- **On Repeated-Single-Root Constacyclic Codes**  
V.C. da Rocha, Jr.
- **Delay Estimation for Truly Random Binary Sequences or How to Measure the Length of Rip van Winkle's Sleep**  
I. Ingemarsson

- Low Weight Parity Checks for Linear Recurring Sequences  
G. Kuehn
- Higher Order Derivatives and Differential Cryptanalysis  
X. Lai
- The Strong Secret Key Rate of Discrete Random Triples  
U.M. Maurer
- International Commercial Standards in Cryptography  
J. Omura
- The Self-Shrinking Generator  
W. Meier and O. Staffelbach
- Models for Adder Channels  
I. Bar-David
- Coding for Adder Channels  
I.F. Blake

## Information Theory

- Orthogonal Checksets in the Plane and Enumerations of the Rationals mod  $p$   
P. Elias
- An Inequality on the Capacity Region of Multiaccess Multipath Channels  
R.G. Gallager
- On the Performance of Aperiodic Inverse Filter  
J. Ruprecht
- Capacity of a Simple Stable Protocol for Short Message Service over a CDMA Network  
A.J. Viterbi
- Random Time and Frequency Hopping for Infinite User Population  
S. Csibi
- Multiple Access Collision Channel Without Feedback and  $\infty$  User Population  
L. Györfi
- Messy Broadcasting in Networks  
R. Ahlswede, H.S. Haroutunian, and L.H. Khachatrian

- **Stochastic Events**

H. Ohnsorge

- **Leaf-Average Node-Sum Interchanges in Rooted Trees with Applications**

R.A. Rueppel and J.L. Massey

- **Some Reflections on the Interference Channel**

E.C. van der Meulen

- **The Sliding-Window Lempel-Ziv Algorithm is Asymptotically Optimal**

A.D. Wyner and J. Ziv

# Contents

|  |            |
|--|------------|
| <b>Talks presented at the Symposium</b>  | <b>vii</b> |
| <b>Foreword</b>  | <b>xv</b>  |
| <b>1 Development of Fast Multiplier Structures with Cryptographic Applications</b> | <b>1</b>   |
| G. Agnew   |            |
| <b>2 Messy Broadcasting in Networks</b>  | <b>13</b>  |
| R. Ahlsweide, H.S. Haroutunian, and L.H. Khachatrian                               |            |
| <b>3 On a Problem of Persi Diaconis</b>  | <b>25</b>  |
| E. Berlekamp   |            |
| <b>4 Aspects of Linear Complexity</b>  | <b>35</b>  |
| S. Blackburn, G. Carter, D. Gollmann, S. Murphy, K. Paterson, F. Piper and P. Wild |            |
| <b>5 Massey's Theorem and the Golay Codes</b>                                      | <b>43</b>  |
| R.E. Blahut  |            |
| <b>6 Coding for Adder Channels</b>   | <b>43</b>  |
| I.F. Blake   |            |
| <b>7 Using Redundancy to Speed up Disk Arrays</b>                                  | <b>59</b>  |
| D.L. Cohn and R.L. Stevenson   |            |
| <b>8 Progress Towards Achieving Channel Capacity</b>                               | <b>69</b>  |
| D.J. Costello, Jr. and L. Perez  |            |
| <b>9 Random Time and Frequency Hopping for Infinite User Population</b>            | <b>81</b>  |
| S. Csibi and L. Györfi   |            |
| <b>10 On Repeated-Single-Root Constacyclic Codes</b>                               | <b>93</b>  |
| V.C. da Rocha, Jr.   |            |

|           |  |            |
|-----------|--|------------|
| <b>11</b> | <b>Orthogonal Checksets in the Plane and Enumerations of the Ratios mod <math>p</math></b>                       | <b>101</b> |
|           | P. Elias   |            |
| <b>12</b> | <b>Spherical Codes from the Hexagonal Lattice</b>  | <b>109</b> |
|           | T. Ericson and V. Zinoviev   |            |
| <b>13</b> | <b>Trellises Old and New</b>   | <b>115</b> |
|           | G.D. Forney, Jr.   |            |
| <b>14</b> | <b>An Inequality on the Capacity Region of Multiaccess Multipath Channels</b>                                    | <b>129</b> |
|           | R.G. Gallager  |            |
| <b>15</b> | <b>A Finite Field Fourier Transform for Vectors of Arbitrary Length</b>  | <b>141</b> |
|           | C.G. Günther   |            |
| <b>16</b> | <b>Soft is Better than Hard</b>  | <b>155</b> |
|           | J. Hagenauer   |            |
| <b>17</b> | <b>Charge Constrained Convolutional Codes</b>  | <b>173</b> |
|           | M.A. Herro, R.H. Deng, and Y.X. Li   |            |
| <b>18</b> | <b>Delay Estimation for Truly Random Binary Sequences or How to Measure the Length of Rip van Winkle's Sleep</b> | <b>179</b> |
|           | I. Ingemarsson   |            |
| <b>19</b> | <b>On Canonical Encoding Matrices and the Generalized Constraint Lengths of Convolutional Codes</b>              | <b>187</b> |
|           | R. Johannesson and Z. Wan  |            |
| <b>20</b> | <b>A Comparison of Error Patterns Corrected by Block Codes and Convolutional Codes</b>                           | <b>201</b> |
|           | J. Justesen  |            |
| <b>21</b> | <b>Encounters with the Berlekamp-Massey Algorithm</b>  | <b>209</b> |
|           | T. Kailath   |            |
| <b>22</b> | <b>Using Zech's Logarithm to Find Low-Weight Parity Checks for Linear Recurring Sequences</b>                    | <b>221</b> |
|           | G.J. Kuhn and W.T. Penzhorn  |            |
| <b>23</b> | <b>Higher Order Derivatives and Differential Cryptanalysis</b>   | <b>227</b> |
|           | X. Lai   |            |

|           |  |            |
|-----------|--|------------|
| <b>24</b> | <b>Coded MPSK Modulation for the AWGN and Rayleigh Fading Channels</b>                 | <b>235</b> |
|           | S. Lin, S. Rajpal, and D.J. Rhee   |            |
| <b>25</b> | <b>On the Basic Averaging Arguments for Linear Codes</b>                               | <b>251</b> |
|           | H.A. Loeliger  |            |
| <b>26</b> | <b>Coding and Multiplexing</b>   | <b>263</b> |
|           | H.J. Matt  |            |
| <b>27</b> | <b>The Strong Secret Key Rate of Discrete Random Triples</b>                           | <b>271</b> |
|           | U.M. Maurer  |            |
| <b>28</b> | <b>The Self-Shrinking Generator</b>  | <b>287</b> |
|           | W. Meier and O. Staffelbach  |            |
| <b>29</b> | <b>Constructions and Decoding of Optimal Group Codes from Finite Reflection Groups</b> | <b>297</b> |
|           | T. Mittelholzer  |            |
| <b>30</b> | <b>Duality of Linear Input-Output Maps</b>   | <b>307</b> |
|           | S.K. Mitter  |            |
| <b>31</b> | <b>Cut-Off Rate Channel Design</b>   | <b>315</b> |
|           | P. Narayan and D.L. Snyder   |            |
| <b>32</b> | <b>Construction of Invertible Sequences for Multipath Estimation</b>                   | <b>323</b> |
|           | A.M. Odlyzko   |            |
| <b>33</b> | <b>Five Views of Differential MSK: A Unified Approach</b>                              | <b>333</b> |
|           | B. Rimoldi   |            |
| <b>34</b> | <b>Leaf-Average Node-Sum Interchanges in Rooted Trees with Applications</b>            | <b>343</b> |
|           | R.A. Rueppel and J.L. Massey   |            |
| <b>35</b> | <b>On the Performance of Aperiodic Inverse Filter Approximations</b>                   | <b>357</b> |
|           | J. Ruprecht  |            |
| <b>36</b> | <b>Inverses of Linear Sequential Circuits: On Beyond Poles and Zeros</b>               | <b>367</b> |
|           | M.K. Sain  |            |
| <b>37</b> | <b>Binary Sequences with Small Correlations</b>  | <b>381</b> |
|           | G. Seguin and G. Drolet  |            |
| <b>38</b> | <b>Fast Bounded-Distance Decoding of the Nordstrom-Robinson Code</b>                   | <b>391</b> |
|           | F.W. Sun and H.C. van Tilborg  |            |

|           |   |            |
|-----------|---|------------|
| <b>39</b> | <b>Binary Convolutional Codes Revisited</b>   | <b>399</b> |
|           | G. Ungerboeck   |            |
| <b>40</b> | <b>Some Reflections on the Interference Channel</b>   | <b>409</b> |
|           | E.C. van der Meulen   |            |
| <b>41</b> | <b>Capacity of a Simple Stable Protocol for Short Message Service over<br/>a CDMA Network</b> | <b>423</b> |
|           | A.J. Viterbi  |            |
| <b>42</b> | <b>The Sliding-Window Lempel-Ziv Algorithm is Asymptotically<br/>Optimal</b>                  | <b>431</b> |
|           | A.D. Wyner and J. Ziv   |            |
| <b>43</b> | <b>On Code Linearity and Rotational Invariance for a Class of Trellis<br/>Codes for M-PSK</b> | <b>439</b> |
|           | L.H. Zetterberg   |            |
| <b>44</b> | <b>Algebraic-Sequential Decoding - Ideas and Results</b>                                      | <b>451</b> |
|           | K.S. Zigangirov   |            |
| <b>45</b> | <b>Index</b>  | <b>461</b> |
| <b>46</b> | <b>Publications by James Massey</b>   | <b>469</b> |

## Foreword

Information theory is an exceptional field in many ways. Technically, it is one of the rare fields in which mathematical results and insights have led directly to significant engineering payoffs. Professionally, it is a field that has sustained a remarkable degree of community, collegiality, and high standards. In this book we celebrate James L. Massey's life and career because they have been so central to the evolution of all these aspects of our field.

Jim came out of a classically American background: childhood in the great Midwestern state of Ohio (1934-52), college at the pre-eminent Catholic University of Notre Dame (1952-56), and service as a communications officer in the U.S. Marine Corps (1956-59). Those who know him well know that the high values of these rugged and demanding institutions, though never accepted unquestioningly, remain lodged deep in his soul.

Jim then obtained his Ph.D. degree (1959-62) in the information theory group at M.I.T., during what has been called its golden age. The objective was clear: find practical ways of realizing some of the gains predicted by Shannon's theory. The task engaged M.I.T.'s brightest faculty and students, many of whom have become renowned in the annals of our field. In this collective enterprise, Jim's thesis on threshold decoding of Elias' convolutional codes was regarded as a decisive breakthrough. It led directly to the founding of Codex Corporation in 1962, where Jim actively consulted throughout the 1960s. His resulting monograph (*Threshold Decoding*, MIT Press, 1963) was one of only two books ever to receive the Prize Paper Award of the IEEE Information Theory Society (1964).

Upon graduation, Jim returned to his beloved Notre Dame and began his remarkable academic career. An inspiring teacher, world-class researcher, and highly respected citizen of his community, he was named the first chaired professor in the history of Notre Dame in 1972. He attracted a notable series of bright and productive graduate students to our field, many of whom have been instrumental in the propagation of information theory teaching and research.

Jim's research, lectures, and personality rapidly made him one of the leading figures in our field. Key research contributions during this time included pioneering the algebraic structure theory of convolutional codes, by regarding convolutional encoders as linear systems, and independent development of what is now known as the Berlekamp-Massey algorithm for decoding of BCH codes, again via an illuminating connection with linear system theory.

Jim went through a difficult period in the mid 1970s that led to his decision to leave Notre Dame in 1977. After three years at MIT (1977-78) and UCLA (1978-80), he was invited to become a Professor at ETH-Zurich in 1980, where his tenure has been as remarkable as his earlier one at Notre Dame. From this prestigious position, he has arguably influenced communications research in Europe in recent years more than any other individual. The success of his infectious open manner seems also to have significantly affected the style of European academic research.

Jim's research interests broadened at ETH to include cryptography and multiaccess communications. Joint work with Jim Omura on application of Galois fields to cryptography led to the founding of Cylink Corporation (1984). Jim was founding president in 1986 of the International Association for Cryptologic Research. His annual short course on cryptography is usually oversubscribed, and his insistence that the ultimate cryptosystems

must be provably secure from a mathematical point of view has had broad impact.

Jim's work on contention resolution protocols for multiaccess communications has been widely recognized. A paper with his student, Peter Mathys, on "The Collision Channel with Feedback" won the 1987 IEEE W.R.G. Baker Award (best paper in any IEEE publication). He continues to insist that good protocols should be provably deadlock-free and should preferably be developed by the methods of information theory rather than by those of computer science.

Jim is an exemplary engineering teacher and scholar. His lectures are in wide demand for their clarity, insightfulness, and wit, and have markedly influenced the development of the field. While he appreciates sharp mathematical tools and has made outstanding contributions to theory, he disdains unnecessary mathematics, and is always looking to unearth and express fundamental concepts in the most intuitive and insightful way. He insists that engineering research must be relevant to practical applications. His advocacy of statistical methods (such as maximum likelihood) in the design of communication systems and his proposal that the "cut-off rate" is a more practically meaningful measure of channel performance than Shannon capacity have been widely adopted by a generation of communications engineers.

As a reviewer, conference organizer, board member, Transactions editor (1975-78), and President (1969) of the IEEE Information Theory Society, Jim has insisted on the highest standards, while maintaining a spirit of participation in an exciting shared enterprise. His reviews, whether private or public, are witty, sharp, and blunt. He always takes the time not only to uncover the flaws but also to praise the merits of a paper, often appreciating them better than the author.

Jim has received many honors, including membership in the U.S. National Academy of Engineering, the Swiss Academy of Engineering Sciences, and the European Academy of Arts and Sciences. In 1988, he was named the Shannon Lecturer for the IEEE Symposium on Information Theory and in 1992 received the IEEE Alexander Graham Bell Medal. The papers in this volume testify not only to the impact of his scholarship, but also to the respect and affection in which he is held around the world.

---

---

**COMMUNICATIONS AND**  
**CRYPTOGRAPHY**  
*Two Sides of One Tapestry*

# Development of Fast Multiplier Structures with Cryptographic Applications

G.B. Agnew

University of Waterloo  
Waterloo, Ontario, Canada N2L 3G1

## Abstract

The status of the development and application of high-speed multiplier structures based on optimal normal bases is explored. These structures are significant in that they can be used to construct multipliers for fields large enough for cryptographic systems based on the discrete logarithm problem. Their characteristics also make them ideal candidates for implementing compact, high-speed elliptic curve cryptosystems.

## I Introduction

In 1976 Diffie and Hellman [1] introduced the notion of cryptographic systems based on an asymmetric key structure (Public Key Cryptography). While not providing a method of sending messages, they did suggest a key exchange system based on the difficulty of finding logarithms over large prime fields. It was several years until implementations of message exchange systems were actually constructed. The first was the system developed by Rivest, Shamir, and Adleman [2] based on the difficulty of factoring the product of two large primes. Many other variants based on the knapsack problem and error correcting codes soon followed. An interesting variant was the Massey-Omura lock [3] which incorporated a three-pass message exchange system based on the difficulty of taking logarithms over fields of characteristic two.

All of these systems rely on the notion of performing operations (specifically, multiplication and modulo reduction) over “large” algebraic systems. At the time when these systems were introduced, the design of devices to perform such operations was not extensively developed. An attractive feature of systems such as Diffie-Hellman key exchange, Omura-Massey, Massey-Omura, and later ElGamal [4], was that the field could be fixed and used by all parties. Thus, hardware to perform multiplication and modulo reduction could be optimized for that field. Fields of characteristic two were particularly attractive because addition operations involved simple XOR operations.

## II Massey-Omura Multiplier

In [3], Massey develops a multiplier structure based on the normal basis representation of the field. To show this, let  $\beta$  be a generator of the normal basis  $N$  in the extension field  $GF(2^n)$ . The normal basis can be represented as:

$$N = \{\beta^{2^0}, \beta^{2^1}, \beta^{2^2}, \dots, \beta^{2^{n-1}}\}$$

with

$$\beta^{2^n} = \beta$$

Let two elements be represented as

$$A = \sum_{i=0}^{n-1} a_i \beta^{2^i}$$

and

$$B = \sum_{i=0}^{n-1} b_i \beta^{2^i}$$

for

$$a_i, b_i \in \{0, 1\}$$

We can write

$$\underline{A} = \{a_0, a_1, a_2, \dots, a_{n-1}\}$$

Let's look at squaring a field element,

$$A^2 = \left[ \sum_{i=0}^{n-1} a_i \beta^{2^i} \right] \left[ \sum_{i=0}^{n-1} a_i \beta^{2^i} \right].$$

The cross-product terms of the form  $a_i * a_j$   $i \neq j$ , will all reduce to zero since coefficients are calculated modulo 2. The only terms that will remain are of the form

$$a_i \beta^{2^i} * a_i \beta^{2^i} = a_i \beta^{2^{i+1}}$$

Thus

$$\underline{A}^2 = \{a_1, a_2, \dots, a_{n-1}, a_0\}$$

i.e., a cyclic shift of A. In hardware, this can be performed in one clock cycle.

To form the product  $C = AB$

$$\begin{aligned} C &= \left[ \sum_{i=0}^{n-1} a_i \beta^{2^i} \right] \left[ \sum_{i=0}^{n-1} b_i \beta^{2^i} \right] \\ &= \sum_{0 \leq i, j \leq n-1} a_i b_j (\beta^{2^i} \beta^{2^j}) \end{aligned}$$

Let

$$\beta^{2^i} \beta^{2^j} = \sum_{k=0}^{n-1} \sigma_{ij}^k \beta^{2^k}$$

where

$$\sigma_{ij} \in \{0, 1\}$$

The values  $\sigma_{ij}$  are determined by the basis. We now write

$$C = \sum_{k=0}^{n-1} c_k \beta^{2^k}$$

and

$$c_k = \sum_{0 \leq i,j \leq n-1} \sigma_{ij}^k a_i b_j.$$

We observe that

$$\begin{aligned} C^2 &= A^2 B^2 = \left( \sum_{i=0}^{n-1} a_{i-1} \beta^{2^i} \right) \left( \sum_{j=0}^{n-1} b_{j-1} \beta^{2^j} \right) \\ &= \sum_{i,j} a_{i-1} b_{j-1} (\beta^{2^i} \beta^{2^j}) \end{aligned}$$

and

$$C^2 = \sum_{i=0}^{n-1} c_{k-1} \beta^{2^k}$$

Thus

$$c_{k-1} = \sum_{i,j} \sigma_{ij}^k a_{i-1} b_{j-1}$$

and

$$c_{k-h} = \sum_{i,j} \sigma_{ij}^k a_{i-h} b_{j-h}$$

This gives a general expression

$$c_k = P_k(\underline{A}, \underline{B}) = \sum_{i,j} \sigma_{ij}^k a_i b_j$$

and

$$P_k(\underline{A}^{2^i}, \underline{B}^{2^i}) = c_{k-i}$$

Let the logic function for calculation of the first bit of  $C$  be

$$\begin{aligned} \lambda_{ij} &= \sigma_{ij}^0 \\ c_0 &= \sum_{ij} \lambda_{ij} a_i b_j \end{aligned}$$

then

$$c_k = \sum_{ij} \lambda_{ij} a_{i+k} b_{j+k}.$$

This means that, if we apply a fixed logic function to successive rotations of  $A$  and  $B$ , we can realize the normal basis multiplier. This is the principle of the Massey-Omura multiplier. In particular, if a circuit is constructed to produce  $c_i$ , then all of the other components of  $C$ ,  $c_j$   $j \neq i$  can be formed with simple shifts of the elements of  $A$  and  $B$ . Thus, a field multiplication can be completed in  $n$  clock cycles with one bit of  $C$  being computed on each cycle. This is shown for a small field in Figure 1. This has the advantage of having a relatively regular structure but suffers from the fact that the complexity of the circuit increases with the square of the number of bits in the field (approximately  $n^2/2$ ). The circuit is impractical to build, using current technology, for fields suitable for cryptographic systems.

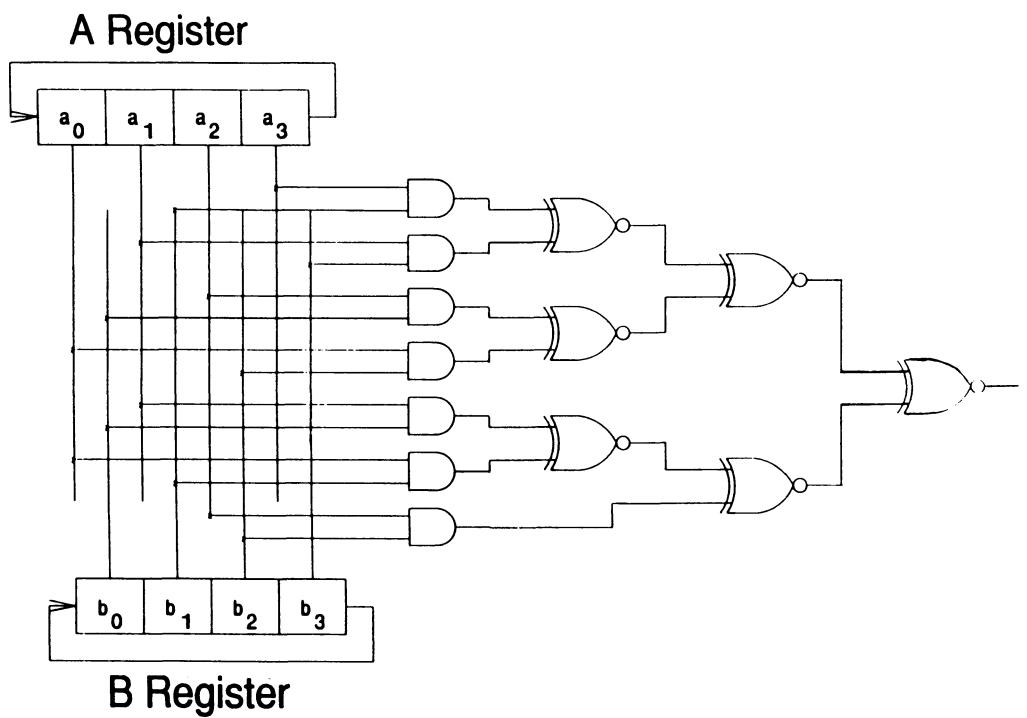


Figure 1 - Massey-Omura Multiplier in GF( $2^4$ )

### III Optimal Normal Basis Multipliers

In [5] Mullin, Vanstone, et. al, observed that certain extension fields of characteristic two have optimal normal bases forms associated with them. They show that the number of nonzero terms in the bilinear form,  $C(N)$ , of  $c_0$  is given as

$$C(N) \geq 2n - 1.$$

If  $C(N) = 2n - 1$ , then the basis  $N$  is referred to as an *optimal normal basis*.

From before, we see that

$$c_k = \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} \lambda_{ij}^{(k)} a_i b_j.$$

This can be rewritten as

$$c_k = \sum_{j=0}^{n-1} b_j \sum_{i=0}^{n-1} \lambda_{ij}^{(k)} a_i.$$

Using the cyclic relation amongst the equations, this equation can also be written as

$$c_k = \sum_{j=0}^{n-1} b_{j+k} \sum_{i=0}^{n-1} \lambda_{ij}^{(0)} a_{i+k},$$

where subscripts are to be reduced modulo  $n$ .

Let  $A_1, A_2, \dots, A_n$  and  $B_1, B_2, \dots, B_n$  be individual cells of cyclic shift registers **A** and **B** respectively. Define logic cells  $C_i, i = 0, 1, 2, \dots, n - 1$ , as follows.

In cell  $C_{k,j}$ , let there be a logical circuit that will compute the expression

$$T_{k,j}(t) = \overline{B}_{j+k}(t) \sum_{i=0}^{n-1} \lambda_{ij}^{(0)} \overline{A}_{i+k}(t)$$

where  $\overline{A}_p(t)$  and  $\overline{B}_q(t)$  are the contents of cells  $A_p$  and  $B_q$  of **A** and **B** respectively at time  $t$ . This cell also contains a storage register  $R_k$  that can store previously calculated results and can add its contents  $\overline{R}_k$  to the value of  $T_k$  calculated above. (Here multiplication represents the logical operation “AND” and addition represents the logical operation “XOR”.) To accomplish this physically, cell  $C_k$  must be connected to the cell  $B_{j+k}$  of **B** and to  $w_j$  cells of **A** where  $w_j$  is the number of nonzero coefficients in  $\{\lambda_{ij}^{(0)} : i = 0, 1, \dots, n - 1\}$ .

The network is considered to operate as follows. The system is initialized by loading the values  $a_i$  and  $b_j$  in the respective cells of  $A_i$  and  $B_j$  respectively, and the registers  $R_i$  of cells  $C_i$  are loaded with zero for  $i = 0, 1, \dots, n - 1$ . At time  $t$ , for  $t = 0, 1, \dots, n - 1$ , the term  $T_k(t)$  is calculated in cell  $C_k(t)$  using the current contents of the **A** and **B** registers. The current contents of  $R_k$  are XOR’ed with  $T_k(t)$  and the results are stored in register  $R_{k+1 \bmod n}$  (see Figure 2).

The logic is clocked  $n$  times ( $t = n - 1$ ) and, at end of the period, the register  $R_k$  contains  $c_k, k = 0, 1, 2, \dots, n - 1$ . For example, consider the contents of register  $R_0$  at the end of time  $n - 1$ . By the functioning of the network, this register contains.

$$\sum_{t=0}^{n-1} T_t(t)$$

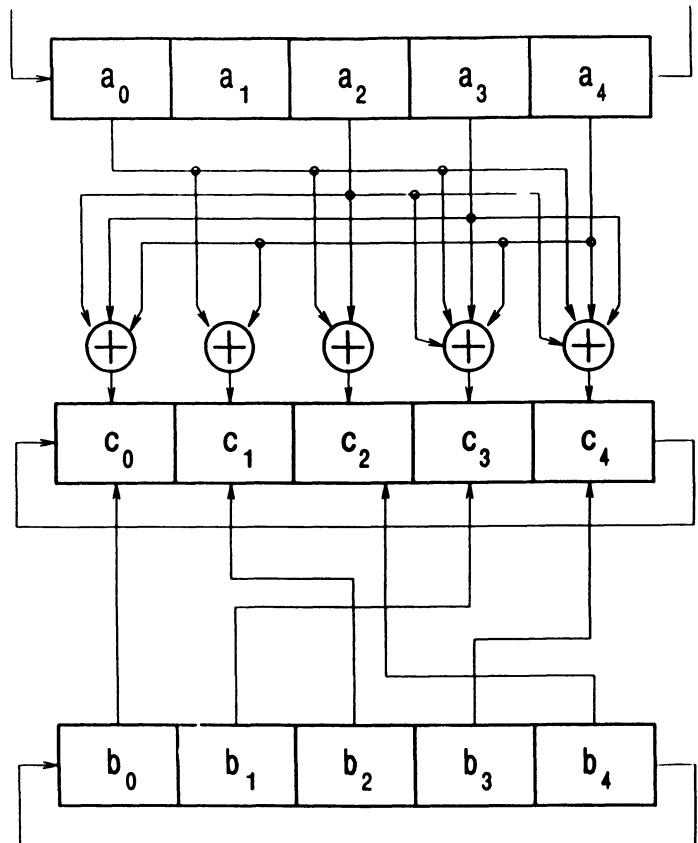


Figure 2 - Register Layout of Optimal Normal Basis Multiplier in  $\text{GF}(2^5)$

But

$$\begin{aligned}
\sum_{t=0}^{n-1} T_t(t) &= \sum_{t=0}^{n-1} \bar{B}_{j(t)+k_{j(t)}}(t) \sum_{i=1}^{n-1} \lambda_{i,j(t)}^{(0)} \bar{A}_{i+k_{j(t)}}(t) \\
&= \sum_{t=0}^{n-1} b_{j(t)} \sum_{i=0}^{n-1} \lambda_{i,j(t)}^{(0)} a_i \\
&= \sum_{t=0}^{n-1} b_t \sum_{i=0}^{n-1} \lambda_{it}^{(0)} a_i = c_0
\end{aligned}$$

Similarly, register  $R_s$  contains

$$\sum_{t=0}^{n-1} T_{s+t}(t)$$

where  $s + t$  is to be reduced modulo  $n$ . It should be noticed from the above form that, in contrast to the Massey-Omura multiplier, all of the bits of  $C$  are simultaneously available after  $n$  clock cycles.

Much of the complexity of implementing these structures arises from the interconnection between the **A** register and the **C** register containing the cells  $C_i$ . The complexity can be reduced by appropriate choice of the basis, using an optimal normal basis when possible. In such cases, the number of connections required is exactly  $2n - 1$ . The implication of this discovery was that a structure capable of performing extension field multiplication in  $n$  clock cycles for  $n > 1000$  could be fabricated using the VLSI technology available in the mid 1980s (see [6, 7, 8]).

In addition to the extension fields of characteristic two, the size of the underlying field can be extended using techniques such as the quadratic, cubic, quartic, etc., extensions of the field. This is useful in improving the security of the system given a multiplier structure that is optimized for a particular field size. For example, let the quadratic extension of  $A$  and  $B$  be  $(x_1, y_1)$  and  $(x_2, y_2)$  respectively. Let  $C$  be represented as  $(x_3, y_3)$ . Then  $C = A * B$  is calculated as

$$x_3 = y_1 y_2 + x_1 y_2 + x_2 y_1$$

and

$$y_3 = x_1 x_2 + x_1 y_2 + x_2 y_1$$

This represents a “cost” of six extra field multiplications.

## IV Elliptic Curve Systems

In 1985, Koblitz [9] and Miller [10] independently put forth the notion of using elliptic curves as the basis of a public key cryptosystem. In these systems, a public key is formed by multiplying a fixed point on the curve  $P$ , by a field element  $k$ . The difficulty facing the attacker is the following, given

$$Q = kP$$

find  $k$  (this is known as the elliptic logarithm problem).

In general, there are two forms of curves useful for implementing cryptographic systems: super-singular and non-super-singular curves. It was originally thought that equivalent

levels of security were attainable in either form of curve. The advantage of these systems is that there is no (known) subexponential method of computing logarithms for elliptic curves. Thus, an elliptic curve system using a base field of only 120 bits would be very secure. The advantage of using super-singular curves, is that point multiplication on the curve involves relatively simple operations in the base field (multiplication and field addition). Later, Menezes, et. al. [11] discovered that the super-singular elliptic logarithm problem can be reduced to finding logarithms in an extension of the base field thus reducing the security for a particular block size.

Non-super-singular curves are not vulnerable to the attacks applied to super-singular curves. The disadvantage is the complexity of computation required to perform point multiplication. We will consider only curves of characteristic two. The curves we are interested in are of the form:

$$y^2 + xy = x^3 + ax + b$$

Operations on the curve are defined in the following way. For points  $P = (x_1, y_1)$  and  $Q = (x_2, y_2)$ , and

$$-P = (x_1, y_1 + x_1)$$

we have

$$\begin{aligned} x_3 &= \begin{cases} \left(\frac{y_1+y_2}{x_1+x_2}\right)^2 + \left(\frac{x_1+y_2}{x_1+x_2}\right) + x_1 + a & P \neq Q \\ \frac{b}{x_1^2} + x_1^2 & P = Q \end{cases} \\ y_3 &= \begin{cases} \left(\frac{y_1+y_2}{x_1+x_2}\right)(x_1 + x_3) + x_3 + y_1 & P \neq Q \\ x_1^2 + \left(x_1 + \frac{y_1}{x_1}\right)x_3 + x_3 & P = Q \end{cases} \end{aligned}$$

Another method of computing points on the curve involves projective coordinates. The projective equation for the non-super-singular elliptic curve is written as:

$$zy^2 + zxy = x^3 + zx^2a + bz^3.$$

We set  $Q = (x_1, y_1, z_1)$ ,  $P = (x_2, y_2, 1)$  and define  $P + Q = (x_3'', y_3'', z_3'')$ . For the case  $P \neq Q$  (addition operation), let

$$\begin{aligned} x_3' &= z_1^2 \{ z_1(b + y_2^2 + x_2(y_2 + x_2^2)) + x_2^2a \} \\ &\quad + (x_2y_1 + x_1(y_2 + x_2^2)) \} + z_1(x_1^2x_2). \end{aligned}$$

Then

$$\begin{aligned} x_3'' &= (x_1 + x_2z_1)x_3' \\ y_3'' &= (x_1 + x_2z_1)[(y_1 + y_2z_1) + y_1(x_1 + x_2z_1)] \\ &\quad + [(y_1 + y_2z_1) + (x_1 + x_2z_1)] \\ z_3'' &= z_1(x_1 + x_2z_1)^3 \end{aligned}$$

For the case  $P = Q$  (doubling operation), let

$$x_3' = x_1^4 + z_1^4b$$

Then

$$\begin{aligned}x_3'' &= (x_1 z_1) x_3' \\y_3'' &= (x_1^2 + y_1 z_1) x_3' + (x_1^4 + x_3') x_1 z_1 \\z_3'' &= (x_1 z_1)^3\end{aligned}$$

The advantage of using this method is that only one inverse operation is required at the end of the calculation to divide out the  $z$  coordinate.

## V A Hardware Implementation

The original work on optimal normal basis multipliers was applied to the elliptic curve problem. Our objective not only was to produce a fast and efficient system but also was to produce a system that would occupy a minimum of area in a VLSI implementation [12, 13, 14]. Our choice was to implement an optimal normal basis multiplier for  $GF(2^{155})$  in a custom gate array. The gate array device uses three registers to implement the multiplier structure and interconnection, a controller to implement the elementary operations (such as shifts, additions, and multiplications) as well as incorporating a very fast 32-bit wide input/output structure. The device was fabricated using a 1.5 micron HCMOS gate array with a clock speed of 40 MHz. and required less than 12,000 gates. For the primary operations, the speed can be calculated as in the following table.

| OPERATION  | SIZE   | CLOCK CYCLES |
|--|--|--------------|
| Multiplication                                   | 155 bit blocks   | 156          |
| Calculation of Inverse                           | 24 multiplications   | approx. 3800 |
| I/O  | 5 - 32 bit transfers per read/write to registers 2 clock cycles per transfer | 10           |
| Addition (XOR) and elementary register operation | 155 bit parallel operation   | 2            |

## VI Throughput Calculations

If we consider point multiplication by an integer with Hamming weight 30, this will require about 154 point doublings and 29 point multiplications. Using projective coordinates for a non-super-singular curve, doubling requires six multiplications and point addition requires thirteen multiplications. At the end of the computation, a single inverse operation followed by two multiplications must be performed to return to affine coordinates. Allowing for I/O overhead in the doubling and multiplication routines, the device will be able to perform at least 145 integer multiplications of a point per second. For use in an encryption system, each of  $X$  and  $Y$  coordinates can be used, so 310 bits can be sent per point calculation for a throughput of approximately 50 Kbps.

We note that a significant portion of the time is spent in doing the point doublings. In elliptic curve systems, the same base point  $P$  can be used repeatedly. If this is the case, then all of the squares can be precomputed which will increase the throughput by a factor of 4 to approximately 200 Kbps. The storage requirements for the point squarings is less than 6 Kbytes, a relatively small amount.

The most significant feature of this implementation is the relatively low complexity of the multiplier core, which requires less than 12,000 transistors. Using current layout and fabrication techniques, the elliptic curve processor core would occupy less than 4% of the area available on current Smart Card devices. This would make the system, we believe, the first practical public key system that could be incorporated in a Smart Card based system.

## VII Summary

We have explored the development and application of high speed multiplier structures based on optimal normal bases. These structures are significant in that they can be used to construct multipliers for fields large enough for cryptographic systems based on the discrete logarithm problem. The characteristics that make them useful for such large fields also make them ideal candidates for implementing compact, high speed elliptic curve cryptosystems (in a Smart Card for example).

## References

- [1] W. Diffie, and M. Hellman, “New directions in cryptography”, IEEE Trans. on Info. Theory, vol. IT-22, pp.644-654, 1976.
- [2] R. L. Rivest, A. Shamir, and L. Adleman, “A method of obtaining digital signatures and public key cryptosystems”, Comm. ACM, Vol.21, pp. 120-126, 1978.
- [3] P. K. S. Wah and M.Z. Wang, “Realization and application of the Massey-Omura lock”, Proceeding of 1984 International Zurich Seminar of Digital Communication, pp. j2.1-2.8.
- [4] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Trans. on Info. Theory, Vol. IT-31, pp. 469-472, 1985.
- [5] R. C. Mullin, I.M. Onyszchuk, S.A. Vanstone, and R.M. Wilson, “Optimal normal bases in  $GF(p^n)$ ”, Discrete Applied Mathematics, Vol. 22, pp. 149-161, 1988-89.
- [6] CA34C168 Data Encryption Processor Data Sheet, Newbridge Microsystems, Kanata, Ontario, Canada.
- [7] G. Agnew, T. Beth, R. Mullin, and S. Vanstone, “Arithmetic operations in  $GF(2^n)$ ”, Journal of Cryptology .
- [8] G. Agnew, R. Mullin, and S. Vanstone, “An implementation of a fast public key cryptosystem”, Journal of Cryptology, Vol. 3 No. 2, Springer-Verlag, pp 63-80, 1991.

- [9] N. Koblitz, “Elliptic curve cryptosystems” Mathematics of computation - 48, pp. 203-209, 1987.
- [10] V. Miller, “Use of elliptic curves in cryptography”, Proceedings of CRYPTO’85, Springer-Verlag, pp. 417-426, Aug. 1985.
- [11] A. Menezes, S. Vanstone, and T. Okamoto, “Reducing elliptic curve logarithms to logarithms in a finite field”, STOC 1991, ACM Press, pp. 80-89, 1991.
- [12] G. Agnew, R. Mullin, and S. Vanstone, “A fast elliptic curve cryptosystem”, Lecture Notes in Computer Science #434, Proceedings of Eurocrypt’89, Springer-Verlag, pp. 706-708, Apr. 1989.
- [13] G. Agnew, R. Mullin, and S. Vanstone, “An implementation of elliptic curve cryptosystems over  $F_{2^{155}}$ ”, IEEE Journal on Selected Areas in Communications Vol. 11, No. 5, pp. 804-811, 1993.
- [14] G. Agnew, R. Mullin, and S. Vanstone, “On the development of a fast elliptic curve cryptosystem”, Lecture Notes in Computer Science, Advances in Cryptography - EUROCRYPT’92, Springer-Verlag, pp. 482-487.

# Messy Broadcasting In Networks

R. Ahlswede  
Universität Bielefeld  
Fakultät für Mathematik  
33501 Bielefeld  
Germany

H.S. Haroutunian  
Universität Bielefeld  
Fakultät für Mathematik  
33501 Bielefeld

L.H. Khachatrian  
Universität Bielefeld  
Fakultät für Mathematik  
33501 Bielefeld  
Germany

Dedicated to James L. Massey on the occasion of his 60th birthday.

## Abstract

In the classical broadcast model it is tacitly assumed that every member of the scheme produces the broadcasting in the most clever way, assuming either that there is a leader or a coordinated set of protocols. In this paper, we assume that there is no leader and that the state of the whole scheme is secret from the members; the members do not know the starting time and the originator and their protocols are not coordinated. We consider three new models of broadcasting, which we call “Messy broadcasting.”

## I Prologue

Among the many discoveries Jim has made until now one observation (Taschkent 1984) is that R.A. seldom does what most people expect him to do. Here we have made an attempt to give a contribution that has no relation to anything J. L. M. ever did. However, it relates to him. It is in the spirit of the following question: “Which travel time to Zürich can an absentminded mathematician guarantee, if at any station he chooses any train in any available direction without going to the same city twice?”

## II Introduction

Broadcasting refers to the process of message dissemination in a communication network whereby a message, originated by one of the members, is transmitted to all members of the network. A communication network is a connected graph  $G = (V, E)$ , where  $V$  is a set of

vertices (members) and  $E$  is a set of edges. Transmission of the message from the originator to all members is said to be broadcasting if the following conditions hold:

1. Any transmission of information requires a unit of time.
2. During one unit of time every informed vertex (member) can transmit information to one of its neighboring vertices (members).

## The Classical Model

For  $u \in V$  we define the broadcast time  $t(u)$  of vertex  $u$  as the minimum number of time units required to complete broadcasting starting from vertex  $u$ . We denote by  $t(G) = \max_{u \in V} t(u)$  the broadcast time of graph  $G$ . It is easy to see that for any connected graph  $G$ ,  $t(G) \geq \lceil \log_2 n \rceil$ , where  $n = |V|$ , since during each time unit the number of informed vertices can at most be doubled.

A minimal broadcast graph (MBG) is a graph with  $n$  vertices in which a message can be broadcast in  $\lceil \log_2 n \rceil$  time units.

The broadcast function  $\beta$  assigns to  $n$  as value  $\beta(n)$  the minimum number of edges in a MBG on  $n$  vertices. Presently exact values of  $\beta(n)$  are known only for two infinite sets of parameters of MBG's, namely, for  $\{n = 2^m : m = 1, 2, 3, \dots\}$  [1] and  $\{n = 2^m - 2 : m = 2, 3, \dots\}$  ([2] and independently [3]). Known are also the exact values of  $\beta(n)$  for some  $n \leq 63$  ([1], [4–7]). We recommend [8] as a survey of results on classical broadcasting and related problems.

## New Models

In this paper we consider three new models of broadcasting, which we call “Messy broadcasting”. We refer to them as  $M_1$ ,  $M_2$ , and  $M_3$ .

In the classical broadcast model it is tacitly assumed that every node (member) of the scheme produces the broadcasting in the most clever way. For this it is assumed that, either there is a leader who coordinates the actions of all members during the whole broadcasting process (which seems to be practically not realistic) or the member must have a coordinated set of protocols with respect to any originator, enough storage space, timing and they must know the originator and its starting time.

Now we assume that there is no leader, that the state of the whole scheme is secret for the members, the members do not know the starting time and the originator, and their protocols are not coordinated.

Moreover, even if the starting time and originator are known, and the scheme is public, it is possible that the nodes of the scheme are primitive. They have only a simple memory, which is not sufficient to keep the set of coordinated protocols. Technically it is much easier to build such a network. It is very robust and reliable.

In all models  $M_1$ ,  $M_2$ , and  $M_3$  in any unit of time every vertex can receive information from several of its neighbors simultaneously, but can transmit only to one of its neighbors.

## Model $M_1$

In this model in any unit of time every vertex knows the states of its *neighbors*, i.e., which are informed and which are not. We require that in any unit of time every informed vertex must transmit information to one of its noninformed neighbors.

## Model $M_2$

In this model we require that in any unit of time every informed vertex  $u$  must transmit the information to one of those of its neighbors that did not send the information to  $u$  and did not receive it from  $u$  before.

## A Model $M_3$

In this model we require that in any unit of time every informed vertex  $u$  must transmit the information to one of those neighbors that did not receive the information from  $u$  before.

For an originator  $u \in G$  the sequence of calls  $\sigma(u)$  is said to be a *strategy* for the model  $M_i$  ( $i = 1, 2, 3$ ) if

- a) Every call in  $\sigma(u)$  is not forbidden in model  $M_i$ ,  $i = 1, 2, 3$ .
- b) After these calls every member of the system got the information.

In broadcast model  $M_1$  for a vertex  $u \in V$  we define  $\Omega_1(u)$  to be the set of all broadcast strategies that start from originator  $u$ . For any vertex  $u \in V$  of the graph  $G = (V, E)$  let  $t_1^\sigma(u)$  be the broadcast time of  $u$  using strategy  $\sigma \in \Omega_1(u)$  i.e.,  $t_1^\sigma(u)$  is the first moment at which every vertex of the scheme got the information by strategy  $\sigma$ . We set  $t_1(u) = \max_{\sigma \in \Omega_1(u)} t_1^\sigma(u)$ .

Actually  $t_1(u)$  is the broadcast time from vertex  $u$  in the worst broadcast strategy. Let  $t_1(G)$  be the broadcast time of graph  $G$ , that is  $t_1(G) = \max_{u \in V} t_1(u)$ . Similarly for models  $M_2, M_3$ :  $\Omega_2(u), t_2(u), t_2(G), \Omega_3(u), t_3(u)$ , and  $t_3(G)$  can be defined. From these definitions it follows that

$$\Omega_1(u) \subseteq \Omega_2(u) \subseteq \Omega_3(u). \quad (1)$$

For  $i = 1, 2, 3$  we define  $\tau_i(n) = \min_{G=(V,E),|V|=n} t_i(G)$ .

From Equation 1 it follows that  $t_1(G) \leq t_2(G) \leq t_3(G)$  for every connected graph  $G$ , and hence  $\tau_1(n) \leq \tau_2(n) \leq \tau_3(n)$  for every positive integer  $n$ .

In Section VI we establish upper bounds on  $\tau_2(n)$  and  $\tau_3(n)$ . Optimal graphs in model  $M_1$  are described in Section VII and a lower bound for  $\tau_3(n)$  is derived in Section VIII. For trees we establish even exact results (Section IV with preparations in Section III). Here we can algorithmically determine the broadcast times (Section V).

## III Auxiliary Results Concerning Optimal Trees

In addition to the notions presented in the Introduction we need the following concepts.

For model  $M_i$ ,  $i = 1, 2, 3$ , we define  $t_i(u, v) = \max_{\sigma \in \Omega_i(u)} t_i^\sigma(u, v)$ , where  $t_i^\sigma(u, v)$  is the broadcast time when broadcasting according to strategy  $\sigma$  starts from originator  $u$  and the information comes to vertex  $v$ .

We denote by  $\rho(v)$  the local degree of vertex  $v$ . Suppose now that we are given a connected tree  $H$ . At first we notice that for every vertex  $u$  of any tree  $H$  the sets of strategies  $\Omega_1(u)$  and  $\Omega_2(u)$  (but not  $\Omega_3(u)$ ) are the same. Hence  $t_1(u) = t_2(u)$  and  $t_1(H) = t_2(H)$  for every tree  $H$ . In this part we use the abbreviation  $t(u)$  for  $t_1(u)$  and for  $t_2(u)$ .

First we consider the following problem. For given broadcast time  $t$  construct a tree with root  $u$  having maximal number of vertices  $g(t)$ , for which  $t(u) = t$ . This tree is called an optimal tree with root  $u$  and broadcast time  $t$  or in short  $(OTR, u, t)$ .

Let, for fixed broadcast time  $t(u) = t$ , an optimal tree  $T$  with root  $u$  be constructed and let  $\sigma_0$  be a strategy for which  $t(u) = t^{\sigma_0}(u) = \max_{\sigma} t^{\sigma}(u)$ . Denote by  $u_1, u_2, \dots, u_k$  the neighbors of root  $u$ . By the tree structure we can assume that, under the strategy  $\sigma_0$  in the unit of time  $i$  ( $i = 1, \dots, k$ ), the vertex  $u$  sends information to vertex  $u_i$ . After removing (in our minds) from the optimal tree all edges  $(u, u_i)$  for  $i = 1, \dots, k$  we get trees  $T_i$ ,  $i = 1, \dots, k$ . It is clear that  $\max_{1 \leq i \leq k} t(u_i) = t(u_k)$ , where for  $i = 1, \dots, k$   $t(u_i)$  is the broadcast time from  $u_i$  in tree  $T_i$ , because otherwise, if  $\max_{1 \leq i \leq k} t(u_i) = t(u_j) > t(u_k)$  for some  $1 \leq j < k$ , then by changing the steps  $j$  and  $k$  in the broadcast strategy  $\sigma_0$  we would get a strategy  $\sigma'_0$  for which  $t^{\sigma'_0}(u) > t^{\sigma_0}(u) = \max_{\sigma} t^{\sigma}(u)$ . This is a contradiction. It is also clear that for all  $i = 1, 2, \dots, k$  the trees  $T_i$  are  $(OTR, u_i, t(u_i))$ .

On the other hand, since the tree  $T$  is assumed to be optimal, necessarily

$$t(u_1) = t(u_2) = \dots = t(u_k) = t - k. \quad (2)$$

Indeed, if otherwise for some  $j \in \{1, \dots, k\}$   $t(u_j) < t(u_k)$ , then by taking subtree  $T_k$  instead of subtree  $T_j$  we will get a tree  $T'$  with  $t(T') = t(T)$  and number of vertices  $|T'| > |T|$ , which is a contradiction. Hence

$$g(t) = \max_k k g(t - k) + 1. \quad (3)$$

The first values of the function  $g$  are

$$g(1) = 2, \quad g(2) = 3, \quad g(3) = 5, \quad g(4) = 7, \quad g(5) = 11, \quad g(6) = 16, \quad g(7) = 23. \quad (4)$$

It can be shown that for  $t \geq 8$

$$g(t) = 3 \cdot g(t - 3) + 1. \quad (5)$$

Therefore, using the initial values in Equations 4, we have

**Lemma 1** (Models  $M_1$  and  $M_2$ ) *For given broadcast time  $t \geq 7$  the optimal tree with root  $u$  for which  $t(u) = t$  has  $g(t)$  vertices, where*

$$g(t) = \begin{cases} \frac{11 \cdot 3^{\frac{t-3}{3}} - 1}{2} & \text{for } t \equiv 0 \pmod{3} \\ \frac{47 \cdot 3^{\frac{t-3}{3}} - 1}{2} & \text{for } t \equiv 1 \pmod{3} \\ \frac{23 \cdot 3^{\frac{t-3}{3}} - 1}{2} & \text{for } t \equiv 2 \pmod{3}. \end{cases} \quad (6)$$

**Lemma 2.** *For any vertices  $v, a \in V$  of the tree  $T = (V, E)$ ,  $t(a, v) \leq t(v) - \rho(v) + 1$ . Moreover, for any  $v \in V$  there exists an  $a_0 \in V$  with  $t(a_0, v) = t(v) - \rho(v) + 1$ .*

**Proof:** For any  $v, a \in V$  we consider the unique path  $v \rightarrow w_1 \rightarrow w_2 \rightarrow \dots \rightarrow w_s \rightarrow a$  between  $v$  and  $a$ .

It is clear that  $t(v, a) = \rho(v) + \sum_{i=1}^s \rho(w_i) - s$ ,  $t(a, v) = \rho(a) + \sum_{i=1}^s \rho(w_i) - s$ , and hence

$$t(a, v) = t(v, a) - \rho(v) + \rho(a). \quad (7)$$

From the definition of  $t(v)$  it follows that

$$t(v) \geq t(v, a) + \rho(a) - 1. \quad (8)$$

Therefore  $t(a, v) \leq t(v) - \rho(v) + 1$ , as claimed. Moreover, since  $t(v)$  is the broadcast time of  $v$ , there exist a  $u_0 \in V$  and a strategy  $\sigma_0$  such that  $t(v) = \max_{u \in V} t(v, u) = t(v, u_0)$ . Obviously  $\rho(u_0) = 1$ . Taking  $a = u_0$  in Equation 7, we get

$$t(u_0, v) = t(v, u_0) - \rho(v) + \rho(u_0) = t(v) - \rho(v) + 1.$$

## IV Construction of Optimal Trees

### Models $M_1$ and $M_2$

Again we use the abbreviation  $t(u)$  for  $t_1(u)$  and  $t_2(u)$ .

For given  $t_0$  we consider the set  $\mathcal{T}(t_0)$  of all connected trees having broadcast time  $t_0$ . We define  $f(t_0) = \max_{T \in \mathcal{T}(t_0)} |T|$ , where  $|T|$  is number of vertices in tree  $T$ . We call the tree  $T$   $t_0$ -optimal if  $t(T) = t_0$  and  $|T| = f(t_0)$ , and present now our main tool for determining the quantity  $f(t_0)$ .

**Lemma 3.** *For every  $t_0 \geq 2$  there exists a  $t_0$ -optimal tree  $T$  having a center of symmetry, that is, there is a vertex  $v_0$  in  $T$  such that after removal of  $v_0$  the tree  $T$  is decomposed into trees  $H_1, \dots, H_s$  with equal cardinalities  $|H_1| = |H_2| = \dots = |H_s|$  and  $t(w_1) = t(w_2) = \dots = t(w_s)$ . Here  $w_i, i = 1, \dots, s$ , are neighbors of  $v_0$  and  $t(w_i)$  is the broadcast time of  $H_i$  when broadcasting starts from root  $w_i$ . Moreover, if  $t_0 \geq 5$ , then every optimal tree has a center of symmetry.*

**Proof:** Suppose  $T$  is  $t_0$ -optimal, that is  $t(T) = t_0$  and  $|T| = f(t_0)$ . Let  $v$  be any vertex of  $T$  with  $\rho(v) \geq 2$ .

Let  $v_1, v_2, \dots, v_k$  be the neighbors of  $v$ . If we remove (in mind) the vertex  $v$ , then the tree  $T$  decomposes into trees  $T_1 = (V_1, E_1), T_2 = (V_2, E_2), \dots, T_k = (V_k, E_k)$  with roots  $v_1, \dots, v_k$ . Let the labeling be such that  $t(v_1) \leq t(v_2) \leq \dots \leq t(v_k)$ , where  $t(v_i)$  is the broadcast time of  $T_i$  when broadcasting starts from vertex  $v_i$ .

Now let us estimate the quantity  $t(a, b)$  for  $a \in T_i$  and  $b \in T_j, i \neq j$ . We see by Lemma 2 that

$$t(a, b) \leq t(v_i) + k + t(v_j)$$

and there exist  $a' \in T_i, b' \in T_j$  for which  $t(a', b') = t(v_i) + k + t(v_j)$ .

Since  $t(v_1) \leq t(v_2) \leq \dots \leq t(v_k)$  we obtain

$$t(T) = \max \left\{ t(v_{k-1}) + k + t(v_k); \max_{a,b \in T_k} t(a,b) \right\}.$$

Now we show that  $t(v_1) = t(v_2) = \dots = t(v_{k-1})$  and that  $|T_1| = |T_2| = \dots = |T_{k-1}| = |T_0|$ , where  $T_0$  is the tree with root  $v_{k-1}$  and  $t(T_0) = t(v_{k-1})$  having a maximal number of vertices. According to Lemma 1  $|T_0| = g(t(v_{k-1}))$ . Indeed, if it is not the case we can change every tree  $T_i, i = 1, \dots, k-1$ , to  $T_0$  and get the tree  $T'$  with  $|T'| > |T|$ . But it is easy to verify that  $t(T') = t(T)$ , which contradicts the optimality of tree  $T$ .

Now if  $t(v_k) = t(v_{k-1})$ , then  $|T_k| \leq |T_0|$  and we can change also  $T_k$  to  $T_0$  to get tree  $T''$ , for which  $|T''| \geq |T|$ ,  $t(T'') = t(T)$ , and  $v$  is the center of symmetry of  $T''$ . Suppose that  $t(v_k) > t(v_{k-1})$  and consider the neighbors of vertex  $v_k : u_1, u_2, \dots, u_{r-1}, v$ . If we mentally remove the vertex  $v_k$ , then the tree  $T$  is decomposed into trees  $L_1, \dots, L_{r-1}, L(v)$  with roots  $u_1, \dots, u_{r-1}, v$ . Let  $t(u_1) \leq t(u_2) \leq \dots \leq t(u_{r-1})$  where  $t(u_i), i = 1, \dots, r-1$ , is the broadcast time of  $L_i$  when broadcasting starts from vertex  $u_i$ . Clearly  $t(v) = k-1 + t(v_{k-1})$ , where  $t(v)$  is the broadcast time of  $L(v)$  when broadcasting starts from vertex  $v$ .

We have to consider two cases: (i)  $t(v) \leq t(u_{r-1})$  and (ii)  $t(v) > t(u_{r-1})$ .

If we are in case (i), then it can be shown as above that  $t(u_1) = t(u_2) = \dots = t(u_{r-2}) = t(v)$ ,  $|L_1| = |L_2| = \dots = |L_{r-2}| = |L(v)|$ , and if  $t(u_{r-1}) = t(u_1) = \dots = t(u_{r-2}) = t(v)$ , then  $v_k$  is the center of the tree  $T$ . Otherwise we will continue our procedure by considering the neighbors of  $u_{r-1}$ . Hence the principle case is (ii):  $t(v) = k-1 + t(v_{k-1}) > t(u_{r-1})$ .

In this case we have already shown that  $t(u_1) = t(u_2) = \dots = t(u_{r-1})$ ;  $|L_1| = \dots = |L_{r-1}| = |L_0|$  where  $L_0$  is the tree with root  $u_{r-1}$ ,  $t(L_0, u_{r-1}) = t(u_{r-1})$ , and having maximal number of vertices equal to  $g(t(u_{r-1}))$  (see Lemma 1). Hence  $t(v_k) = r-1 + t(u_{r-1})$  and by our assumption  $r-1 + t(u_{r-1}) > t(v_{k-1})$ .

It is easy to verify that in this case (ii) we have  $t(T) = t(v_{k-1}) + k + r-1 + t(u_{r-1})$ .

Let us prove that  $t(v_{k-1}) = t(u_{r-1})$  or equivalently that  $|T_0| = |L_0|$ . Suppose that  $t(v_{k-1}) > t(u_{r-1})$  (or equivalently that  $|T_0| > |L_0|$ ). Then in tree  $T$  we remove the edge  $(v_k, u_1)$  with the rooted subtree  $(L_0, u_1)$  and add the new edge  $(v, v')$  with the rooted subtree  $(T_0, v')$ .

Using the restriction  $r-1 + t(u_{r-1}) > t(v_{k-1})$  it is easy to verify that for the obtained tree  $T'$  we have  $t(T') = t(T)$ . However this contradicts the optimality of  $T$  because  $|T'| > |T|$ . Similarly it can be proved that  $t(v_{k-1}) < t(u_{r-1})$  is impossible. Hence  $t(v_{k-1}) = t(u_{r-1}) = t_1$ ,  $|T_0| = |L_0|$  and  $t(T) = 2t_1 + k + r-1$ ,  $|T| = (k+r-2) \cdot |T_0| + 2$ .

Now we can transform our tree  $T$  into the new one  $T^*$  as follows: we remove vertex  $v_k$  with edges  $(v_k, v), (v_k, u_i)$  for  $i = 1, \dots, r-1$ , we add edges  $(v, u_i)$  for  $i = 1, \dots, r-1$  and add a new vertex  $v'$  with rooted subtree  $(T_0, v')$  and edge  $(v, v')$ .

We verify that  $t(T^*) = t(T) = 2t_1 + k + r-1$  and

$$|T^*| = (k+r-1)|T_0| + 1 \geq |T|. \quad (9)$$

Since  $T$  is optimal, we should have equality in Equation 9, which occurs only when  $|T_0| = 1$  (or equivalently when  $t_1 = 0$ ), i.e. all vertices  $v_i, u_j, i = 1, \dots, k-1; j = 1, \dots, r-1$ , are terminal vertices in  $T$ . Hence, if  $|T_0| = 1$ , we have  $|T| = k+r$  and  $t_0 = t(T) = k+r-1 = |T|-1$ . However, it is very easy to construct for every  $t_0 \geq 5$  a tree (not necessarily optimal) having more than  $t_0 + 1$  vertices.

Therefore, if  $t_0 \geq 5$ , the assumption (ii)  $t(v) > t(u_{r-1})$  is impossible and hence for  $t_0 \geq 5$  every optimal tree has a center of symmetry.

We verify that for  $t_0 = 3$  and  $|T| = 4$ , that for  $t_0 = 4$  and  $|T| = 5$ , and that all connected trees on 4 or 5 vertices are optimal. Among these optimal trees there are stars (which have center of symmetry) on 4 or 5 vertices, and this fact completes the proof.

Now let  $v$  be the center of symmetry of an  $t_0$ -optimal tree  $T$ ,  $t_0 \geq 2$ . That is, removing vertex  $v$  from  $T$  the tree  $T$  will be decomposed into  $s$  subtrees  $T_1, \dots, T_s$  with roots  $v_1, \dots, v_s$ ;  $t(T_1, v_1) = t(T_2, v_2) = \dots = t(T_s, v_s) = t_1$  and  $|T_1| = |T_2| = \dots = |T_s| = g(t_1)$ , where  $v_1, \dots, v_s$  are the neighbors of  $v$  and  $g(t_1)$  is described in Lemma 1.

We verify that  $t(T) = t_0 = 2t_1 + s$  and

$$|T| = s \cdot g(t_1) + 1 = (t_0 - 2t_1)g(t_1) + 1. \quad (10)$$

Therefore, by optimality of  $T$ ,  $t_1$  maximizes the quantity

$$\max_{0 \leq x < \frac{t_0}{2}} (t_0 - 2x)g(x) = (t_0 - 2t_1)g(t_1).$$

Using Equations 4 and 6 it is not difficult to find (details are omitted) an appropriate  $t_1$  (and hence value  $s$ ) for every fixed broadcast time  $t_0 \geq 2$  we have

**Theorem 1** (*Models  $M_1$  and  $M_2$* )

Let  $T$  be an optimal tree for which  $t(T) = t_0$  and  $t_0 \geq 2$ . Then

$$f(t_0) = |T| = \begin{cases} 3 & \text{for } t_0 = 2 \\ 4 & \text{for } t_0 = 3 \\ 5 & \text{for } t_0 = 4 \\ 7 & \text{for } t_0 = 5 \\ 9 & \text{for } t_0 = 6 \end{cases}$$

and for  $t_0 \geq 7$

$$f(t_0) = |T| = \begin{cases} 5 \cdot g\left(\frac{t_0-5}{2}\right) + 1, & \text{if } t_0 \equiv 1 \pmod{2} \\ 6 \cdot g\left(\frac{t_0-6}{2}\right) + 1, & \text{if } t_0 \equiv 0 \pmod{2}. \end{cases}$$

Using Theorem 1 and Equation 6 the following can be proved.

**Corollary 1** For large  $t_0$

$$t_0 = \frac{6}{\log_2 3} \cdot \log_2 |T| + O(1) \sim 3.785 \log_2 |T|.$$

### Model $M_3$

Since the optimal trees in models  $M_2$  and  $M_3$  are similar (but not the same!) we represent only the results.

We calculate now the quantity  $g'(t_0)$ , which as in case of model  $M_2$  is defined to be the cardinality of optimal tree  $H$  with root  $u$ , that is  $t_3(u, H) = t_0$  and for any tree  $H'$  with  $t_3(u, H') = t_0$  it follows that  $|H| \geq |H'|$ . The initial values of  $g'(t_0)$  are  $g'(1) = 2$ ,  $g'(2) = 3$ ,

$g'(3) = 4, g'(4) = 5, g'(5) = 7, g'(6) = 10, g'(7) = 13, g'(8) = 17, g'(9) = 22, g'(10) = 31, g'(11) = 41, g'(12) = 53, g'(13) = 69, g'(14) = 94, g'(15) = 125, g'(16) = 165, g'(17) = 213, g'(18) = 283.$

**Lemma 1' (Model  $M_3$ )** For  $t_0 \geq 18$  we have

$$g'(t_0) = \begin{cases} \frac{94 \cdot 4^{\frac{t_0-10}{5}-1}}{3}, & \text{if } t_0 \equiv 0 \pmod{5} \\ \frac{31 \cdot 4^{\frac{t_0-6}{5}-1}}{3}, & \text{if } t_0 \equiv 1 \pmod{5} \\ \frac{10 \cdot 4^{\frac{t_0-2}{5}-1}}{3}, & \text{if } t_0 \equiv 2 \pmod{5} \\ \frac{850 \cdot 4^{\frac{t_0-18}{5}-1}}{3}, & \text{if } t_0 \equiv 3 \pmod{5} \\ \frac{283 \cdot 4^{\frac{t_0-14}{5}-1}}{3}, & \text{if } t_0 \equiv 4 \pmod{5}. \end{cases}$$

**Lemma 3' (Model  $M_3$ )** For every  $t_0 \geq 2$  every optimal tree has a center of symmetry.

**Remark:** The difference between Lemmas 3 and 3' is the following: in the model  $M_2$  for  $t_0 = 3$  and  $t_0 = 4$  there are trees that are optimal but do not have centers of symmetry; in model  $M_3$  there are no such exceptions.

Lemma 2 can be repeated for model  $M_3$ .

**Theorem 1' (Model  $M_3$ )** Let  $H$  be a  $t_0$ -optimal tree and  $t_0 \geq 18$ . Then

$$|H| = \begin{cases} 8 \cdot g'\left(\frac{t_0-11}{2}\right) + 1, & \text{if } t_0 \equiv 1 \pmod{2} \\ 7 \cdot g'\left(\frac{t_0-10}{2}\right) + 1, & \text{if } t_0 \equiv 0 \pmod{2}, \end{cases}$$

where  $g'$  is the quantity described in Lemma 1'.

**Corollary 1' (Model  $M_3$ )**

For large  $t_0$ ,  $t_0 \sim 5 \cdot \log_2 |H|$ .

At the end of this paragraph we discuss the structures of optimal trees in models  $M_2$  and  $M_3$ .

Let  $T$  and  $H$  be optimal trees in models  $M_2$  and  $M_3$ , respectively, and let  $t_2(T) = t_3(H) = t_0$  and let  $t_0$  be large. From Lemmas 3 and 3' it follows:

In  $T$  and  $H$  there are centers of symmetry  $v \in T$  and  $u \in H$ . Now for  $t_0 \equiv 1 \pmod{2}$  we have  $\rho(v) = 5, \rho(u) = 8$  and for  $t_0 \equiv 0 \pmod{2}$  we have  $\rho(v) = 6, \rho(u) = 7$ . The distance from  $v$  to every terminal point in the tree  $T$  is of order  $\frac{t_0}{6}$  and the distance from  $u$  to every terminal point in the tree  $H$  is of order  $\frac{t_0}{10}$ .

It can be shown that every vertex  $v' \in T$  with  $d(v, v') < \frac{t_0}{6} - 3$  ( $d(v, v')$  means distance between  $v$  and  $v'$ ) has local degree  $\rho(v') = 4$ , and for every  $u' \in H$  with  $d(u, u') < \frac{t_0}{10} - 6$ ,  $\rho(u') = 5$ .

## V An Algorithm for Determining the Broadcast Time of a Tree

In this section we present an algorithm for determination of the broadcast time of any given tree.

## Models $M_1$ and $M_2$

Let us find the broadcast time  $t(u)$  of vertex  $u$  in tree  $T = (V, E)$ . Suppose vertex  $u$  has neighbors  $u_1, \dots, u_k$ , which have the broadcast times  $t(u_1), \dots, t(u_k)$  in trees  $T_i = (V_i, E_i)$  with roots  $u_i, i = 1, \dots, k$ , respectively. It is clear that the broadcast time of vertex  $u$  is  $t(u) = \max_{1 \leq i \leq k} t(u_i) + k$ . Our algorithm is based on this fact.

### The algorithm

**Step 1:** Label the terminal vertices of tree  $T$  with 0, that is, if  $\rho(v) = 1$ , then  $\ell(v) = 0$ .

**Step 2:** For all vertices  $v$  ( $v$  has no label), if  $\rho(v) = k$  and all  $k - 1$  neighbors  $v_1, \dots, v_{k-1}$  of  $v$  except  $v_k$  are labeled, then we label the vertex  $v$  with  $\ell(v) = \max_{1 \leq i \leq k-1} \ell(v_i) + k - 1$ .

**Step 3:** If all neighbors  $v_1, \dots, v_k$  of the vertex  $v$  are labeled ( $v$  has no label), then we label vertex  $v$  with  $\ell(v) = \max_{1 \leq i \leq k} \ell(v_i) + k$ .

**Step 4:** The broadcast time of vertex  $v$  (which got the label in Step 3) equals its label:  $t(v) = \ell(v)$ .

**Step 5:** If every  $v \in T$  has  $\ell(v)$ , go to Step 7. If  $v'$  is a neighbor of  $v$  and the broadcast time  $t(v)$  of vertex  $v$  is known, but  $t(v')$  is not known, then cancel the labels  $\ell(v) = t(v)$  and  $\ell(v')$ .

**Step 6:** If  $\rho(v) = k$  and  $v$  has neighbors  $v_1, v_2, \dots, v_{k-1}, v'$ , then we label vertex  $v$  with  $\ell(v) = \max_{1 \leq i \leq k-1} \ell(v_i) + k - 1$ . Go to Step 3.

**Step 7:** Stop.

It can be verified (details are omitted) that this algorithm assigns to every vertex its broadcast time.

## Model $M_3$

A similar algorithm can be designed and we leave it to the reader.

## VI An Upper Bound for $\tau_2(n)$ and $\tau_3(n)$

**Lemma 4** For any connected graph  $G = (V, E)$  with diameter  $d$  and  $\rho(G) \leq k$  we have

- (a)  $t_2(G) \leq d(k - 1) + 1$
- (b)  $t_3(G) \leq dk$ .

### Proof:

(a) We have to prove that in model  $M_2$ ,  $t(v, u) \leq d(k - 1) + 1$  for any  $v, u \in V$ .

Let  $v \rightarrow w_1 \rightarrow w_2 \rightarrow \dots \rightarrow w_{s-1} \rightarrow u$  be the shortest path from  $v$  to  $u$ . Since  $\rho(v) \leq k$ ,  $\rho(w_i) \leq k$  for  $i = 1, \dots, s - 1$ , after at most  $k$  units of time the vertex  $w_1$  will be informed, after at most  $2k - 1$  units of time the information comes to  $w_2$ , etc. and after at most  $s(k - 1) + 1$  units of time the information comes to vertex  $u$ .

Since the graph  $G$  has diameter  $d$  we have  $s \leq d$ . Therefore  $t(v, u) \leq d(k - 1) + 1$  for any  $v, u \in V$

(b) The proof is similar.

We need the following result due to Bollobás and de la Vega [9].

**Theorem** Suppose  $\varepsilon > 0$  and  $k \geq 3$  are fixed. Then if  $d$  is sufficiently large there exists a graph  $G = (V, E)$  with diameter  $d$  and  $\rho(G) \leq k$  for which

$$|V| = n \geq \frac{1 - \varepsilon}{2kd \log_2(k-1)} \cdot (k-1)^{d-1}.$$

Actually for every  $k \geq 3$  and large  $d$  we have

$$\log_2 n \geq (d-1) \log_2(k-1) - O(\log d).$$

Using (a) of Lemma 4 and the Theorem we conclude that for any fixed  $k \geq 3$  and sufficiently large  $d$  there exists a graph  $G = (V, E)$ ,  $\rho(G) \leq k$ , with diameter  $d$ ,  $|V| = n$ , and broadcast time

$$t_2(G) \leq \frac{k-1}{\log_2(k-1)} \cdot \log_2 n.$$

We verify that  $\min_{k \geq 3} \frac{k-1}{\log_2(k-1)} = \frac{3}{\log_2 3} \approx 1.89$  and that the minimum is assumed for  $k = 4$ .

Similarly, using Lemma 4 (b) and again the Theorem above we have that  $t_3(G) \leq \frac{k}{\log_2(k-1)} \log_2 n$ ,  $\min \frac{k}{\log_2(k-1)} = 2, 5$ , and that the minimum is assumed for  $k = 5$ .

We summarize our findings.

**Theorem 2** For sufficiently large  $n$

- (a)  $t_2(n) \leq 1.89 \log_2 n$
- (b)  $t_3(n) \leq 2, 5 \cdot \log_2 n$ .

## VII Some Optimal Graphs (Model $M_1$ )

In this Section we present for broadcast model  $M_1$  some graphs on  $n$  vertices, where  $n \leq 10$  and  $n = 14$ , with minimum possible broadcast time, that is, for these graphs

$$\tau_1(n) = t_1(G), \quad G = (V, E), \quad |V| = n.$$

For  $4 \leq n \leq 8$  the optimal graphs are cycles  $C_n$ .

Denote their vertex set by  $V_n^* = \{0, 1, \dots, n-1\}$  and their edge set by  $E_n^*$ . For  $n = 10$ ,  $\tau_1(10) = 4$  and the optimal graph is the well-known Peterson graph  $G = (\{0, 1, \dots, 4\} \cup \{0', 1', \dots, 4'\}, E_5^* \cup E_5' \cup \{\{i, i'\} : i = 0, 1, \dots, 4\})$ , where  $E_5' = \{\{0', 2'\}, \{2', 4'\}, \{4', 1'\}, \{1', 3'\}, \{3', 0'\}\}$ .

For  $n = 9$ ,  $\tau_1(9) = 4$  and the optimal graph is obtained from Peterson's graph by removing one vertex with its edges.

For  $n = 14$ ,  $\tau_1(14) = 5$  and the optimal graph is

$$G = (V_{14}^*, E_{14}^* \cup \{\{0, 5\}, \{1, 10\}, \{2, 7\}, \{3, 12\}, \{4, 9\}, \{6, 11\}, \{8, 13\}\}).$$

It is necessary to note that these graphs — except for the graphs on 9 and 10 vertices — are optimal even for broadcast model  $M_2$ .

## VIII A Lower Bound for $\tau_3(n)$

Let  $G = (V, E)$  be a connected graph for which  $t_3(G) = t_0$ , that is  $t_0 = t_3(G) = \max_{u \in V} \max_{\sigma \in \Omega_3(u)} t_3^\sigma(u)$ . We take an arbitrary originator  $v \in V$  and consider the following strategy  $\sigma_0 \in \Omega(v)$ .

In any unit of time  $t'$ ,  $t' \in \{1, 2, \dots, t_0 - 1\}$ , let  $N(t') = N_1(t') \cup N_2(t')$  be the set of informed vertices after  $t'$  units of time, where  $N_2$  is the set of “new” informed vertices, that is  $N_2$  is the set of those vertices of  $N$  that were not informed after  $t' - 1$  units of time. It means that every vertex  $u_i \in N_2, i = 1, \dots, |N_2|$  in the  $t'$ th moment received the information from some subsets  $V_i \subset N_1, i = 1, \dots, |N_2|$ ,  $V_i \cap V_j = \emptyset$ . Then the strategy  $\sigma_0$  is the following: in the  $(t' + 1)$ th unit of time every  $u_i \in N_2, i = 1, \dots, |N_2|$  sends the information back to any vertex from subset  $V_i, i = 1, \dots, |N_2|$ .

Hence, using the strategy  $\sigma_0$ , after  $t' + 1$  units of time, the cardinality of the set of informed vertices could increase at most by  $|N_1|$ . So, if we denote by  $n(k)$ ,  $k = 2, \dots, t$ , the cardinality of the set of informed vertices in the  $k$ th unit of time, then we have

$$|V| \leq n(k) \leq n(k-1) + n(k-2).$$

From here for  $t_0$  we have

$$|V| \leq n(t_0) \leq c \cdot \left(\frac{1 + \sqrt{5}}{2}\right)^{t_0} \text{ or } t_0 \geq \frac{1}{\log_2 \frac{1 + \sqrt{5}}{2}} \log_2 |V| \sim 1.44 \log |V|.$$

**Theorem 3** (*Model M<sub>3</sub>*)

$$\tau_3(n) \geq 1.44 \cdot \log_2 n.$$

## References

- [1] A. Farley, S. Hedetniemi, S. Mitchell, and A. Proskurowski, “Minimum broadcast graphs”, *Discrete Math.* 25, 189–193, 1979.
- [2] L.H. Khachatrian and H.S. Haroutunian, “Construction of new classes of minimal broadcast networks”, *Proc. of the third International Colloquium on Coding Theory*, Dilijan, 69–77, 1990.
- [3] M.J. Dinneen, M.R. Fellows, and V. Faber, “Algebraic constructions of efficient broadcast networks”, *Applied Algebra, Algebraic Algorithms and Error Correcting Codes*, 9. Lecture Notes in Computer Science 539, 152–158, 1991.
- [4] J.-C. Bermond, P. Hell, A.L. Liestman, and G. Peters, “Sparse broadcast graphs”, *Discrete Appl. Math.* 36, 97–130, 1992.
- [5] M. Mahéo and J.F. Saclé, “Some minimum broadcast graphs”, *Technical Report 685*, LRI, Université de Paris-Sud, 1991.
- [6] L.H. Khachatrian and H.S. Haroutunian, “On optimal broadcast graphs”, *Proc. of Fourth International Colloquium on Coding Theory*, Dilijan, 65–72, 1991.

- [7] R. Labahn, “A minimum broadcast graph on 63 vertices”, to appear in *Disc. Appl. Math.*
- [8] S.T. Hedetniemi, S.M. Hedetniemi, and A.L. Liestman, “A survey of broadcasting and gossiping in communication networks”, *Networks* 18, 319–349, 1988.
- [9] B. Bollobás and F. de la Vega, “The diameter of random regular graphs”, *Combinatorica* 2, No. 2, 125–134, 1982.

# On a Problem of Persi Diaconis

Elwyn Berlekamp  
University of California at Berkeley  
Berkeley, CA

Dedicated to Jim Massey on the occasion of his 60th birthday.

## Abstract

A problem of Persi Diaconis is to construct a cyclic sequence  $s_1, s_2, \dots, s_n$ , whose elements lie in the  $m$ -dimensional binary vector space, such that every  $d$ -dimensional subspace is spanned by precisely one subsequence of length  $d$ . Because the cyclic sequence has  $n$  subsequences of length  $d$ ,  $n$  must be the number of  $d$ -dimensional subspaces of the  $m$ -dimensional binary vector space, which is

$$\frac{(2^m - 1)(2^m - 2) \dots (2^m - 2^{d-1})}{(2^d - 1)(2^d - 2) \dots (2^d - 2^{d-1})}$$

This paper solves the case of  $d = 2$ .

## I Introduction

A problem of Persi Diaconis is to construct a cyclic sequence  $s_1, s_2, \dots, s_n$ , whose elements lie in the  $m$ -dimensional binary vector space, such that every  $d$ -dimensional subspace is spanned by precisely one subsequence of length  $d$ . This paper solves the case of  $d = 2$ .

Because the cyclic sequences has  $n$  subsequences of length  $d$ ,  $n$  must be the number of  $d$ -dimensional subspaces of the  $m$ -dimensional binary vector space, which is

$$\frac{(12^m - 1)(2^m - 2) \dots (2^m - 2^{d-1})}{(2^d - 1)(2^d - 2) \dots (2^d - 2^{d-1})} \tag{1}$$

As happens in many construction problems, the number of degrees of freedom appears to be far larger than the number of constraints. But there are still sufficiently many constraints to make the problem difficult. Even though one may suspect that there are many solutions, it may be difficult to exhibit any one of them.

A common engineering strategy for dealing with “design” problems of this sort is to impose additional constraints! If these additional constraints are wisely chosen, they may restrict the form of a potential solution enough that studying the restricted domain becomes much more feasible.

In Diaconis’ problem, with  $d = 2$ , we need to ensure that every two-dimensional subspace is spanned by some pair of consecutive elements,  $\{s_i, s_{i+1}\}$ . This subspace is evidently  $\{0, s_i, s_{i+1}, s_i + s_{i+1}\}$ . Yielding to the temptation to invoke the structure of the finite Galois field,  $GF(2^m)$ , we observe that each  $s_i$  is an element of  $GF(2^m) - \{0\}$ , and that the ratios  $\frac{s_{i+1}}{s_i} \neq 1$ .

If we have any cyclic sequence that is a solution, then multiplying every element in this sequence by the same constant yields another solution. To eliminate this  $(2^m - 1)$ -fold degeneracy among solutions, we are naturally led to an investigation of the ratio sequence,

$$\frac{s_2}{s_1}, \frac{s_3}{s_2}, \frac{s_4}{s_3}, \dots, \frac{s_n}{s_{n-1}}, \frac{s_1}{s_n}.$$

As an additional constraint, I attempted to construct the ratio sequence as a repetition of a much shorter sequence. For prime values of  $m$ , this approach soon succeeded. But for other values of  $m$ , complications arose that necessitated modifications to the construction. The final result, as described below, may remind the reader of pre-Copernican astronomy. We have cycles, epicycles, and perhaps epi-epicycles. I hope some reader can find a better way!

The next section of this paper investigates some local symmetries of the relevant objects: two-dimensional subspaces and conjugation of  $GF(2^m)$  over  $GF(2)$ . The following sections present the construction for the case of even  $m$  and odd  $m$ , and the final section fills in the details for the cases  $m \leq 6$ .

## II Symmetry

Suppose  $a + b + c = 0$ . We may view the points  $\{a, b, c\}$  as the vertices of a triangle. We may view the directed edges of this triangle as the following six ratios:

$$\frac{b}{a}, \quad \frac{c}{b}, \quad \frac{a}{c}, \quad \frac{a}{b}, \quad \frac{b}{c}, \quad \frac{c}{a}$$

The group of symmetries of the triangle is  $S_3$ , the symmetric group on three letters. We need to investigate the action of this group on the triangle’s directed edges.

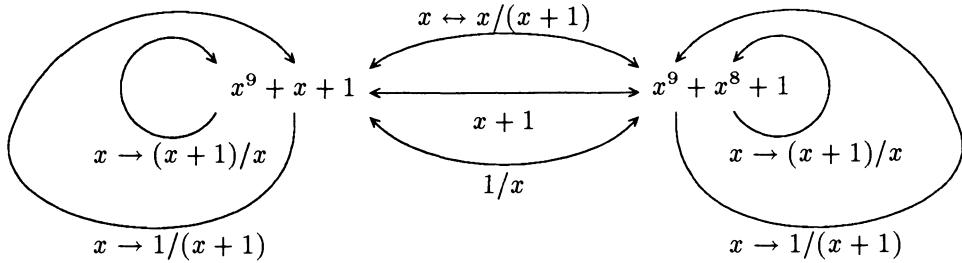
If  $\{s_{i+1}, s_i\} \subset \{a, b, c\}$ , then the ratio  $s_{i+1}/s_i$  might be any of these six quantities:  $\frac{b}{a}, \frac{c}{b}, \frac{a}{c}, \frac{a}{b}, \frac{b}{c}, \frac{c}{a}$ . If  $\xi$  is any one of these six quantities, then the others are  $(\xi + 1)/\xi$ ,  $1/(\xi + 1)$ ,  $1/\xi$ ,  $\xi/(\xi + 1)$ , and  $\xi + 1$ . All six of these quantities lie in the same orbit of  $LF(2)$ , the binary linear fractional group whose six elements are these transformations:

$$\begin{aligned}
\text{Order 1: } & \xi \leftrightarrow \xi \\
\text{Order 2: } & \xi \leftrightarrow 1/\xi \\
& \xi \leftrightarrow \xi + 1 \\
& \xi \leftrightarrow \xi/(\xi + 1) \\
\text{Order 3: } & \xi \rightarrow (\xi + 1)/\xi \rightarrow 1/(\xi + 1) \rightarrow \xi \\
& \xi \rightarrow 1/(\xi + 1) \rightarrow (\xi + 1)/\xi \rightarrow \xi
\end{aligned}$$

The action of this group partitions  $GF(2^m) - GF(2)$  into disjoint orbits. All elements not in  $GF(4)$  lie in orbits of size 6; the two elements in  $GF(4) - GF(2)$  lie in an orbit of size 2. These two elements are the roots of  $x^2 + x + 1$ , or equivalently, the nontrivial cube roots of 1.

$\mathcal{L}F(2)$  also acts on irreducible binary polynomials. We call such a polynomial *asymmetric* if it is not invariant under any nontrivial permutation in  $\mathcal{L}F(2)$ . The six irreducible binary polynomials of degree 5 are all asymmetric. Each of them can be transformed into any other by an appropriate transformation in  $\mathcal{L}F(2)$ .

If an irreducible binary polynomial of degree  $m > 2$  has a symmetry of order 3, then any of the permutations of order 2 transforms it to the only other polynomial in its orbit, as in this example:



Evidently,  $x \rightarrow (x+1)/x$  must permute the roots, so for some  $i$ ,

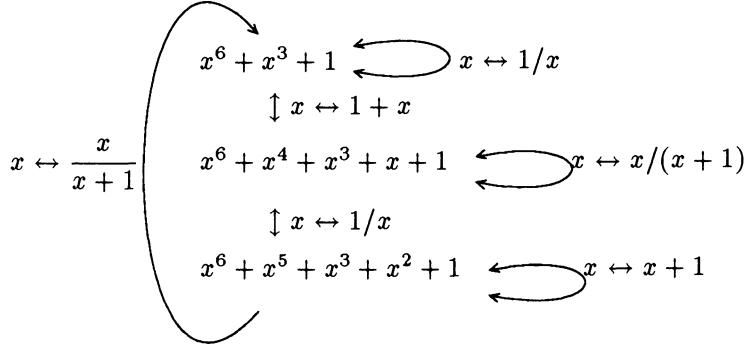
$$\begin{aligned}
\frac{\beta + 1}{\beta} &= \beta^{2^i} \\
1 + \beta &= \beta^{2^i+1} \\
1 + \beta^{2^i} &= (1 + \beta)^{2^i} = \beta^{2^{2^i}+2^i} \\
1 &= \beta + \beta^{2^i+1} = \beta^{2^{2^i}+2^i+1}
\end{aligned}$$

Depending on which of the two polynomials we choose, if their degrees are  $d$ , then either  $i = d/3$  or  $i = 2d/3$ , but in either case

$$\beta^{2^{2^i}+2^i+1} = \beta^{(2^d-1)/(2^{d/3}-1)}$$

We have just shown that any irreducible polynomial of degree  $> 3$  that is invariant under an  $\mathcal{L}F(2)$  symmetry of order 3 is not primitive; the multiplicative order of its roots is a proper divisor of  $2^d - 1$ .

If an irreducible binary polynomial of degree greater than 2 has a symmetry of order 2, then it belongs to an orbit of three polynomials which may be transformed into each other by  $\mathcal{LF}(2)$ . Each of these three must be invariant under a different one of the three permutations of order 2 in  $\mathcal{LF}(2)$ . For example,



In general, if an irreducible binary polynomial of degree greater than 2 is invariant under  $x \leftrightarrow 1/x$ , then the action of this permutation on its roots,  $\beta, \beta^2, \beta^{2^2}, \beta^{2^3}, \dots$ , reveals that for some  $i$ ,

$$\begin{aligned} \beta^{-1} &= \beta^{2^i} \\ \beta = (\beta^{-1})^{-1} &= \beta^{2^{2i}} \quad \text{so} \quad i = \frac{\text{degree}}{2} \text{ and } \beta^{2^{i+1}} - 1 = 0 \end{aligned}$$

Unlike the transformation  $x \leftrightarrow 1/x$ , the transformations  $x \leftrightarrow 1+x$  and  $x \leftrightarrow x/(x+1)$  do not, in general, impose any constraints on the orders of their roots. For example, in the particular case diagrammed above,  $x^6 + x^3 + 1$  has order 9, but  $x^6 + x^4 + x^3 + x + 1$  and its reciprocal each has order 63.

Thus, for  $m > 3$ , although no primitive binary polynomial can have a symmetry of order 3, some of them can have an  $\mathcal{LF}(2)$  symmetry of order 2. We will now present a simple counting argument to show that there must be some primitive binary polynomials that have no  $\mathcal{LF}(2)$  symmetries.

It is well known that the multiplicative group of  $GF(2^m)$  is cyclic, and that the number of primitive elements is given by Euler's function  $\varphi(2^m - 1)$ . These elements lie in  $(2^m - 1)/m$  conjugacy classes, each of which is the set of roots of a primitive polynomial of degree  $m$  over  $GF(2)$ . But the number of irreducible polynomials that have an  $\mathcal{LF}(2)$  symmetry of order 2 is at most

$$2 \cdot \frac{2^{d/2}}{d}$$

If  $d = 4$ , this number is equal to  $\frac{\varphi(15)}{4} = 2$ , so both primitive quartics are symmetric. However, if  $d \geq 6$ , then

$$\varphi(2^d - 1) \geq 2 \cdot 2^{d/2}$$

This is because  $\varphi$  is multiplicative, and for any odd prime  $p$ ,

$$\begin{aligned}\varphi(p^e) &= (p-1)p^{e-1} \geq \sqrt{p^e} \quad \text{always, and} \\ \varphi(p^e) &= (p-1)p^{e-1} \geq 2\sqrt{p^e} \quad \text{unless } p^e = 3 \text{ or } 5 \text{ or } 9\end{aligned}$$

So  $\varphi(2^d - 1) \leq 2 \cdot 2^{d/2}$  only if  $d$  is even and  $2^d - 1 \leq 45$ . Hence,

For every  $m \geq 5$ , there exist *asymmetric* primitive binary polynomials of degree  $m$ .

### III Construction

#### The Case of Even $m > 4$ , Constructed

Let  $\alpha$  be a root of  $f(x)$ , an asymmetric primitive polynomial of degree  $m$ . We represent each element in  $GF(2^m) - GF(2)$  by its  $\log_\alpha$ , written as an  $m$ -bit binary integer that cannot be all zero or all one. In this notation, conjugate elements have logs that are cyclic shifts of each other. The two elements in  $GF(4) - GF(2)$  have logs 1010...10 and 0101...01 =  $t = (2^m - 1)/3$ , or  $t = -(2^m - 1)/3$  if needed to ensure that  $t^2 \not\equiv -t \pmod{2^m - 1}$ . The three solutions of  $3x \equiv 0 \pmod{2^m - 1}$  are 0,  $t$ , and  $-t$ . Since  $t^2$  also solves this equation, it then follows that either  $t^2 \equiv t$  or  $t^2 \equiv 0$ .

Partition all of the other  $2^m - 4$  logarithms into orbits under  $\mathcal{LF}(2)$ . Each orbit has six elements. The  $m$  roots of  $f(x)$  are  $\overline{0} \quad \overline{01}, \overline{0} \quad \overline{101}, \overline{0} \quad \overline{0100}, \dots, \overline{10} \quad \overline{0}$ . These elements lie in  $m$  distinct orbits. In another orbit, find a “ $g$ ” relatively prime to  $2^m - 1$ . Pick the sign of  $g$  such that  $\mod{2^m - 1}, gt \equiv t$  if  $t^2 \equiv t$ , or  $gt \equiv -t$  if  $t^2 \equiv 0$ . Call this orbit “special.” An  $\mathcal{LF}(2)$  permutation of order 3 moves  $\underbrace{g \rightarrow h \rightarrow a}_{\leftarrow}$ , where  $g + h + a \equiv 0 \pmod{2^m - 1}$ .

In terms of the logarithms $_\alpha$  that we now use, the *ratio* sequence described Section I is now a *difference* sequence.

We now construct a difference sequence of length  $j = (2^m - 4)/6$ . This sequence will contain one element from each orbit. Its first element is “ $a$ .” A representative from each other orbit except those containing roots of  $f(x)$  may be picked arbitrarily as  $b_1, \dots, b_{j-m}$ . Finally, from an orbit containing the roots of  $f(x)$ , we pick *either* the log of the root *or* its negative, and make these last  $m$  choices in such a way that

$$a + \sum_{i=1}^{j-1} b_i \equiv t \pmod{2^m - 1}$$

We denote this sequence by  $[a, \overrightarrow{b}]$ .

This sequence and its  $\overrightarrow{b}$  subsequence are then concatenated and expanded into this longer sequence:

$$[-h, \overrightarrow{b}, a, \overrightarrow{b}, a, \overrightarrow{b}, t]$$

which we might write in abbreviated notation as

$$[-h, [\vec{b}, a,]^2, \vec{b}, t]$$

The final (complete) difference sequence is

$$[-h, [\vec{b}, a,]^2, \vec{b}, t]^{t-1} [-h, [\vec{b}, a,]^2, \vec{b}, -t]$$

## The Case of Even $m > 4$ , Proved

We claim the sequence of  $\log_\alpha$  that has these differences is a sequence with the property we seek.

Let us imagine this sequence arranged in  $t$  rows, read from left to right, top to bottom, in the conventional order. Each row then satisfies this sequence of “internal” differences:

$$-h, [\vec{b}, a,]^2, \vec{b}$$

The wrap-around difference between the last entry in any row and the first entry in the following row is  $t$ . The wrap-around difference from the last entry in the bottom row to the first entry in the top row is  $-t$ . We call these differences “external.”

The sum of the differences in the internal subsequence  $[a, \vec{b}]$  is  $t$ . For comparison, we might notice that the sum of  $[a, \vec{b}]^3$  is 0, so sum  $[-h, \vec{b}, [a, \vec{b}]^2]$  is  $-h - a = g$ , and the sum of *all* differences in the complete sequence is

$$t(g + t) - 2t \equiv 0$$

Hence, the sequence is cyclic, as claimed.

Now consider any “typical” orbit, which excludes both  $\{t, -t\}$  and the “special” orbit. Its representative,  $\gamma = \alpha^{\log \gamma}$ , appears in the difference sequence three times in each of  $t$  rows, or  $2^m - 1$  times altogether. We claim that each time it appears, it is followed by a *different* nonzero field element in the final (solution) sequence, because in the final sequence

Corresponding entries on different rows differ mod  $t$ .  
and

Corresponding entries on the same row differ by  $t$  or  $-t$ .

So every triplet  $\{A, B, C\}$  such that  $A + B + C = 0$  and for which some pair has quotient  $\gamma$  is spanned by a pair of consecutive elements in the final sequence.

Next consider the “special” orbit. It is represented sometimes by  $-h$ , sometimes by  $a$ . But, taken together, each of the  $3t$  occurrences of a representative of the special orbit in the difference sequence is preceded by a *different* nonzero field element in the final (solution) sequence. And the relationship between  $a$  and

$-h$  is such that either choice generates the same triangle including  $A$ . One choice generates  $\{A, B, C\}$  via  $B$  and  $A$ ; the other, via  $C$  and  $A$ .

Finally, the orbit containing  $\{t, -t\}$  also works. Unlike all other orbits, which represent  $(2^m - 1)$  different triplets (each an arbitrary scalar multiple of the others), this orbit represents only  $t$  triplets. Because corresponding entries in the same column of the final solution differ mod  $t$ , each such triplet is spanned precisely once.

## The Case of Odd $m \geq 5$

This is much easier. Let  $\overrightarrow{b}$  contain one representative from each of the  $(2^m - 2)/6$  orbits, leaving the orbits containing the roots of the asymmetric polynomial  $f(x)$  until last. Then choose among 1 or  $-1$ , 2 or  $-2$ , 4 or  $-4$ ,  $\dots$ , and do so in such a way to ensure that the sum of the  $b$ 's is relatively prime to  $2^m - 1$ . Then  $[\overrightarrow{b}]^{2^m-1}$  is a difference sequence that works. Details are a proper subset of the arguments for the case of even  $m$ .

## The Case $m = 6$

Let  $\alpha$  be a root of  $x^6 + x + 1 = 0$ . Then  $\mathcal{LF}(2)$  partitions  $GF(2^6) - GF(2^2)$  into the following equivalence classes, where each element is denoted by its  $\log_\alpha$ :

$$\begin{aligned} & 1, \quad 6, \quad 5, \quad -1, \quad -6, \quad -5. \\ & 2, \quad 12, \quad 10, \quad -2, \quad -12, \quad -10. \\ & 4, \quad 24, \quad 20, \quad -4, \quad -24, \quad -20. \\ & 8, \quad 48, \quad 40, \quad -8, \quad -48, \quad -40. \\ & 16, \quad 33, \quad 17, \quad -16, \quad -33, \quad -17. \\ & 32, \quad 3, \quad 34, \quad -32, \quad -3, \quad -34. \\ \\ & 7, \quad 26, \quad 37, \quad -7, \quad 19, \quad -19. \\ & 14, \quad -11, \quad 11, \quad -14, \quad 38, \quad -38. \\ & 28, \quad -22, \quad 22, \quad -28, \quad 13, \quad -13. \\ \\ & 27, \quad 18, \quad 54, \quad 36, \quad 45, \quad 9. \end{aligned}$$

Acting together, conjugation and  $\mathcal{LF}(2)$  partition  $GF(2^6) - GF(2^2)$  into only three sets, consisting of the first six rows above, the next three rows, and the last row. The element  $g$  must be chosen from among the last four rows, and it must be relatively prime to 63. Since  $t = 21$  and  $21^2 \equiv 0 \pmod{63}$ , we also require that  $gt \equiv -t \pmod{63}$ , or  $g \equiv -1 \pmod{3}$ .

One suitable choice is  $g = 11$ , which we select. We choose  $a = 14$  and  $h = 38$  instead of vice versa. We then arbitrarily take the first number in each of the other last four 6-tuples as its representative:

$$[a, \overrightarrow{b}] = 14, 7, 28, 27, b_4, b_5, \dots, b_9$$

Since

$$14 + 7 + 28 + 27 \equiv 13 \pmod{63}$$

and we desire that

$$a + \sum_{i=1}^9 b_i \equiv 21$$

we seek

$$\sum_{i=4}^9 b_i \equiv 21 - 13 = 8$$

We introduce  $c_0, c_1, c_2, \dots, c_5$ , each of whose values will be 0 or 1, such that

$$\begin{aligned} b_4 &= \pm 2^5 = c_5 \cdot 2^6 - 2^5 \\ b_5 &= \pm 2^4 = c_4 \cdot 2^5 - 2^4 \\ b_6 &= \pm 2^3 = c_3 \cdot 2^4 - 2^3 \\ &\vdots \\ b_9 &= \pm 2^0 = c_0 \cdot 2^1 - 2^0 \end{aligned}$$

whence

$$\sum_{i=4}^9 b_i = 2 \cdot \sum_{i=0}^5 c_i 2^i - 63 \equiv 2 \sum_{i=0}^5 c_i 2^i$$

So we can solve the equation

$$\sum_{i=4}^9 b_i = 8$$

by setting  $\sum c_i 2^i = 4$ , whence  $c_2 = 1$  and  $c_0 = c_1 = c_3 = c_4 = c_5 = 0$  and

$$\overrightarrow{b} = 7, 28, 27, -32, -16, -8, +4, -2, -1$$

The total cyclic ratio sequence has length  $21 \cdot 31$ , which we view as 21 rows of 31 entries each.

Each row except the last is

$$\overrightarrow{\text{row}} = [-h, \overrightarrow{b}, a, \overrightarrow{b}, a, \overrightarrow{b}, t]$$

The last row is

$$\overrightarrow{\text{row}}' = [-h, \overrightarrow{b}, a, \overrightarrow{b}, a, \overrightarrow{b}, 2t]$$

The sum mod 63, of the 31 entries in  $\overrightarrow{\text{row}}$ , is computed as follows:

$$\begin{aligned}
 -38 + 7 + 28 + 27 - 32 - 16 - 8 + 4 - 2 - 1 &= 11 + 21 \\
 +14 + 7 + 28 + 27 - 32 - 16 - 8 + 4 - 2 - 1 &+ 21 \\
 +14 + 7 + 28 + 27 - 32 - 16 - 8 + 4 - 2 - 1 &+ 21 \\
 +21 &+ 21 \\
 &= 11 + 0 + 21
 \end{aligned}$$

The sum of the 31 entries in the last row is  $11 + 42$ . So the sum of all 21 rows is

$$20(11 + 21) + (11 + 42) = 21 \cdot 11 + 21^2 + 21 = 21 \cdot 12 = 0,$$

as required.

### The Case $m = 4$

Let  $\alpha$  be a root of  $x^4 + x + 1$ . There are only two 6-tuples:

$$\begin{array}{l} 1, -1, 4, 11, 3, 12 \\ \text{and } 2, -2, 8, 7, 6, 17 \end{array}$$

$$\begin{array}{l} \text{Take } \overrightarrow{\text{row}} = 1, 2, 1, 2, 1, 2, 5 \\ \text{and } \overrightarrow{\text{row}'} = 1, 2, 1, 2, 1, 2, 10 \end{array}$$

The full cyclic ratio sequence of length  $5 \cdot 7$  is  $\overrightarrow{\text{row}}^4, \overrightarrow{\text{row}}'$ .

### The Case $m = 5$

Let  $\alpha$  be a root of  $x^5 + x^2 + 1$ . There are five 6-tuples:

$$\begin{array}{l} 1, -1, 18, 13, 17, 14 \\ \text{and conjugates thereof.} \end{array}$$

Let  $\overrightarrow{\text{row}} = 16, -8, -4, -2, -1$ . The full cyclic ratio sequence is  $\overrightarrow{\text{row}}^{31}$ .

### The Case $m = 3$

Let  $\alpha$  be a root of  $x^3 + 1$ . Let  $\overrightarrow{\text{row}} = 1$ . The full cyclic ratio sequence is  $\overrightarrow{\text{row}}^{37}$ .

# Aspects of Linear Complexity

S Blackburn, G Carter\*, D Gollmann, S Murphy,  
K Paterson, F Piper, P Wild

Royal Holloway Information Security Group  
University of London  
Egham, Surrey TW20 0EX  
England

## Abstract

This paper summarizes the results of four separate research topics related to the generation of binary sequences by linear feedback shift registers for applications in cryptographic systems.

## I Introduction

The work of Jim Massey and his research students has had a significant impact on the cryptographic research here at Royal Holloway. One of Jim's many major attributes is his willingness to encourage other people's research and we are all very grateful for our close collaboration with him. Consequently when one of us was invited to contribute to this volume it seemed appropriate for us to combine and produce this 'joint effort'.

In this paper we summarize the results of four separate research topics studied by the group. They can all be 'linked' to Jim through his work on binary sequences but, more importantly, each of the authors readily acknowledges the personal contribution that Jim afforded by informal conversations and inspiring lectures.

## II Local Linear Complexity Profiles

The wide use of linear feedback shift registers (LFSRs) to generate sequences, and the fact that the Berlekamp-Massey algorithm (Massey [11]) efficiently determines the shortest LFSR that can generate a given sequence, has made linear complexity an important property of sequences in cryptographic use.

It has long been recognized that sequences generated for use as keystreams in a stream cipher should appear to be random. They should disguise the statistical properties of the plaintext so that they are not reflected in the ciphertext and a cryptanalyst who knows part of the keystream should not be able to predict subsequent bits of it.

The local linear complexity of a finite sequence  $s_0, s_1, \dots, s_{n-1}$  is the length  $L = L(n)$  of the shortest linear feedback shift register that can be used to generate the sequence (as

---

\*Smith System Engineering, Guildford

the first  $n$  terms). A portion of keystream of linear complexity  $L$  is completely determined by  $2L$  consecutive bits and the corresponding LFSR that generates it may be used to predict other keystream bits. It follows that keystream sequences should have large linear complexity. However this is not the only criterion that may be applied to keystreams using linear complexity. Statistical information may also be gleaned from calculations of linear complexity.

The Berlekamp-Massey algorithm, in computing  $L(n)$ , also computes the local linear complexities  $L(1), L(2), \dots, L(n-1)$  of all the subsequences beginning with  $s_0$ . The vector  $(L(1), \dots, L(n))$  is termed the local linear complexity profile of the sequence  $s_0, \dots, s_{n-1}$ . The sequence is said to jump with  $s_{k-1}$  if  $L(k) - L(k-1) > 0$ .  $L(k) - L(k-1)$  is known as the height of the jump. The distance between successive jumps is known as a step.

Rueppel [18] considered the local linear complexity profile of a random sequence and showed that the expected height of a jump is two and the expected length of a step is four. Moreover for large  $n$  the expected value of  $L(n)$  is approximately  $n/2$  with variance approximately  $86/81$ . Thus one expects a random sequence to have a local linear complexity profile to follow closely a line of slope  $1/2$ . Indeed Niederreiter [13] has shown that asymptotically this is the case for almost all sequences.

However the local linear complexity profile of keystream sequences should not follow the line of slope  $1/2$  too closely as Wang and Massey's [19] perfect profile characterization theorem shows. A sequence is said to have the perfect linear complexity profile if all the jumps in local linear complexity have height 1. Wang and Massey [19] show that  $s_0, \dots, s_{n-1}$  has the perfect profile if and only if  $s_0 = 1$  and  $s_{2i} = s_{2i-1} + s_{i-1}$  for  $1 \leq i \leq (n-1)/2$ .

Carter [2] has extended the work of Wang and Massey to show that sequences satisfying certain linear equations similar to those of the perfect profile characterization theorem also have a linear complexity profile in which the jumps are constrained. Also, analogous to the results of Rueppel, Carter, and Niederreiter consider the number of jumps in local linear complexity for a random sequence. For a random sequence of large length  $n$  the expected value of the number of jumps is approximately  $n/4$  and the variance is approximately  $n/8$ .

The distribution of jumps in the local linear complexity of a random sequence has also been determined [2]. Consideration of the Berlekamp-Massey algorithm shows that the linear complexity of the sequence  $s_0, \dots, s_{n-1}$  can only jump with the  $m$ th bit  $s_{m-1}$  in the sequence if  $L(m-1) \leq (m-1)/2$ . If this inequality holds then the linear complexity jumps or not according to whether or not  $s_{m-1}$  is the next bit output by the current LFSR found by the algorithm. For a random sequence this probability is  $1/2$ . So, when  $L(m-1) \leq (m-1)/2$ , the linear complexity jumps with probability  $1/2$ . If there is a jump then the height of the jump is  $m - 2L(m-1)$  (since the new linear complexity  $L(m) = m - L(m-1)$ ). It follows that the probability that a jump has height  $k$  is  $(1/2)^k$ . Thus the jump heights for a random sequence are distributed according to the geometric distribution with parameter  $1/2$ .

Carter and Niederreiter have developed statistical tests based on the number of jumps and the distribution of jumps in the local linear complexity profile of a sequence. These may be used in conjunction with other statistical tests to test local randomness properties of keystreams.

### III Rapid Generation of $m$ -Sequences

The properties of  $m$ -sequences have been intensively studied in cryptology because of their easy generation in hardware and software, their good distribution properties and because the linear complexity of sequences resulting from combining several  $m$ -sequences can be readily analyzed. Since many applications require the production of  $m$ -sequences at high rates, it is desirable to investigate methods of speeding up  $m$ -sequence generation. We present a technique that interleaves the outputs of  $k$  identical LFSRs to produce a single  $m$ -sequence at  $k$  times the rate of an individual register. The construction is given in [16] for the case where  $k$  is a power of 2, and is generalized in [1].

Let  $(s^0), \dots, (s^{k-1})$  be binary sequences where

$$(s^j) := s_0^j, s_1^j, \dots$$

We say that the sequence

$$(z) := z_0, z_1, \dots$$

is produced by interleaving  $(s^0), \dots, (s^{k-1})$  if

$$z_{ki+j} = s_i^j$$

for all nonnegative integers  $i$  and  $j$  such that  $0 \leq j < k$ . In other words,  $(z)$  is produced by taking one element in turn from each of the sequences  $(s^0), \dots, (s^{k-1})$ .

**Theorem 1** Let  $(s^0)$  be an  $m$ -sequence of period  $2^n - 1$ . Let  $k$  be a positive integer coprime to  $2^n - 1$  and let  $k^{-1}$  be the inverse of  $k$  mod  $2^n - 1$ . Define sequences  $(s^1), \dots, (s^{k-1})$  by

$$s_i^j := s_{i+jk^{-1}}^0,$$

so  $(s^j)$  is a shift of  $(s^0)$  by  $jk^{-1}$  terms. Then the sequence  $(z)$  resulting from the interleaving of the sequences  $(s^0), \dots, (s^{k-1})$  is an  $m$ -sequence of period  $2^n - 1$ .

*Proof:* Using the trace representation of an  $m$ -sequence, we may write

$$s_i^0 = Tr(A\alpha^i)$$

where  $A \in \mathbb{F}_{2^n} \setminus \{0\}$  and where  $\alpha$  is a primitive element of  $\mathbb{F}_{2^n}$ . Then

$$z_{ki+j} = s_i^j = Tr(A\alpha^{i+jk^{-1}}) = Tr(A(\alpha^{k^{-1}})^{ki+j}),$$

for all nonnegative integers  $i$  and  $j$  such that  $0 \leq j < k$ . Since  $\alpha^{k^{-1}}$  is a primitive element of  $\mathbb{F}_{2^n}$ ,  $(z)$  is an  $m$ -sequence, as required.  $\square$

This construction may be extended to the case when  $k$  and  $2^n - 1$  are not coprime: However, in this case the sequences to be interleaved are not in general  $m$ -sequences.

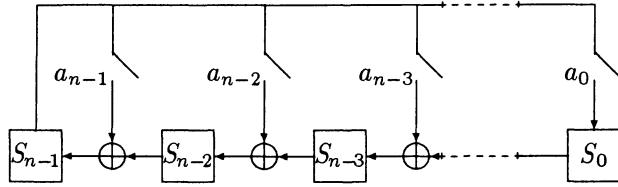


Figure 1: An LFSR of Galois type

## IV Alternative Output of Feedback Registers

Linear feedback shift registers are widely used to produce sequences, such as  $m$ -sequences, with high periods and good distribution properties. However, a drawback with using such a device in some applications is the low linear complexity of sequences by an LFSR. In [17], Robshaw investigated nonstandard methods of outputting a sequence from an LFSR. He uses an LFSR of Galois type (see Figure 1) as the basis for his construction.

**Theorem 2** *Let  $R$  be a primitive Galois register of length  $n$  with feedback polynomial  $m$ . If we take the contents of  $2^r$  adjacent stages and output them in series whilst clocking the register  $2^r$  times then the output sequence is either*

- (i) *an  $m$ -sequence with minimal polynomial  $m$  if there is no feedback to the stages from which we sample, or*
- (ii) *a sequence of period  $2^r(2^n-1)$  with minimum polynomial  $m^k$  where  $2^{r-1}+1 \leq k \leq 2^r$ .*

Thus, by extracting a sequence from an LFSR in a nonstandard fashion, sequences with high linear complexity and large periods may be generated.

## V Clock Controlled Shift Registers

In the previous section, we have shown how to achieve high linear complexity with LFSRs by choosing appropriate output schemes. An alternative source of nonlinearity is the clocking input of the LFSR [3]. Instead of stepping a LFSR at each time step, we use the output of an LFSR, with output sequence  $(a_i)$  of period  $q$  and linear complexity  $L_1$ , to drive a second LFSR, with output sequence  $(b_i)$  of period  $p$  and linear complexity  $L_2$ . In a simple stop-and-go clocking scheme, the second register steps if the clock input  $a_i$  is 1 and remains unchanged if  $a_i = 0$ . Define  $\sigma(i)$  to be the displacement at time  $i$ , i.e.

$$\sigma(i) = \sum_{j=0}^{i-1} a_j ,$$

then the clock-controlled register will output the sequence  $(b_{\sigma(i)})$ . Let  $d := \sigma(q)$  denote the total number of shifts within a period of the driving register. When we arrange the sequence  $(b_{\sigma(i)})$ , which has period at most  $p \cdot q$ , in an  $p \times q$ -array, i.e.

$$\begin{matrix} b_0 & b_{\sigma(1)} & b_{\sigma(2)} & \dots & b_{\sigma(q-1)} \\ b_d & b_{\sigma(1)+d} & b_{\sigma(2)+d} & \dots & b_{\sigma(q-1)+d} \\ b_{2d} & b_{\sigma(1)+2d} & b_{\sigma(2)+2d} & \dots & b_{\sigma(q-1)+2d} \\ \vdots & \vdots & \vdots & & \vdots \\ b_{(p-1)d} & b_{\sigma(1)+(p-1)d} & b_{\sigma(2)+(p-1)d} & \dots & b_{\sigma(q-1)+(p-1)d} \end{matrix}$$

we see that each column contains a  $d$ -decimation of the sequence  $(b_i)$ . It is obviously of cryptographic benefit to choose  $(a_i)$  so that  $d$  and  $p$  are coprime. For the sake of simplicity, assume that  $(b_i)$  has an irreducible minimal polynomial  $f(x)$  with root  $\alpha$ . The decimated sequence  $(b_{di})$  has then the irreducible minimal polynomial  $f_d(x)$  with root  $\alpha^d$  and the sequence  $(b_{\sigma(i)})$  has as its minimal polynomial a factor of  $f_d(x^q)$ . This polynomial can be irreducible only if all prime factors of  $q$  divide  $p$ . If  $p$  is prime, then  $q$  has to be a power of  $p$ .

The statistical properties of the sequence  $(b_{\sigma(i)})$  are not overly attractive. Adding the output of the driving register to the output of the clock-controlled register, we are able to combine the statistical properties of the first with the high linear complexity of the second. As a matter of fact, if  $f_d(x^q)$  is irreducible then the sequence  $(a_i \oplus b_{\sigma(i)})$  has linear complexity  $L_1 + qL_2$ .

Using clock-control recursively, we can build cascades of clock-controlled shift registers. Above, we have seen that the simplest way to achieve high linear complexity (and period) is to construct such a cascade from registers of uniform length  $p$ . If, for example,  $p$  is chosen so that  $(x^p - 1)/(x - 1)$  is irreducible, then the output of a cascade of length  $n$  will have linear complexity at least equal to

$$(1 + p + p^2 + \dots + p^{n-1})(p - 1) = p^n - 1.$$

## VI Linear Complexity and de Bruijn Sequences

Linear complexity has recently been applied in a novel way as a tool in the construction of de Bruijn sequences, their generalizations to two and more dimensions and related combinatorial objects.

Let  $(s) = s_0, s_1, \dots$  be a binary sequence of period  $2^k$ , for some  $k \geq 0$ . We define the left shift operator  $E$  acting on  $(s)$  as follows:

$$E(s) = (d) \text{ where } d_i = s_{i+1} \text{ for all } i \geq 0$$

and denote the linear complexity of  $(s)$  by  $l(s)$ . Then according to the results of Chan, Games, and Key, [4],

$$l(s) = l \Leftrightarrow (E + 1)^{l-1}(s) = 1, 1, \dots \quad (1)$$

the ‘all one’ sequence. Moreover,  $m(X)$ , the minimal polynomial of  $(s)$ , is  $(X + 1)^{l(s)}$ . Following from this we have:

**Theorem 3** [4] Suppose  $(s)$  has least period  $2^k$ . Then

$$2^{k-1} + 1 \leq l(s) \leq 2^k \quad (2)$$

and  $l(s) = 2^k$  if and only if  $s_0 + s_1 + \dots + s_{2^k-1} = 1$ , i.e. if and only if  $(s)$  has odd weight.

In [10], Lempel made a study of the operator  $D = E + 1$  in the context of cycles in the de Bruijn graph, and showed that  $D$  acts as a graph homomorphism from the de Bruijn graph of order  $k$ ,  $G_k$  to the graph  $G_{k-1}$ . This result leads to a description of the subsequences of length  $k$  of a sequence of the set  $D^{-1}(s)$  in terms of the subsequences of length  $k - 1$  of sequence  $(s)$ :

**Theorem 4** [10] Let  $(s)$  be a binary sequence of period  $t$ . If  $(s)$  has even weight then  $D^{-1}(s)$  consists of a pair of complementary sequences of period  $t$ . If  $(s)$  has odd weight then  $D^{-1}(s)$  consists of one self-complementary sequence of period  $2t$ . In both cases, the subsequences of length  $k$  of the sequences of  $D^{-1}(s)$  are of the form

$$(0, x_1, x_1 + x_2, \dots, x_1 + x_2 + \dots + x_{k-1})$$

and its complement, where  $(x_1, x_2, \dots, x_{k-1})$  is a subsequence of  $(s)$ .

From (1) above, we can deduce that applying  $D^{-1}$  to a sequence  $(s)$  of period  $2^k$  results in sequences whose linear complexity is one greater than that of  $(s)$ . The period and weight of such sequences is then determined by (2). Thus knowledge of the linear complexity of a sequence  $(s)$  can be used to control the period of the sequences resulting on repeated application of  $D^{-1}$  to  $(s)$ . Paterson [14] has used these ideas to show that the necessary conditions of [12] for the existence of binary perfect maps are also sufficient.

An  $(r, s; u, v)$  binary perfect map is defined to be an  $r \times s$  periodic binary array  $A$  with the property that every  $u \times v$  array arises as a sub-array of  $A$  exactly once (thus perfect maps are the generalization of the de Bruijn sequences to two dimensions).

**Theorem 5** [12] [14] An  $(r, s; u, v)$  binary perfect map exists if and only if

- (i)  $rs = 2^{uv}$ ,
- (ii)  $u < r$  or  $u = r = 1$ ,
- (iii)  $v < s$  or  $v = s = 1$ .

Paterson makes use of a class of perfect maps constructed by Etzion [5] and two constructions of Fan *et al.*, [7]. In the first of these constructions an  $(r, s; u, v)$  perfect map whose columns all have even weight was used to produce a  $(r, 2^v s; u + 1, v)$  perfect map and in the second an  $(r, s; u, v)$  perfect map whose columns all have odd weight was used to produce a  $(2r, 2^{v-1} s; u + 1, v)$  PM. Both of these constructions rely on applying  $D^{-1}$  to the sequences formed from the columns of the original perfect map. The essence of Paterson's result lies in determining conditions that allow the repeated application of these constructions with the perfect maps constructed by Etzion as starting point. These conditions can be described in terms of the linear complexities of the sequences that form the rows and columns of the perfect maps.

The use of linear complexity in the construction of perfect maps has recently been extended from the binary case to arbitrary finite fields by Paterson in [15], where a large class of nonbinary perfect maps, analogous to that of [5], was obtained.

## References

- [1] S.R. Blackburn. Increasing the Rate of Output of  $m$ -Sequences. *Preprint*.
- [2] G.D. Carter. *Aspects of Local Linear Complexity*. Ph.D. thesis, University of London, 1989.
- [3] D.Gollmann and W.G.Chambers, *Clock Controlled Shift Registers: A Review*, IEEE JSAC, Vol.7, No.4, pp.525–533, 1989
- [4] A.H. Chan, R.A. Games and E.L. Key. On the complexities of de Bruijn sequences. *Journal of Combinatorial Theory, Series A*, **33**:233–246, 1982.
- [5] T. Etzion. Constructions for Perfect Maps and pseudo-random arrays. *IEEE Transactions on Information Theory*, **IT-34**:1308–1316, 1988.
- [6] T. Etzion and A. Lempel. Construction of de Bruijn sequences of minimal complexity. *IEEE Transactions on Information Theory*, **IT-30**:705–709, 1984.
- [7] C.T. Fan, S.M. Fan, S.L. Ma, and M.K. Siu. On de Bruijn arrays. *Ars Combinatoria*, **19A**:205–213, 1985.
- [8] R.A. Games. There are no de Bruijn sequences of span  $n$  with complexity  $2^{n-1} + n + 1$ . *Journal of Combinatorial Theory, Series A*, **34**:248–251, 1983.
- [9] R.A. Games and A.H. Chan. A fast algorithm for determining the complexity of a binary sequence with period  $2^n$ . *IEEE Transactions on Information Theory*, **IT-29**:144–146, 1983.
- [10] A. Lempel. On a homomorphism of the de Bruijn graph and its applications to the design of feedback shift registers. *IEEE Transactions on Computers*, **C-19**:1204–1209, 1970.
- [11] J.L. Massey. Shift register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, **IT-15**:122–127, 1969.
- [12] C.J. Mitchell and K.G. Paterson. Decoding Perfect Maps. *Designs, Codes and Cryptography*, to appear.
- [13] H. Niederreiter. The probabilistic theory of linear complexity. *Advances in Cryptology, Eurocrypt '88 LNCS* **330**:191-209 1988.
- [14] K.G. Paterson. Perfect Maps. *IEEE Transactions on Information Theory*, to appear.
- [15] K.G. Paterson. Perfect Factors in the de Bruijn graph. *Designs, Codes and Cryptography*, submitted.
- [16] M.J.B. Robshaw. Increasing the Rate of Output for  $m$ -Sequences. *Electronics Letters*, **27**:1710–1712, 1991.

- [17] M.J.B. Robshaw. *On binary sequences with certain properties*. Ph.D. thesis, University of London, 1992.
- [18] R.A. Rueppel. *Analysis and Design of Stream Ciphers*. Springer-Verlag, Berlin, 1986.
- [19] M.Z. Wang and J.L. Massey. The characterization of all binary sequences with perfect linear complexity profiles. *Presented at Eurocrypt 1986*.

# Massey's Theorem and the Golay Codes

Richard E. Blahut  
University of Illinois  
Urbana, Illinois, 61801

## Abstract

Massey's theorem is used to determine the minimum distance of the Golay codes. The same method can be used to determine the minimum distance of other cyclic codes. This suggests that Massey's theorem may be a powerful tool whose uses are not yet fully uncovered.

## I Introduction

Massey's theorem relates the length of the minimum-length linear feedback shift register needed to produce a sequence and the length of the minimum-length linear feedback shift register needed to produce a maximal proper subsequence of the sequence. The theorem appeared originally as a stepping stone to prove the properties of the Berlekamp-Massey algorithm [1]. Our purpose there is to suggest that Massey's theorem has a wider range of applications, and it should be seen in a more fundamental role. We will show how the theorem can be used to find the minimum distance of the two Golay codes by an elementary, though tedious, evaluation. The same method can be used to find the minimum distance of other cyclic codes.

A *connection polynomial*  $\Lambda(x)$  is a polynomial of the form  $\Lambda(x) = 1 + \sum_{1 \leq j \leq t} \Lambda_j x^j$ . A *linear feedback shift register* is the pair  $(\Lambda(x), t)$ . We say that the linear feedback shift register  $(\Lambda(x), t)$  generates the sequence  $(S_0, S_1, \dots, S_{r-1})$  if the recursion

$$S_i = -\sum_{j=1}^t \Lambda_j S_{i-j}$$

is satisfied for  $i = t, \dots, r-1$ . The smallest integer  $t$  for which such a recursion exists for the sequence  $S = (S_0, S_1, \dots, S_{r-1})$  is called the *linear complexity* of the sequence  $S$  and is denoted  $L(S)$ .

**Massey's Theorem** If the minimum-length linear feedback shift register that generates  $(S_0, S_1, \dots, S_{r-2})$  has length  $t$  and does not generate  $S = (S_0, S_1, \dots, S_{r-2}, S_{r-1})$ , then  $L(S) \geq \max[t, r-t]$ .

A recent, appealing proof of Massey's theorem has been formulated by Maurer and Viscardi [2]. As an example of Massey's theorem, the sequence consisting of the first one million terms of the Fibonacci sequence followed by a one cannot satisfy any linear recursion of length less than 999,999 because the first one million terms are generated by a linear feedback shift register of length two.

Massey's theorem plays an important role in cryptography and in the proof of decoding algorithms for Reed-Solomon and other BCH codes. A two-dimensional version of Massey's

theorem, formulated by Sakata [3], shows up in some recent decoding algorithms [4] for algebraic-geometry codes. In this paper, we give another use of Massey's theorem. By combining it with the linear complexity property of the Fourier transform, it will be used to lower-bound the minimum distance of cyclic codes.

The *weight* of a vector is the number of components of the vector that are nonzero. The *linear complexity property*, (see [5,6] for a proof) says that the weight of a vector (i.e., a codeword)  $c = (c_0, c_1, \dots, c_{n-1})$  is equal to the linear complexity of the cyclically repeated "Fourier transform" (or spectrum)  $C_j = \sum_{i=0}^{n-1} c_i \alpha^{ij}$ ,  $j = 0, \dots, n-1$ , where  $\alpha$  is any element of order  $n$  of the underlying field or an extension of that field. Simply compute the shortest linear feedback shift register that will cyclically generate the spectrum

$$C_j = -\sum_{k=1}^w \Lambda_k C_{((j-k))} \quad j = 0, \dots, n-1$$

(The double parentheses denote modulo  $n$ .) The smallest  $w$  for which such a recursion exists equals the weight of the codeword  $c$  and  $\Lambda(x)$  will have degree exactly equal to that  $w$ .

The spectrum  $C_j = c(\alpha^j)$  of any  $q$ -ary cyclic codeword lies in the extension field  $GF(q^m)$  and is constrained by the zeros of the generator polynomial  $g(x)$  and by the *conjugacy constraint*  $C_j^q = C_{((qj))}$  (the conjugacy constraint follows from the relationship  $(a+b)^q = a^q + b^q$  in extensions of  $GF(q)$ ). These two conditions constrain the minimum-length linear feedback shift register. Hence to determine a bound on  $w$ , we examine constraints on the shift register.

The method of proof is elementary. Assume a linear feedback shift register  $(\Lambda(x), w^*)$  with  $\deg \Lambda(x) = w^*$ . We simply write out the sequence of equations

$$C_j = -\sum_{k=1}^{w^*} \Lambda_k C_{((j-k))}$$

for  $j = w^*, w^* + 1, \dots$ . Each step in  $j$  imposes a condition on the terms of  $\Lambda(x)$ . We use these conditions step by step to determine the coefficients of  $\Lambda(x)$  until  $\Lambda(x)$  is fully determined and at some step, denoted  $j = r - 1$ , the new condition cannot be satisfied. Then Massey's theorem says that  $w \geq \max[w^*, r - w^*]$ .

## II The Golay (23,12,7) Binary Code

The Golay (23,12,7) code over  $GF(2)$  has its Fourier transform in an extension of  $GF(2)$ . The code is usually defined in terms of a generator polynomial  $g(x)$  that satisfies  $g(\alpha) = 0$  and  $g(\alpha^j) = 0$  if  $\alpha^j$  is a conjugate of  $\alpha$ . In the language of the Fourier transform, this statement becomes: The binary Golay code is the set of codewords  $c$  that satisfy  $C_j = 0$  for every  $j$  in the conjugacy class  $\{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$ . We call such a  $j$  a *spectral zero* of the code. Thus, with these spectral zeros, the spectrum of any codeword  $c$  is

$$\begin{aligned} C &= (C_0, 0, 0, 0, 0, C_5, 0, C_7, 0, 0, C_{10}, C_{11}, 0, 0, C_{14}, C_{15}, 0, C_{17}, 0, C_{19}, C_{20}, C_{21}, C_{22}) \\ &= (C_0, 0, 0, 0, 0, C_5, 0, C_5^{512}, 0, 0, C_5^2, C_5^{16}, 0, 0, C_5^{1024}, C_5^{256}, 0, C_5^8, 0, C_5^{128}, C_5^4, C_5^{64}, C_5^{32}) \end{aligned}$$

where  $C_0 \in GF(2)$  is given by  $C_0 = \sum_{i=0}^{n-1} c_i$ , and  $C_5 \in GF(2^{11})$  is given by  $C_5 = \sum_{i=0}^{n-1} c_i \alpha^{5i}$ . We can restrict  $C_5$  to be nonzero. (Otherwise the codeword is the all-zero codeword or

the all-one codeword). Because there are four consecutive zeros in the spectrum, the BCH bound implies that every nonzero codeword has weight at least five. Suppose that codeword  $c$  has weight less than seven; then it has weight 5 or 6. We will use Massey's theorem to show that these weights cannot occur, and so the binary Golay code has minimum distance at least seven.

If  $C_0 = 1$ , then the weight  $w$  of codeword  $c$  is odd (because  $C_0 = \sum_{i=0}^{n-1} c_i$  is the modulo-two sum of  $w$  ones). If  $w$  is smaller than 7, then it must be 5. Then the connection polynomial  $\Lambda(x)$  for generating the spectrum  $C$  has degree 5.

If  $C_0 = 0$ , then the weight  $w$  of codeword  $c$  is even. If  $w$  is smaller than 7, then it must be 6. Then the connection polynomial  $\Lambda(x)$  for generating  $C$  has degree 6.

We will proceed with both cases simultaneously with the linear feedback shift register  $(\Lambda^*(x), w^*)$  for  $w^* = 5$  or 6 where

$$\Lambda^*(x) = \Lambda_6 x^6 + \Lambda_5 x^5 + \Lambda_4 x^4 + \Lambda_3 x^3 + \Lambda_2 x^2 + \Lambda_1 x + 1$$

and if  $w^* = 5$ ,  $\Lambda_6 = 0$  and  $C_0 = 1$ . We trace through the steps of the recursion

$$C_j = - \sum_{k=1}^{w^*} \Lambda_k C_{((j-k))}$$

and then conclude from Massey's theorem that  $w \geq \max[w^*, r - w^*] \geq 7$ .

Because  $C_5$  is nonzero,  $C_1, C_2, C_3, C_4$  and  $C_6$  are zero, and  $\Lambda_6 C_0 = 0$ , we have

$$\begin{aligned} \text{Step 1)} \quad & C_6 = \Lambda_1 C_5 + \Lambda_2 C_4 + \Lambda_3 C_3 + \Lambda_4 C_2 + \Lambda_5 C_1 + \Lambda_6 C_0 \\ & 0 = \Lambda_1 C_5 \quad \Rightarrow \Lambda_1 = 0 \end{aligned}$$

$$\begin{aligned} \text{Step 2)} \quad & C_7 = \Lambda_2 C_5 + \Lambda_3 C_4 + \Lambda_4 C_3 + \Lambda_5 C_2 + \Lambda_6 C_1 \\ & C_7 = \Lambda_2 C_5 \quad \Rightarrow \Lambda_2 = C_7/C_5 \end{aligned}$$

$$\begin{aligned} \text{Step 3)} \quad & C_8 = \Lambda_2 C_6 + \Lambda_3 C_5 + \Lambda_4 C_4 + \Lambda_5 C_3 + \Lambda_6 C_2 \\ & 0 = \Lambda_3 C_5 \quad \Rightarrow \Lambda_3 = 0 \end{aligned}$$

$$\begin{aligned} \text{Step 4)} \quad & C_9 = \Lambda_2 C_7 + \Lambda_4 C_5 + \Lambda_5 C_4 + \Lambda_6 C_3 \\ & 0 = C_7^2/C_5 + \Lambda_4 C_5 \quad \Rightarrow \Lambda_4 = (C_7/C_5)^2 \end{aligned}$$

$$\begin{aligned} \text{Step 5)} \quad & C_{10} = \Lambda_2 C_8 + \Lambda_4 C_6 + \Lambda_5 C_5 + \Lambda_6 C_4 \\ & C_5^2 = \Lambda_5 C_5 \quad \Rightarrow \Lambda_5 = C_5 \end{aligned}$$

$$\begin{aligned} \text{Step 6)} \quad & C_{11} = \Lambda_2 C_9 + \Lambda_4 C_7 + \Lambda_5 C_6 + \Lambda_6 C_5 \\ & C_{11} = C_7^3/C_5^2 + \Lambda_6 C_5 \end{aligned}$$

At this point we proceed separately for the case with  $w^* = 5$  and the case with  $w^* = 6$ .

If  $w^* = 5$ , then  $\Lambda_6 = 0$  and the polynomial  $\Lambda(x)$  is fully determined at the end of Step 5.

$$\Lambda(x) = C_5 x^5 + (C_7/C_5)^2 x^4 + (C_7/C_5)x^2 + 1$$

Therefore to satisfy Step 6,  $C_{11} = C_7^3/C_5^2$ . Now either Step 6 is satisfied or it is not. If Step 6 is not satisfied, Massey's theorem says that

$$w \geq \max[w^*, r - w^*] = \max[5, 12 - 5] = 7$$

because  $C_1, \dots, C_{10}$  are correctly generated and  $C_{11}$  is not. On the other hand, if Step 6 is satisfied, then the next two steps are

$$\begin{aligned} \text{Step 7)} \quad & C_{12} = \Lambda_2 C_{10} + \Lambda_4 C_8 + \Lambda_5 C_7 \\ & 0 = (C_7/C_5) C_5^2 + \Lambda_5 C_7 = 0 \end{aligned}$$

$$\begin{aligned} \text{Step 8)} \quad & C_{13} = \Lambda_2 C_{11} + \Lambda_4 C_9 + \Lambda_5 C_8 \\ & 0 = \Lambda_2 C_{11} \end{aligned}$$

But  $\Lambda_2$  and  $C_{11}$  are both nonzero so the equation of Step 8 is not satisfied. Therefore

$$w \geq \max[w^*, r - w^*] = \max[5, 14 - 5] = 9$$

Thus, we conclude that if the minimum weight is odd, it is at least 7.

If  $w^* = 6$ , then Step 6 implies that  $\Lambda_6 = (C_{11}/C_5) + (C_7/C_5)^3$ , and  $\Lambda(x)$  is completely determined. The next two steps are

$$\begin{aligned} \text{Step 7)} \quad & C_{12} = \Lambda_2 C_{10} + \Lambda_4 C_8 + \Lambda_5 C_7 + \Lambda_6 C_6 \\ & 0 = (C_7/C_5) C_5^2 + C_5 C_7 = 0 \end{aligned}$$

$$\begin{aligned} \text{Step 8)} \quad & C_{13} = \Lambda_2 C_{11} + \Lambda_4 C_9 + \Lambda_5 C_8 + \Lambda_6 C_7 \\ & 0 = (C_7/C_5) C_{11} + ((C_{11}/C_5) + (C_7/C_5)^3) C_7 \\ & = (C_7/C_5)^3 C_7 \end{aligned}$$

Because the right side is nonzero, this condition cannot be satisfied; Massey's theorem then gives

$$w \geq \max[w^*, r - w^*] = \max[6, 14 - 6] = 8$$

Thus we conclude that the minimum distance is at least 7.

### III The Golay (11,6,5) Ternary Code

The Golay (11,6,5) ternary code over  $GF(3)$  has its Fourier transform in an extension of  $GF(3)$ . Every codeword has a spectral zero,  $C_j = 0$ , for every  $j$  in the conjugacy class  $\{1, 3, 9, 5, 4\}$ . The spectrum of any nonzero codeword is

$$\begin{aligned} C &= (C_0, 0, C_2, 0, 0, 0, C_6, C_7, C_8, 0, C_{10}) \\ &= (C_0, 0, C_2, 0, 0, 0, C_2^3, C_2^9, C_2^{81}, 0, C_2^{27}) \end{aligned}$$

where  $C_0 \in GF(3)$  and  $C_2 \in GF(3^5)$ . We can restrict  $C_2$  to be nonzero. (Otherwise the codeword is the all-zero codeword, the all-one codeword, or the all-two codeword.)

Because there are three consecutive spectral zeros, the BCH bound says that  $w \geq 4$ . We will use Massey's theorem to show that it must be larger than four, and so the ternary Golay code has minimum distance at least 5.

Assume that the linear feedback shift register  $(\Lambda^*(x), w^*)$ , with  $w^* = 4$  and

$$\Lambda^*(x) = \Lambda_4 x^4 + \Lambda_3 x^3 + \Lambda_2 x^2 + \Lambda_1 x + 1$$

generates an initial segment of the spectrum  $C$ . Then the recursion gives

$$\begin{aligned} \text{Step 1)} \quad C_4 &= -\Lambda_1 C_3 - \Lambda_2 C_2 - \Lambda_3 C_1 - \Lambda_4 C_0 \\ 0 &= -\Lambda_2 C_2 - \Lambda_4 C_0 \end{aligned} \Rightarrow \Lambda_4 C_0 = -\Lambda_2 C_2$$

Because  $C_2$  is nonzero, this implies that either  $C_0$  is nonzero or  $\Lambda_2$  is zero.

$$\begin{aligned} \text{Step 2)} \quad C_5 &= -\Lambda_1 C_4 - \Lambda_2 C_3 - \Lambda_3 C_2 - \Lambda_4 C_1 \\ 0 &= -\Lambda_3 C_2 \end{aligned} \Rightarrow \Lambda_3 = 0$$

$$\begin{aligned} \text{Step 3)} \quad C_6 &= -\Lambda_1 C_5 - \Lambda_2 C_4 - \Lambda_3 C_3 - \Lambda_4 C_2 \\ C_2^3 &= -\Lambda_4 C_2 \end{aligned} \Rightarrow \begin{aligned} \Lambda_4 &= -C_2^2 \\ \Lambda_2 &= C_2 C_0 \end{aligned}$$

$$\begin{aligned} \text{Step 4)} \quad C_7 &= -\Lambda_1 C_6 - \Lambda_2 C_5 - \Lambda_3 C_4 - \Lambda_4 C_3 \\ C_2^9 &= -\Lambda_1 C_2^3 \end{aligned} \Rightarrow \Lambda_1 = -C_2^6$$

The polynomial  $\Lambda(x)$  is now fully determined. It is

$$\Lambda(x) = -C_2^2 x^4 + C_2 C_0 x^2 - C_2^6 x + 1$$

Continuing the recursion gives

$$\begin{aligned} \text{Step 5)} \quad C_8 &= -\Lambda_1 C_7 - \Lambda_2 C_6 - \Lambda_3 C_5 - \Lambda_4 C_4 \\ C_2^{81} &= C_2^{15} - C_2^4 C_0 \end{aligned}$$

This is satisfied if and only if  $C_2$ , an element of  $GF(3^5)$ , is a zero of  $x^{77} - x^{11} + C_0$ . If Step 5 is not satisfied, then

$$w \geq \max[4, 9 - 4] = 5.$$

If Step 5 is satisfied, we move on to the next iteration.

$$\begin{aligned} \text{Step 6)} \quad C_9 &= -\Lambda_1 C_8 - \Lambda_2 C_7 - \Lambda_3 C_6 - \Lambda_4 C_5 \\ 0 &= C_2^6 C_2^{81} - C_2 C_2^9 C_0 \end{aligned} \Rightarrow C_0 = C_2^{77}$$

If Step 6 is not satisfied, then

$$w \geq \max[4, 10 - 5] = 6$$

But if  $C_0 = 0$ , Step 6 is not satisfied even if Step 5 is satisfied. If  $C_0$  is nonzero, then Step 5 and Step 6 can both be satisfied only if the two equations

$$C_2^{77} - C_2^{11} + C_0 = 0$$

$$C_2^{77} = C_0$$

are simultaneously satisfied. But in fields of characteristic 3 these imply that

$$C_2^{11} = 2C_0 = -C_0$$

Therefore because  $C_0 = \pm 1$ ,

$$C_2^{77} = (C_2^{11})^7 = -C_0^7 = -C_0$$

which is a contradiction. Hence either  $C_2$  fails to satisfy Step 5 or fails to satisfy Step 6. We conclude that the minimum distance is at least 5.

## References

- [1] J. L. Massey, "Shift-register synthesis and BCH decoding", *IEEE Transactions on Information Theory*, Vol. 15, pp. 122-127, 1969.
- [2] U. Maurer and R. Viscardi, *Running-Key Generators with Memory in the Nonlinear Combining Function*, Diploma thesis, ETH Swiss Federal Institute of Technology, Zurich, 1984.
- [3] K. Saints and C. Heegard, "On hyperbolic cascaded Reed-Solomon codes", *Proceedings of the Tenth International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, San Juan, Puerto Rico, 1993.
- [4] S. Sakata, "Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array", *Journal of Symbolic Computation*, Vol. 5, pp. 321-337, 1988.
- [5] R. E. Blahut, *Algebraic Methods for Signal Processing and Communication Coding*, Springer-Verlag, 1992.
- [6] T. Schaub, *A Linear Complexity Approach to Cyclic Codes*, Doctor of Technical Sciences Dissertation, ETH Swiss Federal Institute of Technology, Zurich, 1988.

# Coding for Adder Channels

Ian F. Blake

Department of Electrical and Computer Engineering  
University of Waterloo  
Waterloo, Ontario, Canada N2L 3G1

Dedicated to James L. Massey on the occasion of his 60th birthday.

## Abstract

An informal survey of certain aspects of coding for a particular multiaccess channel is given. The situation of particular interest is the two-user binary adder channel in which each user transmits a zero or a one and the channel output is the sum, as integers, of the two inputs. The known results in coding for this channel are discussed.

## I Introduction

This article gives an informal survey of certain aspects of coding for a particular multiaccess channel. The situation of interest is best described by considering the two-user case, where each user transmits a binary codeword of blocklength  $n$  over the alphabet  $\mathcal{A}_2 = \{0, 1\}$ . The channel adds the two vectors over the reals to give an  $n$ -tuple over the alphabet  $\mathcal{A}_3 = \{0, 1, 2\}$ . Thus in any coordinate position, if both codewords contain a 0, the result is a 0, if both contain a 1, the output is a 2 and if one contains a 1 and the other a 0, the result is a 1. For the  $N$ -user case, each user transmits a binary codeword and the channel outputs a vector over the alphabet  $\mathcal{A}_{N+1} = \{0, 1, 2, \dots, N\}$ . The channels are referred to as the noiseless 2BAC and NBAC respectively. Certain aspects of coding for these channels are considered. The next section briefly discusses their basic properties. Section 3 considers some upper bounds on the size of codes for the users. The following two sections discuss code construction techniques that have appeared in the literature. The article concludes with a mention of some of the many interesting relatives of the problem that have been investigated.

## II Basic Properties of the Adder Channel

Consider the following discrete memoryless channel: each input is labeled with one of  $2^N$  binary  $N$ -tuples. An input with (Hamming) weight  $i$  is connected to the output labeled  $i$ . This  $2^N$ -input,  $(N + 1)$ -output channel is readily seen to have capacity

$$\mathcal{C}_N = - \sum_{i=0}^N \frac{\binom{N}{i}}{2^N} \log_2 \frac{\binom{N}{i}}{2^N}$$

It has been shown [2] that this expression can be upper and lower bounded as follows

$$\frac{1}{2} \log_2 \pi N/2 < \mathcal{C}_N < \frac{1}{2} \log_2 \pi e N/2$$

where the upper bound is asymptotically tight [2] as observed in Table 1. Experimentally it can be shown that  $1 + (1/2)\log_2(N)$  is a much tighter lower bound. It is also possible to consider a noisy  $N$ -user adder channel where each of the  $N$  outputs is reachable from each of the  $2^N$  inputs, but this will not be pursued here.

| $N$ | $1/2 \log_2(\pi N/2)$ | Exact    | $1/2 \log_2(\pi e N/2)$ |
|-----|-----------------------|----------|-------------------------|
| 2   | .825748               | 1.500000 | 1.547095                |
| 3   | 1.118229              | 1.811278 | 1.839577                |
| 4   | 1.325748              | 2.030639 | 2.047096                |
| 5   | 1.486712              | 2.198192 | 2.208060                |
| 6   | 1.618229              | 2.333362 | 2.339577                |
| 7   | 1.729426              | 2.446640 | 2.450773                |
| 8   | 1.825748              | 2.544198 | 2.547096                |
| 9   | 1.910711              | 2.629928 | 2.632058                |
| 10  | 1.986712              | 2.706429 | 2.708060                |
| 20  | 2.486712              | 3.207723 | 3.208060                |
| 30  | 2.779193              | 3.500397 | 3.500541                |
| 50  | 3.147676              | 3.868974 | 3.869024                |
| 100 | 3.647676              | 4.369011 | 4.369024                |
| 200 | 4.147676              | 4.869021 | 4.869024                |

Table 1. Bounds and Capacity for the NBAC.

Liao [17] has shown that the capacity region for this channel is described by the inequalities:

$$0 \leq R_1 \leq 1, \quad 0 \leq R_2 \leq 1, \quad R_1 + R_2 \leq 3/2.$$

More generally, for the  $N$ -user case the  $N$ -dimensional region can be described by [2]

$$\mathcal{C}_N = \{(R_1, R_2, \dots, R_N)\}$$

where

$$\begin{aligned}
0 &\leq R_i \leq 1 \\
0 &\leq R_i + R_j \leq C_2, \quad i \neq j \\
0 &\leq R_i + R_j + R_k \leq C_3, \quad i \neq j \neq k \\
&\vdots \\
0 &\leq R_1 + \cdots + R_N \leq C_N
\end{aligned}$$

To consider coding for this noiseless 2BAC, let  $\mathcal{A}_2^n$  denote the set of binary  $n$ -tuples and let  $C_1, C_2 \subseteq \mathcal{A}_2^n$ . For  $\underline{x}, \underline{y} \in \mathcal{A}_2^n$  define  $\underline{x} + \underline{y} = (x_1 + y_1, \dots, x_n + y_n)$  with real addition. The code pair  $(C_1, C_2)$  is called *uniquely decodable* if the set

$$C_1 + C_2 = \{\underline{c}_1 + \underline{c}_2, \underline{c}_1 \in C_1, \underline{c}_2 \in C_2\}$$

contains distinct vectors over  $\mathcal{A}_3$  i.e. each possible received vector is uniquely decomposable over  $\mathcal{A}_2$ . It is worth noting that received 0's or 2's decompose uniquely and so the problem is to construct the code in such a manner that received 1's can be resolved.

Define the distance function on  $\mathcal{A}_3$  by

$$d_3(x, y) = |x - y| \quad x, y \in \mathcal{A}_3$$

and, by extension,

$$d_3(\underline{x}, \underline{y}) = \sum_{i=1}^n d_3(x_i, y_i), \quad \underline{x}, \underline{y} \in \mathcal{A}_3^n.$$

The minimum distance of the code pair  $(C_1, C_2)$  is then

$$\delta = \min\{d_3(\underline{c}_1 + \underline{c}_2, \underline{c}_1' + \underline{c}_2') | \underline{c}_i, \underline{c}_i' \in C_i, i = 1, 2\}$$

and the code is uniquely decodable if  $\delta \geq 1$ .

For the noisy 2BAC mentioned earlier, it is straightforward to show that the code pair  $(C_1, C_2)$  can correct up to  $\lfloor (\delta - 1)/2 \rfloor$  errors if it has minimum distance at least  $\delta$ . In this case it is easy to show that both codes  $C_1$  and  $C_2$  require minimum Hamming distances of at least  $\delta$ .

The primary interest of this report is on constructive aspects of uniquely decodable codes for the noiseless 2BAC, although some results for the NBAC will be mentioned. It is assumed the users are both word-synchronized and bit-synchronized. For a code pair  $(C_1, C_2)$ , for the 2BAC, the rate sum is

$$\frac{1}{n} \log_2 |\mathcal{C}_1||\mathcal{C}_2| = R_1 + R_2$$

where  $|\cdot|$  indicates set cardinality, and the objective is to find codes  $C_1$  and  $C_2$  for which  $R_1 + R_2$  is maximized. From the previous observation,  $R_1 + R_2 \leq 3/2$ . Notice that the channel can be time shared, giving simple code pairs that trivially achieve the rate sum  $R_1 + R_2 = 1$ . Thus interest is confined to the region  $1 \leq R_1 + R_2 \leq 3/2$ . The next section discusses an upper bound to  $|\mathcal{C}_1||\mathcal{C}_2|$  (equivalently  $R_1 + R_2$ ) due to van Tilborg [28], and its extension to the NBAC [1].

### III An Upper Bound on the Size of Code Dictionaries

The following is a purely combinatorial approach to upper bounding  $M(n, \delta)$ , defined as the maximum of  $|\mathcal{C}_1||\mathcal{C}_2|$  over all code pairs of blocklength  $n$  with minimum distance  $\delta$ . Let

$$W_k(n) = \{\underline{u} | \underline{u} \in \mathcal{A}_3^n, \underline{u} \text{ has exactly } k \text{ coordinates equal to 1}\}$$

and

$$D_k = |(\mathcal{C}_1 + \mathcal{C}_2) \cap W_k(n)|$$

and note that  $\sum D_k = |\mathcal{C}_1||\mathcal{C}_2|$ . Inequalities for the numbers are derived as:

$$\begin{aligned} D_k &+ \binom{k+1}{k} D_{k+1} + \binom{k+2}{k} D_{k+2} + \cdots + \binom{l}{k} D_l \\ &\leq \binom{n}{k} A(k, \lfloor \frac{\delta+1-2(l-k)}{2} \rfloor) \end{aligned}$$

where  $0 \leq k < l \leq n$ ,  $2(l-k) \leq \delta - 1$  and  $A(n, d)$  is the maximum size of a binary code of blocklength  $n$  and Hamming distance  $d$ .

For uniquely decodable codes ( $\delta = 1$ ), the above inequalities yield:

$$|\mathcal{C}_1||\mathcal{C}_2| \leq \begin{cases} 2 \sum_{k=0}^m \binom{2m+1}{k} 2^k & n = 2m + 1 \\ 2 \sum_{k=0}^m \binom{2m+2}{k} 2^k + \binom{2m+2}{m+1} 2^{m+1} & n = 2m + 2. \end{cases}$$

It is also shown that, asymptotically, these upper constructive bounds are tight in the sense that

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log_2 |\mathcal{C}_1||\mathcal{C}_2| = \frac{3}{2}.$$

These results extend to the NBAC using similar techniques [1]. The general results for the NBAC, each user with a binary code of blocklength  $n$ , are:

$$\begin{aligned} |\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdots \cdot |\mathcal{C}_N| &\leq \\ \sum_{\substack{k_1, k_2, \dots, k_{N-1} \\ 0 \leq k_1 + k_2 + \dots + k_{N-1} \leq n}} &\binom{n}{k_1} \binom{n - k_1}{k_2} \cdots \binom{n - \sum_{i=1}^{N-1} k_i}{k_{N-1}} \cdot \\ \min \left\{ \max (2^{k_1}, 2^{k_2}, \dots, 2^{k_{N-1}}), 2^{(n - \sum_{i=1}^{N-1} k_i)} \right\} \end{aligned}$$

The upper bound is iterative i.e., the  $(N-1)$ -user bound can be obtained by projecting the right side of the above equation on a subspace of  $(N-1)$  combinatorial variables (e.g. by setting  $k_{N-1} = 0$ ). Thus, for example, for  $N = 3$  the bound admits the form

$$|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot |\mathcal{C}_3| \leq \sum_{\substack{k, l \\ 0 \leq k+l \leq n}} \binom{n}{k} \binom{n - k}{l} \min \left\{ \max (2^k, 2^l), 2^{n-(k+l)} \right\}$$

which yields van Tilborg's result for the  $N = 2$  case. An asymptotic analysis on this expression shows that the limit as  $N \rightarrow \infty$  is approximately  $1 + \frac{1}{2} \log_2(N)$ , which is close to the capacity bounds noted earlier.

It is of interest that the combinatorial techniques used in the derivation of these bounds essentially achieve, asymptotically, the channel capacity. Presumably the arguments used will indicate the properties required of code sets to come close to achieving capacity.

## IV Block Code Construction Techniques

A wide variety of construction techniques for uniquely decodable binary block codes for the 2BAC and NBAC have been considered in the literature. Only a few of these that are either of particular interest or are easy to describe are noted here.

Kasami and Lin [14] show that if  $\mathcal{C}_1$  is chosen as a linear  $(n, k)$  code, then *exactly* two vectors from each coset of  $\mathcal{C}_1$  (but only  $\underline{0}$  from  $\mathcal{C}_1$  itself) can be chosen for  $\mathcal{C}_2$  so that  $(\mathcal{C}_1, \mathcal{C}_2)$  is uniquely decodable. Thus  $|\mathcal{C}_1||\mathcal{C}_2| = 2^k(2(2^{n-k} - 1) + 1)$ . For example, if  $\mathcal{C}_1$  is the  $(7, 4)$  Hamming code, then  $|\mathcal{C}_2| = 2(8-1)+1 = 15$  giving a rate sum of  $(1/7)\log_2(16 \cdot 15) \simeq 1.129$ .

The following two constructions of Weldon [31] are of interest. First let  $\mathcal{C}_1 = \{\underline{0}, \underline{1}\}$ , the all-zero and all-one vector, respectively. Let  $\mathcal{C}_2 = \mathcal{A}_2^n \setminus \{\underline{1}\}$  be the set of all binary words of blocklength  $n$  except the all-one word. The unique decodability of  $(\mathcal{C}_1, \mathcal{C}_2)$  follows by inspection.

A second construction [31] assumes that  $\mathcal{C}_1$  is a linear  $(n, k_1)$  code. Code  $\mathcal{C}_2$  will be the set of all binary  $n$ -tuples,  $\mathcal{A}_2^n$ , with certain words removed. For  $\underline{c}_1, \underline{c}'_1 \in \mathcal{C}_1$ ,  $\underline{u}$  and  $\underline{v}$  are said to be *linked* by  $\underline{c}_1$  and  $\underline{c}'_1$  if  $\underline{c}_1 - \underline{c}'_1 = \underline{u} - \underline{v}$ . Clearly then both  $\underline{u}$  and  $\underline{v}$  cannot be in  $\mathcal{C}_2$  for  $(\mathcal{C}_1, \mathcal{C}_2)$  to be a uniquely decodable code pair. Then  $\mathcal{C}_2$  is  $\mathcal{A}_2^n$  with one vector of every pair of vectors that are linked by a pair of words in  $\mathcal{C}_1$  removed. If  $\mathcal{C}_1$  has  $A_i$  codewords of Hamming weight  $i$  then it is shown that  $\mathcal{C}_2$  can be chosen so that

$$|\mathcal{C}_2| \geq 2^n - 2^{n+k_1} \sum_{i=d}^n A_i 2^{-i}$$

by this expurgation process.

Van Tilborg [29] considers the question of how large  $\mathcal{C}_2$  can be for a given  $\mathcal{C}_1$  by using the notions of distance distribution [19] of not necessarily linear codes. Let  $Y_k$  denote the set of binary  $n$ -tuples of weight  $k$  and define the characteristic numbers of  $\mathcal{C}$  by

$$B_k = \frac{1}{|\mathcal{C}|^2} \sum_{\underline{u} \in Y_k} |\sum_{\underline{c} \in \mathcal{C}} (-1)^{(\underline{u}_1 c_1 + \dots + \underline{u}_n c_n)}|^2$$

which satisfy the equation

$$B_k = \frac{1}{|\mathcal{C}|} \sum_{l=0}^n A_l P_k(n, l)$$

where  $A_l$  is the  $l$ th coefficient of the distance enumerator of  $\mathcal{C}$  and  $P_l(n, x)$  is a Krawtchouk polynomial. The annihilator polynomial of the code  $\mathcal{C}$ , with characteristic numbers  $B_k$ ,  $\alpha(x)$ , has roots for values of  $B_k$  that are nonzero. The expansion of this polynomial in terms of Krawtchouk polynomials is

$$\alpha(x) = \sum_{k=0}^r \alpha_k P_k(n, x).$$

It is shown that if code  $\mathcal{C}_1$  has annihilator polynomial  $\alpha(x)$  and if  $(\mathcal{C}_1, \mathcal{C}_2)$  is uniquely decodable then

$$|\mathcal{C}_2| \leq \sum_{k=0}^r \max(0, \alpha_k) \binom{n}{k} 2^{\min(k, n-k)}.$$

The theorem can be used effectively to reduce the search for good codes after  $\mathcal{C}_1$  is chosen. A demonstration of this is given in the paper by constructing a code pair with rate pair  $(R_1, R_2) = (.5170, .7814)$ .

Coebergh van den Braak and van Tilborg [6] demonstrate another construction. Let  $\mathcal{Z}$  be a binary code of blocklength  $s$  and  $n$  an arbitrary positive integer. Let  $\mathcal{C}$  be the binary code of blocklength  $ns$  obtained by repeating each symbol of each codeword of  $\mathcal{Z}$   $n$  times i.e.  $|\mathcal{C}| = |\mathcal{Z}|$  and

$$\mathcal{C} = \{(\underline{c}_1, \dots, \underline{c}_s, \underline{c}_i = z_i \underline{1}, \underline{z} = (z_1, \dots, z_s)) \in \mathcal{Z}\}$$

To construct a code  $\mathcal{D}$  so that  $(\mathcal{C}, \mathcal{D})$  is uniquely decodable consider

$$\underline{r} = \underline{c} + \underline{d} = (\underline{c}_1 + \underline{d}_1, \dots, \underline{c}_n + \underline{d}_n), \quad \underline{c}_i = z_i \underline{1}$$

and note that if  $\underline{c}_i + \underline{d}_i \neq \underline{1}$  then  $\underline{c}_i$  and  $\underline{d}_i$  can be uniquely determined. If  $\underline{c}_i + \underline{d}_i = \underline{1}$  then  $(\underline{c}_i, \underline{d}_i)$  is either  $(\underline{0}, \underline{1})$  or  $(\underline{1}, \underline{0})$ . The structure of the code  $\mathcal{Z}$  is then used to resolve the ambiguity. Actually the construction is more involved, replacing 0's and 1's of the codewords of  $\mathcal{Z}$  by codewords of carefully chosen codes  $\mathcal{C}_0$  and  $\mathcal{C}_1$  respectively.

Kasami et. al [16] use graphical techniques to construct  $\delta$ -decodable codes by associating a graph with a given code  $\mathcal{C}_1$ . The vertices of the graph are labeled with the set of binary  $n$ -tuples with two vertices being adjacent according to a property of the code. A set of vertices in the graph is independent if no two are independent. It is shown that a subset  $\mathcal{C}_2 \subset \mathcal{A}_2^n$  gives  $(\mathcal{C}_1, \mathcal{C}_2)$   $\delta$ -decodable if and only if  $\mathcal{C}_2$  is an independent set of the graph. Thus a maximal independent subset gives the optimum rate sum by this construction. The same technique is specialized in [10] for the noiseless 2BAC.

The above constructions give the flavor of some of the interesting approaches to the problem that have been taken in the literature. It would be of great interest to derive a simple combinatorial construction technique that specifically addresses the characteristics required and that asymptotically achieves the maximum rate sum. None of the constructions discussed in the literature appear to do this or to rule out the possibility of such a construction. The problem does not seem to be on the same order of difficulty as constructing random error correcting codes to exceed the Varshamov-Gilbert bound and it is not yet clear that the essence of the combinatorial character of the problem has been captured by these approaches.

## V Code-Combining Construction Techniques

Many of the criteria for a code pair to be uniquely decodable have been used to give insight to the construction of good codes. This section continues in this line by briefly considering constructions that use two uniquely decodable code pairs to give a third uniquely decodable code pair.

For the NBAC, if  $(R_1, R_2, \dots, R_N)$  and  $(R'_1, R'_2, \dots, R'_N)$  are sets of achievable code rates for the NBAC, of the same length, then the line between these also yield achievable code rates. This is the simple time-sharing argument that shows the convexity of the capacity region for the NBAC. A variant of this scheme, due to Shannon [25], [31], time shares a code set of blocklength  $n$ , with code rate set  $(R_1, R_2, \dots, R_N)$  and a code set of

blocklength  $n'$  with rates  $(R_1', R_2', \dots, R_{N'}')$  to yield a code of blocklength  $an + a'n'$ , for positive integers  $a$  and  $a'$  and code rates

$$\left( \frac{anR_1 + a'n'R_1'}{an + a'n'}, \dots, \frac{anR_N + a'n'R_N'}{an + a'n'} \right).$$

To achieve this code rate set, the users agree to use code set 1 for  $a$  times, then to use code set 2, for  $a'$  times.

These constructions cannot move “closer” to the capacity bound than the original codes and thus are of limited interest in achieving this particular goal. A more elegant and potentially extremely useful construction is due to Jim Massey, who once again appears to have gone to the heart of a matter [20].

The Massey Theorem: Let  $\mathcal{C}_1$  and  $\mathcal{C}_2$  be binary codes of blocklength  $n$  and constant weights  $w_1$  and  $w_2$  and minimum Hamming distances  $d_1$  and  $d_2$  respectively. Let

$$D_{\min} = \min\{d_H(\underline{x}, \underline{y}) , \underline{x} \in \mathcal{C}_1, \underline{y} \in \mathcal{C}_2\}, \quad D_{\max} = \max\{d_H(\underline{x}, \underline{y}) , \underline{x} \in \mathcal{C}_1, \underline{y} \in \mathcal{C}_2\}.$$

Then  $\max\{d_1, d_2\} + D_{\min} > D_{\max}$  is a sufficient condition for the code pair  $(\mathcal{C}_1, \mathcal{C}_2)$  to be uniquely decodable for the 2BAC.

The theorem gives a simple criteria for determining the unique decodability of a given code pair. While it is not a necessary condition, it nonetheless seems to give interesting information as to what should be sought after when constructing uniquely decodable code pairs. Combined with the following lemma, interesting approaches are opened up.

Blowing up lemma (Massey [20]): If  $\mathcal{C}_i$  is a binary blocklength  $n$  code with constant weight  $w_i$ ,  $i = 1, 2, 3, 4$  with  $w_1 < w_3$ ,  $w_2 < w_4$  such that  $(\mathcal{C}_1, \mathcal{C}_2)$ ,  $(\mathcal{C}_3, \mathcal{C}_4)$ ,  $(\mathcal{C}_1, \mathcal{C}_4)$ ,  $(\mathcal{C}_2, \mathcal{C}_3)$  are all uniquely decodable code pairs, then

$$w_1 + w_4 \neq w_2 + w_3$$

is a sufficient condition for  $(\mathcal{C}_1 \cup \mathcal{C}_3, \mathcal{C}_2 \cup \mathcal{C}_4)$  to be uniquely decodable.

The theorem and lemma are illustrated with an example, also due to Massey:

Example (Massey [20]):

Consider the four codes

$$\begin{aligned} \mathcal{C}_1 &= \{1100, 0011, 1001, 0110\}, \quad d_1 = 2, \quad w_1 = 2 \\ \mathcal{C}_2 &= \{0101, 1010\}, \quad d_2 = 4, \quad w_2 = 2 \\ \mathcal{C}_3 &= \{1110, 1101, 1011, 0111\}, \quad d_3 = 2, \quad w_3 = 3 \\ \mathcal{C}_4 &= \{1111\}, \quad d_4 = \infty (\text{by convention}), \quad w_4 = 4 \end{aligned}$$

The four required code pairs noted in the lemma, are easily seen to be uniquely decodable either by using the theorem or by inspection. Consequently the code pair  $(\mathcal{C}_1 \cup \mathcal{C}_3, \mathcal{C}_2 \cup \mathcal{C}_4)$  is uniquely decodable. Notice that the rate sum of the code pair  $(\mathcal{C}_1, \mathcal{C}_2)$  is .75 while the rate sum of  $(\mathcal{C}_1 \cup \mathcal{C}_3, \mathcal{C}_2 \cup \mathcal{C}_4)$  is  $(1/4)\log_2(24) \simeq 1.146$ . Clearly the blowing up lemma can help in our quest for good code pairs.

## VI Comments

This note has considered only a special case of the construction of binary block codes for the 2BAC. Codes for the noiseless NBAC have also been considered (e.g. [4], [9]). Numerous other related situations have been investigated in the literature and some of these are briefly noted. Although mention has not been made here, many of the references cited also consider the construction of error correcting codes for the noisy 2BAC and NBAC. In addition to block codes, both trellis and convolutional codes have been considered for these channels, for both the noiseless and noisy case and for the synchronous, quasi-synchronous (there is an offset between codewords transmitted by the users, but the offsets are known to the decoder) and the asynchronous cases (e.g. [5], [7], [11], [18], [22]).

Chang and Wolf [3] consider code construction for a case where there are  $N$  users, each with an  $M$ -ary input alphabet (actually representing frequencies). Two cases are studied, one where the channel output yields the different symbols that appeared among the  $N$  channel inputs at that time, but not the number of times each such symbol appeared (without intensity) and the other when this information is provided. For  $M$  equal to 2, this last situation is the NBAC.

The case where there is a population of  $N$  users but at most  $T$  are allowed to transmit at any particular time, has been considered [12], [21]. The constraint changes the code construction significantly. Mathys [21] also considers the case where the inputs may be real numbers (on an adder channel) which adds another intriguing aspect to the problem.

The binary multiplying channel (BMC) for a two-user case, outputs a binary symbol  $y = x_1 \wedge x_2$  to two binary inputs. Considerations for this channel are quite different than for the 2BAC and the literature on it is now quite extensive (e.g. [23], [24], [26]). Similarly the binary switching channel has also been of interest [27]. The channel output to the binary channel inputs  $x_1$  and  $x_2$  is  $y = x_1/x_2$ , giving a ternary output alphabet  $\{0, 1, \infty\}$  where  $x/0 = \infty, x \in \{0, 1\}$ . Although the channel may appear to have some similar structural characteristics, its behavior is quite different. For example, Vanroose [27] displays a code with a rate sum of 1.58496.

Two rather different situations are noted. In [32] the construction of codes for the 2BAC is considered wherein either one (partial feedback) or both (full feedback) of the coders observe the previous channel outputs for that codeword. A class of codes based on Fibonacci sequences is constructed that asymptotically achieve the rates  $R_1 = R_2 = \log_2((1 + \sqrt{5})/2)$  for the partial feedback case. Ericson [8] discusses a noncooperative binary adder channel where a transmitter sends a binary codeword and a jammer adds a binary codeword  $\underline{s}$ , chosen with knowledge of the sender's codebook, to give a channel output  $\underline{y} = \underline{x} + \underline{s}$ . Random codes are considered for this situation.

Many of the situations noted here represent approximations to real channels and provide fertile ground for further study.

## References

- [1] S. Bross and I. F. Blake, "Upper bound for uniquely decodable codes in a binary input  $N$ -user adder channel," *International Symposium on Information Theory*, San Antonio, Texas, January, 1993.

- [2] S.-C. Chang and E.J. Weldon, Jr., " Coding for  $T$ -user multiple-access channels," *IEEE Trans. Inf. Theory*, vol. IT-25, 684-691, 1979.
- [3] S.-C. Chang and J.K. Wolf, "On the T-user M-frequency noiseless multiple-access channel with and without intensity information", *IEEE Trans. Inf. Theory*, vol. IT-27, 41-48, 1981.
- [4] S.-C. Chang, "Further results on coding for the T-user multiple-access channels," *IEEE Trans. Inf. Theory*, vol. IT-30, 411-415, 1984.
- [5] P. Chevillat, " $N$ -User trellis coding for a class of multiple-access channels", *IEEE Trans. Inf. Theory*, vol. IT-27, 114-120, 1981.
- [6] P.A.B.M. Coebergh van den Braak and H.C.A. van Tilborg, "A family of good uniquely decodable code pairs for the two-access binary adder channel", *IEEE Trans, Inf. Theory*, vol. IT-31, 3-9, 1985.
- [7] M.A. Deaett and J.K. Wolf, "Some very simple codes for the nonsynchronized two-user multiple-access adder channel with binary inputs," *IEEE Trans, Inf. Theory*, vol. IT-24, 635-636, 1978.
- [8] T. Ericson, "The noncooperative binary adder channel," *IEEE Trans. Inf. Theory*, vol. IT-32, 365-374, 1986.
- [9] T.J. Ferguson, "Generalized T-user codes for multiple-access channels," *IEEE Trans, Inf. Theory*, vol. IT-31, 775-778, 1985.
- [10] F. Guo and Y. Watanabe, "Graph-theoretical construction and uniquely decodable code pair for the two-user binary adder channel", *IEICE Trans. Fundamentals*, vol. E75-A, 492-497, 1992.
- [11] L. Györfi and I. Kerekes, "A block code for noiseless asynchronous multiple access OR channel", *IEEE Trans, Inf. Theory*, vol. IT-27, 788-791, 1981.
- [12] D.B. Jevtić, "Disjoint uniquely decodable codebooks for noiseless synchronized multiple-access adder channels generated by integer sets", *IEEE Trans. Inf. Theory*, vol. IT-38, 1142-1146, 1992.
- [13] T. Kasami, S. Lin, and S. Yamamura, "Further results on coding for a multiple-access channel", *Colloq. Math. Soc. Janos Bolyai*, vol. 16, Topics in Information Theory, Keszthely, 369-392, 1975.
- [14] T. Kasami and S. Lin, "Coding for a multiple-access channel", *IEEE Trans. Inf. Theory*, vol. IT-22, 129-137, 1976.
- [15] T. Kasami and S. Lin, "Bounds on the achievable rates for a memoryless multiple-access channel", *IEEE Trans. Inf. Theory*, vol. IT-24, 187-197, 1978.
- [16] T. Kasami, S. Lin, V.K. Wei and S. Yamamura, "Graph theoretic approaches to the code construction for the two-user multiple-access binary adder channel", *IEEE Trans. Inf. Theory*, vol. IT-29, 114-130, 1983.
- [17] H. Liao, *Multiple Access Channels*, Ph.D. Dissertation, Department of Electrical Engineering, University of Hawaii, 1972.
- [18] S. Lin and V.K. Wei, "Nonhomogeneous trellis codes for the quasi-synchronous multiple-access binary adder channel with two users", *IEEE Trans. Inf. Theory*, vol. IT-32, 787-796, 1986.
- [19] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error Correcting Codes*, North-Holland Publishing Co., Amsterdam, 1977.
- [20] J. L. Massey, "On codes for the two-user binary adder channel", Oberwolfach, April, 1992.

- [21] P. Mathys, "A class of codes for a  $T$  active users out of  $N$  multiple-access communication system", *IEEE Trans. Inf. Theory*, vol. IT-36, 1206-1219, 1990.
- [22] R. Peterson and D. Costello, "Binary convolutional codes for a multiple-access channel", *IEEE Trans. Inf. Theory*, vol. IT-25, 101-105, 1979.
- [23] J.P. Schwalwijk, "The binary multiplying channel - a coding scheme that operates beyond Shannon's inner bound region", *IEEE Trans. Inf. Theory*, vol. IT-28, 107-110, 1982.
- [24] J.P. Schwalwijk, "On an extension of an achievable rate region for the binary multiplying channel", *IEEE Trans. Inf. Theory*, vol. IT-29, 445-448, 1983.
- [25] C.E. Shannon, "Two-way communication channels", in *Proc. 4th Berkeley Symp. Math. Stat. Prob.*, vol. 1, 611-644, 1961.
- [26] W.M.C.J. van Overveld, "Fixed-length strategies for the binary multiplying channel", *IEEE Trans. Inf. Theory*, vol. IT-34, 314-318, 1988.
- [27] P. Vanroose, "Code construction for the noiseless binary switching multiple-access channel", *IEEE Trans. Inf. Theory*, vol. IT-34, 1100-1106, 1988.
- [28] H.C.A. van Tilborg, "An upper bound for codes in an two-access binary erasure channel", *IEEE Trans. Inf. Theory*, vol. IT-24, 112-116, 1978.
- [29] H.C.A. van Tilborg, "Upper bounds on  $|C_2|$  for a uniquely decodable code pair  $(C_1, C_2)$  for a two-access binary adder channel", *IEEE Trans. Inf. Theory*, vol. IT-29, 386-389, 1983.
- [30] H.C.A. van Tilborg, "An upper bound for the noisy two-access binary adder channel", *IEEE Trans. Inf. Theory*, vol. IT-32, 436-440, 1986.
- [31] E.J. Weldon, Jr., "Coding for a multiple-access channel", *Information and Control*, vol. 36, 256-274, 1978.
- [32] Z. Zhang, T. Berger, and J. L. Massey, "Some families of zero-error block codes for the two-user binary adder channel with feedback", *IEEE Trans. Inf. Theory*, vol. IT-33, 613-619, 1987.

# Using Redundancy to Speed up Disk Arrays

David L. Cohn  
University of Notre Dame  
Notre Dame, Indiana, USA

Robert L. Stevenson  
University of Notre Dame  
Notre Dame, Indiana, USA

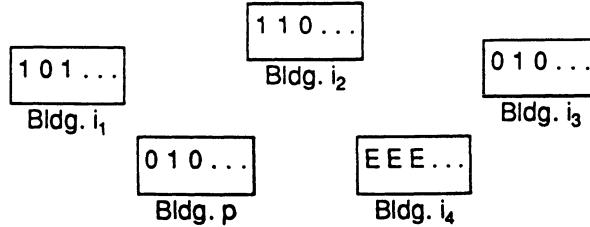
## Abstract

The rate of performance improvement for disk subsystems has been substantially below that for other parts of computer systems. In particular, access speeds have been increasing at one-fifth the rate of processor speeds and less than one half the rate of raw transfer bandwidth. Arrays of disk drives provide parallel data paths to effectively increase bandwidth. Redundancy is used in such arrays to mitigate the increased failure probabilities. This paper shows that scattering redundant blocks throughout the array can provide better average access time statistics. The redundant blocks require significant disk capacity sacrifices and impose a computational burden. However, if current technology trends continue, these costs will be substantially less than the value of improved access times.

## I Introduction

Although information theory is just that, a theory, its application to practical problems has always intrigued information theorists. Now, of course, it is widely used to solve communication problems and for home entertainment. In 1963, James L. Massey was given the "Best Tutorial Paper" award by the National Electronics Conference for a paper entitled "Error-Correcting Codes Applied to Computer Technology." In his 1976 inaugural address as the Freiman Professor of Electrical Engineering at the University of Notre Dame, [9] Professor Massey went so far as to propose applying information theory to disaster recovery:

"This application occurred to me recently while reflecting upon the tremendous chaos that resulted from a fire which destroyed a large number of military personnel records at a government facility near Kansas City. One way to avoid such a loss is, of course, to keep duplicate records in another location. But this doubles the cost of record keeping! Suppose, however, that one had four separate buildings where records were stored (the information buildings). One could add a single extra building (the parity building) each of whose stored bits was chosen to make the number of 1's even in the five bits at corresponding position in all the buildings. A fire, or other catastrophe, destroying any one of the five buildings, say  $i_4$ , would leave us with the situation shown in the following figure:



Knowing that the number of 1's at each position in all five buildings had been even, we could now use the records in the four surviving buildings to reconstruct the destroyed records as 0 1 1 . . . Our cost for safeguarding the records in this scheme is only 25% more than for storing the data with no protection at all."

Massey's comment may have been made in jest, but it presaged the use of redundancy to improve the reliability of inexpensive disk drives in computer systems. Today's RAID systems, for Redundant Arrays of Inexpensive Disks, are essentially spinning realizations of Massey's record storage buildings. The simple addition of parity prevents data loss even if one of these inexpensive, and presumably less reliable, disk drives fails. The systems are selling well, so the marketplace is saying that the improved reliability is worth the loss of capacity and *performance* necessary to realize RAID technology.

However, as all information theorists know, redundancy is a remarkable tool. It can be used to do more than just mitigate the loss of a single disk drive. In this paper, we argue that it can actually be used to improve the performance of that array of disk drives! This note shows that the powerful redundancy of Reed-Solomon codes can substantially reduce the access time of data on disks, albeit with a substantial loss in useful capacity. However, disk capacity is growing at a rapid rate, and access time is declining only slowly. Thus, it is possible that this tradeoff will eventually become commercially attractive.

## II Raid Systems

In a recent survey [4] of RAID systems [5], it was noted that the improvement rate for microprocessor performance has been 50% per year [11], but for transfer bandwidth (and storage capacity) it is just over 20%, and for disk access times less than 10%. Even today, this mismatch is making the speed of the disk subsystem a key factor in overall computer system performance. If the trends continue, and they give every indication of doing so, disk access will become a critical bottleneck.

The time it takes to access data on a disk is primarily determined by three factors:

- *Seek time* – time to move the head to correct radial position.
- *Rotational latency* – time for desired sector to appear under head.
- *Data transfer rate* – number of bytes/second that the head can read.

Typical modern disks have seek times and rotational latencies on the order of 10-15 milliseconds and data transfer rates of a few Mbytes/second. Thus, for small files, say a thousand bytes, seek time and rotational latency dominate; for large files, the transfer time can be much longer.

The data transfer delay can be mitigated by *striping* data across an array of disk drives. A portion, or *block*, of each file is concurrently written on, or read from, each disk. This provides parallel data paths, effectively giving a linear increase in data transfer rate. Unfortunately, it also increases disk subsystem failure rates. More disks mean more possible failures. Just as with communication systems, redundancy is used to tolerate these failures. Fortunately, increases in disk capacity mitigate the attendant loss of information storage capability.

A key paper on RAID [10] categorizes five different approaches, or levels, to using redundancy for disk reliability. These include simple replication [1] (Level 1), the use of Hamming codes to identify and correct faulty disks (Level 2), and, much as Massey proposed, a single parity check across all of the disks (Levels 3-5). The latter three methods differ in how data is distributed across the disks and where parity is stored. Level 3 has a block size, or *striping grain*, of a single byte, so even small files are written to all disks. Levels 4 and 5 allow larger block sizes, and small files will be written to only some disks. Level 5 adds the improvement of spreading the parity data over all disks, eliminating “hot spots” [8].

Recently, *two* additional levels of RAID have been defined. Level 0 is simply an array of disks with no redundancy and is used as a reference. Level 6 uses a Reed-Solomon code to generate two parity blocks so it can accommodate double disk failures. In each case, the redundant blocks are computed for a stripe across some or all of the disks.

As is typical with real systems, there are many subtle tradeoffs in determining the optimal RAID system. File sizes and workload nature make a substantial difference and the problem is further complicated by multiple simultaneous disk transfers. Simulation is frequently used to evaluate the options and to provide guidelines for practical realizations [3], [4], [2]. The key issues are:

- *Reliability* – How well does the scheme tolerate disk failures and other faults?
- *Performance* – How slow is the scheme relative to Level 0 for reads and writes?
- *Cost* – How much additional disk capacity is needed to support the scheme?

In general, Levels 5 and 6 are the most flexible, but certain workloads work best with the wide data distribution of Level 3.

### III The New Scheme

Now we will consider a different purpose for redundancy. We will use it to speed up the process of retrieving data from the disk array at the cost of slower writes, a reduction in disk capacity and complex reading and writing algorithms. However, as we argue later, current technology trends should eventually make these costs quite reasonable. In this approach, rather than preserving the parity of stripe of data across a set of disks, redundancy will be computed for files.

We will first explain the scheme for a simple example and then show that the simplifications are not critical. Our example file will be  $k$  blocks long, where  $k$  is something less than the number of disks in the array. We will use a Reed-Solomon [12] code to compute an  $n - k$  redundant blocks were  $n$  is no more than the number of disks. When the file is written, all  $n$  blocks will be scattered across the disks, no more than one block per disk. Unlike normal RAID, the blocks will *not* be written in one stripe. Rather, they will be independently placed on their respective disks.

From the properties of Reed-Solomon codes, any  $k$  of the blocks can be used to reconstruct the file. Therefore, when the file is to be retrieved, any  $k$  of the  $n$  blocks can be read from the disk array. A file read, then, consists of asking each disk to find its block and using the first  $k$  that are found. The time it takes to read a file is thus the seek time and rotational latency of the  $k^{\text{th}}$  fastest of  $n$  retrievals, plus the data transfer time of one block, plus the computational time to reconstruct the file. As we show in the next section, proper selection of  $k$  will provide a meaningful reduction in file access time.

Of course, this speed-up is not without its costs. The use of redundant blocks adds the following penalties:

- Calculating the redundant blocks is computationally intensive.
- Reconstructing the file from the fastest  $k$  blocks is relatively complex.
- Saving the additional blocks slows down the writing process.
- The redundancy reduces the information capacity of the disk array.

Fortunately, technology trends and careful design will mitigate these disadvantages. The continuing improvement in compute power should make the first two of little consequence. Indeed, for this analysis, we ignore them. For many computer systems, disk usage is quite bursty and the additional writes can be delayed until a period of inactivity. The loss of capacity is quite significant, but technology is pushing disk sizes up fast, and the performance improvement may be worth it.

## IV Performance

We will compute the performance of the new scheme relative to RAID Level 0, that is, a disk array with no redundancy. (Chen, et al. [4] argue that Level 1 RAID, which stores the original data plus a copy, can be faster than Level 0 since the “closest” replica can be read. Of course, Level 1 RAID can be viewed as a simple case of the new scheme with  $k = 1$  and  $n = 2$ .) For Level 0 RAID, the  $k$  blocks of our example file will each be written to one of the disks; retrieving the file means reading all  $k$  blocks.

The time it takes to read  $k$  blocks from  $k$  disks depends on whether the disks are *synchronized*. In some RAID realizations, electronic means are used to assure rotational synchronization. (This is particularly valuable for Level 3 RAID, since each transaction will involve all disks.) Further, if each disk is always told to fetch data from the same physical location, the heads will also be synchronized. For such arrays, all of the blocks for a given file are placed at the same physical location on their respective disks. Thus, for a synchronized disk array, all blocks of our file would be found simultaneously. If  $X$  represents the sum of the seek time and rotational latency,  $T$  the time to transfer one block, and  $R_{\text{sync}}$  the file’s retrieval time with synchronized disks, then the expected value of  $R_{\text{sync}}$

will be

$$\bar{R}_{sync} = \bar{X} + T$$

For other arrays, the disks are *not* synchronized; following [7], we call such arrays *asynchronous*. Typically, the blocks of a file on an asynchronous array are independently scattered about their disks. Thus, the retrieval time,  $R_{async}$  will be the maximum of  $k$  samples of  $X$  plus  $T$ . If we use  $X_k$  to denote this maximum, we have

$$\bar{R}_{async} = \bar{X}_k + T$$

Finally, for the new scheme, the block retrieval times are also independent. Using  $X_{k,n}$  for the  $k^{th}$  largest of  $n$  samples of  $X$ , the expected retrieval time will be

$$\bar{R}_{new} = \bar{X}_{k,n} + T$$

If we use  $F(x)$  as the distribution function of  $X$  and  $f(x)$  as its density function, then the corresponding functions for  $X_k$  are:

$$F_k(x) = F^k(x)$$

and

$$f_k(x) = kF^{k-1}(x)f(x)$$

The distribution function of  $X_{k,n}$  is just the probability

$$P | X_k \leq x | = p\gamma[ \text{at least } k \text{ of } nXs \text{ are each } \leq x ]$$

which is

$$F_{k,n} = \sum_{i=k}^n \binom{n}{i} F^i(x)[1 - F(x)]^{n-i}$$

and, thus,  $X_{k,x}$  will have density function

$$f_{k,n}(x) = \frac{n!}{(k-1)!(n-k)!} F^{k-1}(x)[1 - F(x)]^{n-k} f(x)$$

To do a comparison, we will have to make an assumption about  $f(x)$ . If, as is quite reasonable,  $f(x)$  is uniform, say between zero and one, the expected retrieval time for a synchronous disk array is

$$\bar{R}_{sync} = \frac{1}{2} + T$$

and for an asynchronous one, it is

$$\bar{R}_{async} = k \int_0^1 x^k dx + T = \frac{k}{k+1} + T$$

Using the uniform density for  $X$ , the density for  $X_{k,n}$  becomes:

$$f_{k,n}(x) = \frac{n!}{(k-1)!(n-k)!} x^{k-1} (1-x)^{n-k} ; 0 \leq x \leq 1$$

Then, the expected value of the retrieval time for the new scheme is:

$$\bar{R}_{new} = \frac{n!}{(k-1)!(n-k)!} \int_0^1 x^k (1-x)^{n-k} dx + T$$

But this integral is just the definition of the beta function:

$$\bar{R}_{new} = \frac{n!}{(k-1)!(n-k)!} B(k+1, n-k+1) + T$$

Then, substituting the gamma function form for the beta function, we get:

$$\bar{R}_{new} = \frac{n!}{(k-1)!(n-k)!} \frac{\Gamma(k+1)\Gamma(n-k+1)}{\Gamma(n+2)}$$

Since the arguments of the gamma functions are all integers, this simplifies to:

$$\bar{R}_{new} = \frac{k}{n+1} + T$$

Therefore, for a uniform distribution of seek time and rotational latency, the new scheme is always better than asynchronous Level 0 and, for  $k$  less than  $n/2$ , it is better than synchronous Level 0.

Although the assumption of uniform distribution is convenient, it is not necessary. We can show that the new scheme is better than either synchronous or asynchronous Level 0 RAID regardless of  $f(x)$ . The probability that a single trial of the new scheme will retrieve the file faster than synchronous Level 0 RAID is:

$$P[R_{new} < R_{sync}] = \int_{-\infty}^{\infty} \int_{-\infty}^{\beta} f_{k,n}(\alpha) f(\beta) d\alpha d\beta = \int_{-\infty}^{\infty} F_{k,n}(\beta) f(\beta) d\beta$$

Using the definition of the distribution function  $F_{k,n}(\beta)$ , this becomes:

$$P[R_{new} < R_{sync}] = \int_{-\infty}^{\infty} \sum_{i=k}^n \binom{n}{i} F^i(\beta) [1 - F(\beta)]^{n-i} f(\beta) d\beta$$

But, since  $f(\beta)d\beta$  is just  $dF(\beta)$  we can do a change of variables to make this:

$$P[R_{new} < R_{sync}] = \sum_{i=k}^n \binom{n}{i} \int_0^1 F^i [1 - F]^{n-i} dF = \sum_{i=k}^n \binom{n}{i} B(i+1, n-i+1)$$

Then, substituting the gamma function definition for the beta function and expanding the binomial and the gamma functions as factorials, this gives us:

$$P[R_{new} < R_{sync}] = \sum_{i=k}^n \frac{n!}{(n-1)!i!} \frac{i!(n-i)!}{(n+1)!} = \frac{n-k+1}{n+1}$$

In other words, if  $k$  is smaller than  $(n+1)/2$ , the access time for the new system will probably be shorter than for a synchronous Level 0 RAID system, regardless of the actual seek time density.

To show that the new scheme is better than asynchronous Level 0, we just show that synchronous Level 0 is better than asynchronous Level 0. Using the previous arguments with  $n = k$ , we see that

$$P[R_{async} < R_{sync}] = \frac{1}{k+1}$$

Then, for large enough  $k$ , the probability that  $R_{new}$  is smaller than  $R_{async}$  can be made arbitrarily large.

## V Discussion

As noted, the new scheme associates redundant information with files rather than with a stripe of data across disks. This has several implications, some favorable and some unfavorable, which are examined in this section. The file-oriented nature of redundancy can cause problems for small files, but allows per-file access time optimization and capacity-dependent redundancy tuning. It also effects the interaction of the new scheme with classic RAID. Finally, we will examine the effects of the simplifications introduced to explain the new scheme.

For small files, the value of  $k$  will be small. Indeed, for modern disk drives, the minimum block size can be on the order of the length of a small file, implying a  $k$  value of 1. Thus, the code rate ( $k/n$ ) can be no more than 1/2. This might be wasteful if the goal is to out-perform an asynchronous system, but it is necessary to beat a synchronous one. Thus, the small file penalty is not significant.

Since the level of redundancy is associated with each file, it can differ for different files. Therefore, files that might be retrieved often could have more redundancy and, hence, faster retrieval. Archival files, whose access times are not critical, might have little or no redundancy. When a disk subsystem begins to fill up (as they all seem to do), it is possible to reduce the redundancy to conserve disk space. Indeed, it is even possible to remove some redundancy from selected files as the disks become full.

The various levels of RAID are designed to provide reliability in the face of disk failures. The new scheme is inherently redundant and would render the major arguments in favor of RAID moot. The redundancy used to reduce access times can also be used to reconstruct data destroyed by faults. However, it is also possible for the new scheme to be combined with RAID; the additional redundancy would then be used only to improve the access times of selected files.

In the explanation of the new scheme, we assumed that the file had  $k$  blocks, where  $k$  was somewhat less than the number of disks. Further, we assumed that the  $n$  blocks used to represent the file would each be on a separate disk. Although this is a pleasant model, it is far from necessary. Actually, all we require is that the seek time and rotational latency of each block be independent with density  $f(x)$ . To use the redundancy for fault recovery, we want the blocks to be scattered across the disks, providing the required diversity. Both of these goals are easily achievable for arbitrary  $k$  and  $n$ .

## VI Conclusions

If current technology trends continue, computer system performance will be largely determined by disk subsystem performance, and disk subsystem performance will be largely determined by access times. Access times are improving at less than half the rate of transfer bandwidth and capacity and one-fifth the rate of processor power. Eventually, the capacity and processing penalties of the proposed scheme should be substantially less than the value of the reduction in access time. The future appears bright for a method that trades a relatively scarce resource for ones that are becoming plentiful.

Given the complexity of practical disk subsystems, a complete assessment of the value of the proposed scheme awaits a careful simulation study. However, the analysis presented here argues strongly that such a study is worthwhile.

## References

- [1] D. Bitton and J. Gray, "Disk Shadowing", *Very Large Database Conf. XIV*, pp 331-338, 1988.
- [2] J. Chandy and A. L. N. Reddy, "Failure Evaluation of Disk Array Organizations", *Proc. 1993 Int'l. Symp. Comp. Arch.*, May 1993.
- [3] P. M. Chen, G. Gibson, R. H. Katz, and D. A. Peterson, "An Evaluation of Redundant Arrays of Disks using an Amdahl 5890," *Proc. 1990 SCM SIGMETRICS Conf. on Measurement and Modeling of Comp. Sys.*, May 1993.
- [4] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: High-Performance, Reliable Secondary Storage", Tech. Rpt. UCB/CSD-93-778, U. Cal. Berkeley, 1993.
- [5] G. A. Gibson, *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*, PhD Thesis, U. Calif. at Berkeley, December 1991.
- [6] M. Y. Kim, "Synchronized Disk Interleaving," *IEEE Trans. on Comp.*, C-35(11), pp 978-988, 1986.
- [7] M. Y. Kim and A. N. Tantawi, "Asynchronous Disk Interleaving: Approximating Access Delays," *IEEE Trans. on Comp.*, C-40(7), pp 801-810, 1991.
- [8] E. K. Lee and R. H. Katz, "An Analytic Performance Model of Disk Arrays and its Applications," Tech Rpt UCB/CSD 91/660, Univ. of Calif. at Berkeley, 1991.
- [9] J. L. Massey, *Information Theory: Profiting from Errors*, Univ. of Notre Dame, 1976.
- [10] D. A. Patterson, G. Gibson, and R. H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)", *Int'l Conf. on Management of Data (SIGMOD)*, pp 109-116, June 1988.
- [11] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann, 1994.

- [12] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," *J. Soc. Ind. App'l. Math.*, 8, pp 300-304, 1960.

# Progress Towards Achieving Channel Capacity \*

Daniel J. Costello, Jr.

Department of Electrical Engineering  
University of Notre Dame  
Notre Dame, Indiana 46556

Lance C. Perez

Department of Electrical Engineering  
University of Notre Dame  
Notre Dame, Indiana 46556

Dedicated to James L. Massey on the occasion of his 60<sup>th</sup> birthday.

## Abstract

A historical overview of the progress of coding theory towards achieving channel capacity is presented. The first half of the paper is concerned with advances in the power-limited environment of scientific space and satellite communication links. The second half of the paper is concerned with progress in the bandwidth-limited environment of commercial satellite and wireline applications. Special attention is paid to contributions made by Massey, his students, and his co-workers.

## I Introduction

With his 1948 paper “*The Mathematical Theory of Communication*,” Claude E. Shannon stimulated a body of research that has evolved into the two modern fields of Information Theory and Coding Theory [1]. The fundamental philosophical contribution of this seminal treatise was the formal application of probability theory to the study and analysis of communication systems. The theoretical contribution of Shannon’s work was a useful definition of “information” and several “channel coding theorems” which gave explicit upper bounds, called the channel capacity, on the rate at which “information” could be transmitted reliably on a given communications channel.

In the context of current research in coded modulation, the result of primary interest is the “noisy channel coding theorem for continuous channels with average power limitations.” This theorem states that the capacity,  $C$ , of a continuous additive white Gaussian noise (AWGN) channel with bandwidth  $W$  is given by

$$C = \frac{1}{2} \log_2 \left( 1 + \frac{E_s}{N_0 W} \right) \quad (1)$$

---

\*This work was supported in part by NASA Grants NAG5-557 and NAG3-1549, and NSF Fellowship NSF-RCD89-54851.

where  $E_s$  is the average signal energy in each signaling interval,  $T$ , and  $N_0/2$  is the two-sided noise power spectral density. This theorem is both profound in its implications and, fortunately so for communication engineers, frustrating in its ambiguity.

It is profound, because it states unequivocally that for any transmission rate,  $R$ , less than or equal to the channel capacity,  $C$ , there exists a coding scheme that achieves an arbitrarily small probability of error; conversely, if  $R$  is greater than  $C$ , no coding scheme can achieve reliable communication. It is frustrating, because like most existence theorems it gives no hint as to how to find the appropriate coding scheme or how complex it must be. Communication engineers and coding theorists make their living trying to create schemes that achieve the levels of performance promised by Shannon's results.

In this paper, a historical overview is given of the progress of coding theory towards achieving channel capacity in real communication systems. Special attention is paid to the contributions made by Massey, his students, and his co-workers. We shall see that his research has played a key role in closing the gap between real system performance and the promised land of channel capacity.

## II A Review of the Concept of Capacity

The bound in (1) can be put into a form more useful for the present discussion by introducing the parameter  $\eta$  called the spectral efficiency. That is,  $\eta$  represents the average number of information bits transmitted per signaling interval. Assuming perfect Nyquist signaling, then

$$0 \leq \eta \leq C/B$$

and

$$E_s/N_0 = \eta E_b/N_0,$$

where  $E_b$  is the average energy per information bit. Substituting the above relations into (1) and performing some minor manipulations yields

$$E_b/N_0 \geq \frac{2^\eta - 1}{\eta}, \quad (2)$$

which relates the spectral efficiency,  $\eta$ , to the signal-to-noise ratio (SNR), as expressed by  $E_b/N_0$ . The bound of (2) manifests the fundamental tradeoff between bandwidth efficiency and  $E_b/N_0$ . That is, increased bandwidth efficiency can be reliably achieved only with a corresponding increase in the minimum required  $E_b/N_0$ . Conversely, the minimum required SNR can be reduced only by decreasing the bandwidth efficiency of the system.

The bound of (2) is shown plotted in Figure 1 and labeled Shannon's bound. This curve represents the absolute best performance possible for a communication system on the AWGN channel. The performance of a particular system relative to Shannon's bound may be interpreted in two distinct ways.

First, the capacity bound may be interpreted as giving the minimum signal-to-noise ratio required to achieve a specific bandwidth efficiency with an arbitrarily small probability of error. For example, if one wants to transmit  $\eta = 1$  information bit per signal, then there exists a coding scheme that operates reliably with an SNR of 0dB. Conversely, any coding scheme, no matter how complex, sending  $\eta = 1$  information bit per signal with an SNR

less than 0dB will be unreliable. This interpretation, referred to as the power-limited case, indicates the power reduction available using appropriate coding schemes.

Second, the capacity bound may be interpreted as giving the maximum spectral efficiency at which a system may operate reliably for a fixed signal-to-noise ratio. For example, if an SNR of 0dB is available, then there exists a coding scheme that operates reliably with a spectral efficiency of  $\eta = 1$  information bit per channel signal. Conversely, any coding scheme, no matter how complex, sending *more than*  $\eta = 1$  information bit per signal will be unreliable if the SNR is 0dB. This interpretation, referred to as the bandlimited case, indicates the increase in spectral efficiency available using appropriate coding schemes.

In real communication systems, there are many practical considerations that take precedence over Shannon's bound in design decisions. For example, satellite communication systems that use nonlinear travelling wave tube amplifiers (TWTA's) require constant envelope signaling such as  $M$ -ary phase shift keying (MPSK). It therefore seems reasonable to ask what the minimum SNR required to achieve reliable communication is *given* a modulation scheme and a bandwidth efficiency,  $\eta$ .

For the discrete input, continuous output, memoryless AWGN channel with  $M$ -ary one dimensional, e.g., amplitude modulation (AM), or two dimensional, e.g., amplitude/phase (PSK, QAM), modulation and assuming equiprobable signaling, the capacity bound becomes [13]

$$\eta^* = \log_2(M) - \frac{1}{M} \sum_{i=0}^{M-1} E \left\{ \log_2 \sum_{j=0}^{M-1} \exp \left[ \frac{|a^i + n - a^j|^2 - |n|^2}{N_0} \right] \right\}, \quad (3)$$

where  $a^i$  is a channel signal,  $n$  is a Gaussian distributed noise random variable with mean 0 and variance  $N_0/2$ , and  $E$  is the expectation operator. The bound of (3) is plotted in Figure 1 for BPSK, QPSK, and 8PSK modulation.

For a specified signaling method and spectral efficiency, this bound represents the minimum SNR required to achieve reliable communication. For example, to send  $\eta = 1.5$  information bits per signal using QPSK modulation requires a minimum SNR of  $E_b/N_0 = 1.64$ dB. This is 0.76dB more than an ideal system without any modulation constraints. Alternatively, a QPSK system operating with an SNR of  $E_b/N_0 = 1.64$ dB can transmit a maximum of  $\eta = 1.5$  bits per signal. This is 0.44 bits per signal less than an ideal system without any modulation constraints.

### III The Power Limited Case

Most of the early work in coding theory was directed at the power limited interpretation of the capacity curve [3]. This was principally due to two factors. First, many early applications of coding were on NASA space communication and satellite systems where power was very expensive and bandwidth was plentiful. Second, it was widely believed that no coding schemes existed that could increase the spectral efficiency of a system without a corresponding increase in power. Thus, our survey begins with the early applications of coding theory to space and satellite systems.

The starting point is taken to be an uncoded BPSK system with coherent detection. This system has a spectral efficiency of  $\eta = 1.0$  bit per signal. Simulation results and

analytical calculations have shown that this system achieves a bit error rate (BER) of  $10^{-5}$  with an SNR of  $E_b/N_0 = 9.6$  dB. Assuming that a BER of  $10^{-5}$  is “virtually error free”, this point is plotted on Figure 1 with the label BPSK.

From the capacity curve, it can be seen that the minimum required SNR to achieve a spectral efficiency of  $\eta = 1.0$  bit per symbol is 0.0 dB and thus a power savings of 9.6 dB is theoretically possible with an appropriate coding scheme. Looking at the BPSK capacity curve, however, reveals that to achieve  $\eta = 1.0$  with BPSK modulation requires a much larger minimum SNR. (In fact, the BPSK capacity curve does not reach 1.0 bit/symbol until  $E_b/N_0 \approx 10.25$  dB. The apparent conflict between this result and the BPSK simulation results is due to the discrepancy between a BER of  $10^{-5}$  and true error free performance.) Thus, to significantly improve on the performance of uncoded BPSK requires either a reduced spectral efficiency or the use of a larger signal set.

One of the earliest attempts to close the 9.6dB gap between the performance of uncoded BPSK and capacity was the use of a rate 6/32 biorthogonal (Reed-Muller) block code. This code was used on the Mariner Mars and Viking missions in conjunction with BPSK modulation and a soft decision maximum-likelihood decoder, the “Green Machine”. This system had  $\eta = 6/32 = 0.1875$  bits/signal and achieved a BER of  $10^{-5}$  with an SNR of  $E_b/N_0 = 6.4$  dB. This point is plotted in Figure 1 with the label “(32,6) Biorthogonal”. From Figure 1, it is easily seen that the (32,6) biorthogonal code requires 3.2dB less power than uncoded BPSK for the same BER, but it requires more than 5 times the bandwidth and is still 7.7 dB away from capacity and 7.5 dB away from the BPSK curve.

A significant advance occurred in 1963 when Massey invented threshold decoding [4], a suboptimal algebraic method for decoding convolutional codes that lends itself to relatively inexpensive decoder implementations. Threshold decoding made it practical to use high rate, long constraint length convolutional codes and changed the way capacity was approached in practical applications. Previously, because of decoding complexity considerations, power reductions could only be obtained by sacrificing bandwidth efficiency. With threshold decoding, however, it became practical to obtain power reductions at high bandwidth efficiencies with only modest increases in encoder/decoder complexity.

For example, a rate 7/8, constraint length  $\nu = 146$ , convolutional code with threshold decoding was used in the DITEC system built by the Communications Satellite Corporation to transmit digital television signals over the INTELSAT IV satellite link [5]. This coding scheme had  $\eta = 7/8 = 0.875$  bits/signal and achieved a BER of  $10^{-5}$  with an SNR of 6.8dB and thus required 2.8 dB less power than the uncoded system. Its performance is plotted in Figure 1 with the label “DITEC”. Though the (32,6) biorthogonal code has a slightly greater power reduction than the (8, 7, 146) convolutional code with threshold decoding, it requires close to 5 times the bandwidth. The DITEC system is 7.35dB away from capacity, but only 4.0dB away from the BPSK curve. From the capacity perspective, the DITEC system is a more efficient coding scheme.

In 1967, a new algebraic decoding technique for hard decision decoding of Bose-Chaudhuri-Hocquenghem (BCH) codes was discovered by Berlekamp [6] and Massey [7]. This enabled a whole class of powerful block codes to be efficiently decoded. For example, the (255,123) BCH code has  $\eta = 123/255 \approx 0.5$  bits/signal and achieves a BER of  $10^{-5}$  with an SNR of 5.7 dB when decoded using the Berlekamp-Massey Algorithm. (See Figure 1, “(255,123) BCH”.) Unfortunately, the Berlekamp-Massey Algorithm and most other block decoding

techniques are hard decision decoding algorithms which incur a 2.0 to 3.0dB penalty on AWGN channels compared to soft-decision decoding algorithms.

The invention of sequential decoding [8] for convolutional codes and its subsequent refinement [9] marked the advent of a new era in the application of coding to space and satellite systems. It was now possible to use powerful long constraint length convolutional codes with soft decision decoding. For the first time, practical communication systems were approaching channel capacity.

Sequential decoding was first used on the Pioneer 9 mission on an experimental basis [10], but the coding scheme was changed for subsequent missions. The Pioneer 10 and Pioneer 11 missions in 1972 and 1973, respectively, both used a long constraint length (2, 1, 31) nonsystematic, quick-look-in (QLI) convolutional code constructed by Massey and Costello [11]. A sequential decoder using the Fano algorithm with 3-bit soft decisions was chosen for decoding [9], [12]. This scheme had  $\eta = 1/2 = 0.5$  bits/signal and achieved a BER of  $10^{-5}$  with an SNR of 2.5dB. This is only 3.3dB away from capacity and only 2.4 dB away from the BPSK curve. (See Figure 1, “Pioneer 10”.)

Unfortunately, sequential decoding algorithms have a variable computation characteristic which results in large buffering requirements, and consequently large decoding delays, and/or incomplete decoding of the received sequence. In addition, the performance of convolutional codes with sequential decoding is ultimately limited by the computational cutoff rate,  $R_0$ , which requires higher SNR’s than capacity to achieve reliable communication at a given spectral efficiency [13]. For example, to achieve reliable communication at  $\eta = 0.5$  bits/signal using sequential decoding and BPSK modulation on the AWGN channel requires  $E_b/N_0 = 2.4$ dB, whereas the capacity bound requires an SNR of only  $-0.8$ dB. The SNR at which the (2, 1, 31) Pioneer code achieves a BER of  $10^{-5}$  is only 0.1 dB away from the  $R_0$  bound, and thus there is little to be gained with longer constraint length codes and sequential decoding at this spectral efficiency and BER.

These undesirable characteristics and the possibility of higher decoder speeds led to the use of maximum-likelihood Viterbi decoding [14] in the next generation of space systems. The Viterbi algorithm, like sequential decoding, is compatible with a variety of modulation and quantization schemes. Unlike sequential decoding, the Viterbi algorithm has a fixed number of computations per decoded branch and thus does not suffer from incomplete decoding and, ultimately, is not limited by the computational cutoff rate.

The Voyager spacecraft launched in 1977 used a short constraint length (2, 1, 6) convolutional code with a 3-bit soft decision Viterbi decoder. This system also has  $\eta = 1/2 = 0.5$  bits/signal, but requires an SNR of 4.5 dB to operate at a BER of  $10^{-5}$ . (See Figure 1, “(2,1,6) BPSK”.) Though this scheme results in significant power reductions compared to uncoded systems, its performance is nearly 2.0 dB worse than the Pioneer system, due to the short constraint length used. However, it guarantees complete decoding of the received sequence and is capable of very high speed operation [15].

Recently, technological advances have made it practical to build maximum-likelihood Viterbi decoders for moderate constraint length convolutional codes. The current culmination of this effort is the Big Viterbi Decoder (BVD) for a (4, 1, 14) convolutional code built at the Jet Propulsion Laboratory for use on the Galileo mission [16]. This code has  $\eta = 1/4 = 0.25$  bits/signal and achieves a BER of  $10^{-5}$  with an SNR of only 1.75dB. (See Figure 1, “(4,1,14) BPSK”.) The performance of the (4, 1, 14) code with the BVD is only

2.96 dB away from capacity and 2.65 dB away from the BPSK curve [17].

The  $(2, 1, 6)$  and  $(4, 1, 14)$  convolutional codes can also be used as inner codes in a concatenated coding scheme [18]. With concatenation, the required SNR is reduced even further with only a slight reduction in spectral efficiency. In NASA applications, the outer code is usually chosen to be the  $(255, 223)$  Reed-Solomon code over  $GF(2^8)$  [19]. This code results in an additional power reduction of 2.0dB for the  $(2, 1, 6)$  code and 0.8 dB for  $(4, 1, 14)$  code. The performance of the  $(2, 1, 6)$  and  $(4, 1, 14)$  codes in a concatenated system with the  $(255, 223)$  RS outer code is shown in Figure 1 with the labels “Voyager” and “Galileo”, respectively. The outer Reed-Solomon code can be efficiently decoded using the Berlekamp-Massey algorithm [6], [7]. The concatenated Voyager and Galileo systems operate in the same region as the Pioneer system. Thus, short and moderate constraint length codes with Viterbi decoding in concatenated systems can be considered as alternatives to long constraint length codes with sequential decoding.

Recently, there have been two notable advancements in the power limited environment. First, it has been shown that iterative decoding techniques for concatenated systems can result in an additional gain of roughly 0.5 dB [20],[21], [22]. A more recent result, involving a class of codes called “Turbo-codes” with an iterative decoding algorithm, may come spectacularly close to the Shannon limit [23]. Second, progress in soft decision decoding of block codes shows promise of adding up to 2.0dB to the performance of block codes that have previously been decoded using only hard decisions [24], [25], [26]. The performance of a  $(64, 40)$  code, based on the  $(64, 42)$  Reed-Muller code, and a  $(64, 22)$  Reed-Muller code with soft decision decoding is shown in Figure 1 [26]. These schemes could find application in low rate, power limited environments where very high speed decoding is required.

## IV The Bandwidth Limited Case

As was mentioned previously, the focus of much of the early work in coding theory was on reducing power requirements. This was due, in part, to the belief that this was the only area in which coding could be helpful. This viewpoint was changed dramatically with Ungerboeck’s discovery of trellis coded modulation [27], which drew much of its inspiration from a paper by Massey [28].

In his 1982 paper, Ungerboeck found trellis codes for one-dimensional amplitude modulation and two-dimensional amplitude/phase modulation schemes. His rate 1/2, constraint length  $\nu = 6$  code with 4-ary Amplitude Modulation has a spectral efficiency of  $\eta = 1.0$  bit/signal and requires an SNR of 5.1dB to reach a BER of  $10^{-5}$  with soft decision Viterbi decoding. This code has the same spectral efficiency as uncoded BPSK and requires 4.5dB less power to achieve the same BER! (See Figure 1, “ $(2,1,6)$  AM”.) Note, however, that the  $(2, 1, 6)$  convolutional code used on Voyager can be used with Gray mapped QPSK modulation to achieve a BER of  $10^{-5}$  with  $E_b/N_0 = 4.5$ dB, 0.6 dB less than the AM code! (See Figure 1, “ $(2,1,6)$  QPSK”).

Ungerboeck was able to construct better trellis codes by using two-dimensional signal sets. The rate 2/3, constraint length  $\nu = 2$  and  $\nu = 6$  codes with 8PSK modulation have a spectral efficiency of  $\eta = 2.0$  bits/signal and require SNR’s of 7.2dB and 6.0dB, respectively, to reach a BER of  $10^{-5}$  with soft decision Viterbi decoding [29]. (See Figure 1, “ $(3,2,2)$  8PSK” and “ $(3,2,6)$  8PSK”, respectively.) The constraint length  $\nu = 2$  and  $\nu = 6$  8PSK

Ungerboeck codes require 2.4 dB and 3.6 dB less power, respectively, than uncoded QPSK to achieve the same BER and spectral efficiency.

It was subsequently shown that the Voyager (2, 1, 6) convolutional code could also be used as a suboptimal trellis code with two-dimensional QAM and PSK signal sets [30]. This enabled a standard (2, 1, 6) Viterbi decoder to be modified for use with 8PSK modulation to achieve a spectral efficiency of  $\eta = 2.0$  bits/signal. It is surprising that a simple modification of an existing code requires only 0.32 dB more power than Ungerboeck's optimal  $\nu = 6$  code to achieve a BER of  $10^{-5}$ !

The performance of Ungerboeck's codes quickly dispelled the belief that power reduction is only attainable with a corresponding decrease in spectral efficiency. This was a welcome result for designers of satellite systems, to whom it was becoming increasingly apparent that bandwidth was indeed a precious commodity, and to modem designers who had been frustrated in their attempts to go beyond data rates of 9600 bits per second [31].

In 1990, Pietrobon, et. al. [32] constructed a class of multidimensional  $LxM$ PSK trellis codes for  $L = 1, \dots, 4$  and  $M = 4, 8$ , and 16. These codes had improved rotational invariance properties compared to Ungerboeck's original two-dimensional MPSK codes and allowed fractional spectral efficiencies. A soft decision Viterbi decoder for a rate 5/6,  $\nu = 4$ , 2x8PSK code was implemented for testing on the NASA Tracking and Data Relay Satellite System (TDRSS). This code has a spectral efficiency of  $\eta = 2.5$  bits per signal and achieves a BER of  $10^{-5}$  with an SNR of  $E_b/N_0 = 7.95$ dB. (See Figure 1, "(6,5,4) 2x8PSK".) This code requires 1.95dB less power than uncoded QPSK to achieve the same BER, yet has a spectral efficiency that is 0.5 bits/signal greater! This clearly demonstrates that it is possible to simultaneously reduce power and increase spectral efficiency.

Not surprisingly, sequential decoding of convolutional codes and block coding have undergone a renaissance in the world of coded modulation. A rate 2/3, constraint length  $\nu = 17$ , 8PSK trellis code with a spectral efficiency of  $\eta = 2.0$  bits/signal has been proposed for use on NASA's Advanced Tracking and Data Relay Satellite System (ATDRSS) [33]. Using soft decision sequential decoding, this code needs an SNR of 4.94 dB to attain a BER of  $10^{-5}$ , which is only 3.8dB away from capacity and 3.2dB away from the 8PSK bound [34].

The primary application of block codes to coded modulation is in the area of multilevel coding [35]. Multilevel techniques allow the simple construction of very complex block coded modulation (BCM) schemes using known binary block codes. Many of the properties of the BCM scheme, in particular the minimum squared Euclidean distance (MSED), can be determined using the structure of the binary block codes. In addition, multilevel codes can be decoded using suboptimal multistage decoding [35], [36], which may give them a performance versus complexity advantage compared to trellis codes. A BCM scheme [37] is shown in Figure 1 with the label "(17,16,2) 8x8PSK BCM". This code has a spectral efficiency of  $\eta = 2.286$  bits per signal and reaches a BER of  $10^{-5}$  with an SNR of 8.1 dB using a soft decision multistage decoder that is only slightly more complex than a 4-state Viterbi decoder.

For spectral efficiencies of 3.0 bits/signal or greater, rectangular Quadrature Amplitude Modulation (QAM) signal sets offer better performance than PSK signal sets. (QAM signal sets more efficiently pack the two-dimensional signal space and thus can achieve the same minimum distance between signal points, which effectively limits the performance of a

trellis code, with less average energy per signal than a comparable PSK signal set.) This is manifest in the sequential decoding performance of two  $\nu = 16$  codes with 16QAM and 16PSK modulation and  $\eta = 3.0$  bits/signal. Plotted in Figure 1 with the labels “(4,3,16) 16QAM” and “(4,3,16) 16PSK”, there is a gap of over 2.0dB in favor of the QAM code. This is almost identical to the gap between the 16PSK and 16QAM capacity bounds. The (4, 3, 16) code is also within 2.1 dB (or 0.6 bits/signal) of the 16QAM capacity curve.

The performance advantage of QAM signal sets and the difficulty of detecting the phase of MPSK signal sets with  $M$  large has led NASA to consider using QAM trellis codes on future satellite links that require high spectral efficiencies. Of particular interest are a class of rotationally invariant, nonlinear, QAM trellis codes constructed by Pietrobon, et. al. [38]. These codes offer full rotational invariance, but have slightly reduced minimum distances compared to linear QAM codes. However, because the nonlinear codes have lower error coefficients and sparser distance spectra than the linear codes, their performance at moderate SNR’s and practical error rates is very close to that of the linear codes.

For example, the linear constraint length  $\nu = 4$ , 16QAM code is  $180^\circ$  rotationally invariant and has a minimum distance of 6.0, while the nonlinear code with  $\nu = 4$  is  $90^\circ$  invariant and has a minimum distance of only 5.0. However, to achieve a BER of  $10^{-5}$  with a spectral efficiency of  $\eta = 3.0$  bits per signal, the linear code requires an SNR of 8.4 dB while the nonlinear code needs only a slightly higher SNR of 8.5 dB! (See Figure 1, “(4,3,4) 16QAM Nonlinear”.) Although the performance of the (4, 3, 4) nonlinear code is worse than the (4, 3, 16) code, it offers improved rotational invariance properties and can be decoded using a Viterbi decoder without encountering the incomplete decoding problem inherent to sequential decoding.

The practical importance of rotational invariance and the desire to avoid nonlinear encoders has resulted in an effort to find linear nonbinary trellis codes with good distance properties. Massey, et. al., found a class of “ring codes” over the integer ring  $Z_M$  for  $M$ -ary PSK modulation [39]. These codes have good distance properties and have an algebraic structure that makes them naturally rotationally invariant. It has since been realized that the algebraic structure of a field or a ring is not necessary in order to find good codes, and a number of researchers are now trying to construct good trellis codes over groups [40], [41].

Once considered inappropriate for the telephone channel due to its relatively high SNR and low bandwidth, coding is now commonplace in high-speed modems [31]. A constraint length  $\nu = 3$  nonlinear code with a modified 128 QAM signal set constructed by Wei [42], which achieves a gain of 4dB compared to uncoded modulation, was adopted for the CCITT V.32 and V.33 modem standards for data transmission at rates up to 14.4 kbps. A class of multidimensional trellis codes constructed by Wei [43] is currently being considered for use in even higher speed modems. The  $\nu = 3$  and  $\nu = 4$ , 4-dimensional Wei codes are 2.7 and 3.0 dB away from the capacity curves, respectively, for spectral efficiencies between  $\eta = 5$  to  $\eta = 7$  bits/signal. The  $\nu = 4$  code is likely to be adopted for the CCITT V.34 modem standard for data rates up to 28.8 kbps. A more thorough discussion of these codes and their application to high speed modems can be found in [31].

## V Conclusion

A historical overview of the progress coding theory has made toward achieving channel capacity has been presented with an emphasis on the work of Massey and his colleagues. In the power limited environment, long constraint length convolutional codes with sequential decoding and concatenated coding systems are within a few decibels of capacity. In the bandwidth limited environment, once not even considered an appropriate milieu for coding, coded modulation has resulted in significant power savings, previously thought to be unachievable, at high data rates. Perhaps we are again “nearing practical achievement of the goals established by Shannon” 45 years ago [3].

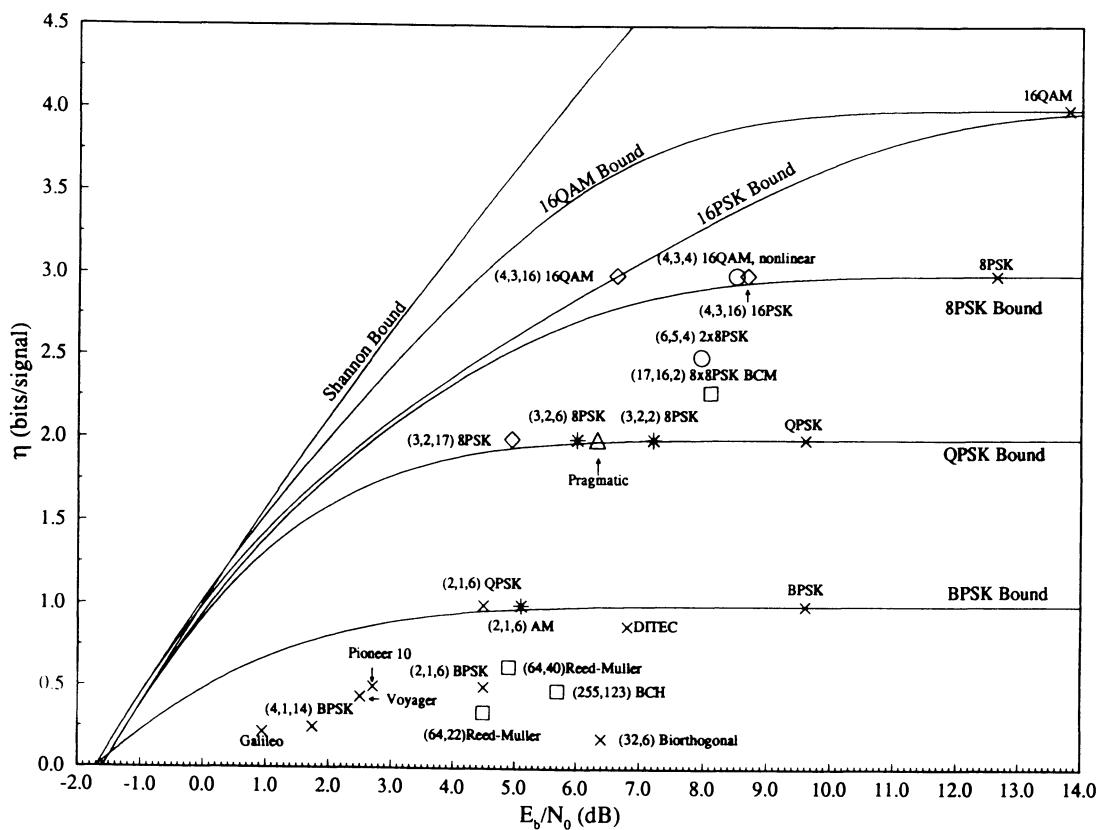
## References

- [1] C. E. Shannon, “The Mathematical Theory of Communication”, *Bell Syst. Tech. J.*, **Vol. 27**, pp. 379-423, 1948.
- [2] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York, 1968.
- [3] G. D. Forney, Jr., “Coding and its application in space communications,” *IEEE Spectrum*, pp. 47–58, June 1970.
- [4] J. L. Massey, *Threshold Decoding*, MIT Press, Cambridge, Mass., 1963.
- [5] W. W. Wu, “New convolutional codes - Part 1,” *IEEE Trans. Commun.*, **COM-23**, pp. 942–956, 1975.
- [6] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, New York, 1968.
- [7] J. L. Massey, “Shift register synthesis and BCH decoding,” *IEEE Trans. Inform. Theory*, **IT-15**, pp. 122-127, 1969.
- [8] J. M. Wozencraft and B. Reiffen, *Sequential Decoding*, MIT, Cambridge, Mass., 1961.
- [9] R. M. Fano, “A heuristic discussion of probabilistic decoding,” *IEEE Trans. Inform. Theory*, **IT-9**, pp. 64–74, 1963.
- [10] G. D. Forney, Jr., “Final report on a study of a sample sequential decoder,” Appendix A, Codex Corp., Watertown, Mass., U. S. Army Satellite Communication Agency Contract DAA B 07-68-C-0093, April 1968.
- [11] J. L. Massey and D. J. Costello, Jr., “Nonsystematic convolutional codes for sequential decoding in space applications,” *IEEE Trans. Commun. Technol.*, **COM-19**, pp. 806–813, 1971.
- [12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, New Jersey, 1983.
- [13] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York, 1968.

- [14] A. J. Viterbi, "Error bounds for convolutional codes and an Asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, **IT-13**, pp. 260-269, 1967.
- [15] G. Fettweis and H. Meyr, "High-speed parallel viterbi decoding: algorithm and VLSI-architecture," *IEEE Communications Magazine*, Vol. 29, pp. 46-55, 1991.
- [16] O. M. Collins, "The subtleties and intricacies of building a constraint length 15 convolutional decoder," *IEEE Trans. Commun.*, **COM-40**, pp. 1810-1819, 1992.
- [17] J. H. Yuen and Q. D. Vo, "In Search of a 2-dB Coding Gain," *TDA Progress Report*, pp. 26-33, July-September 1985.
- [18] G. D. Forney, *Concatenated Codes*, M.I.T. Press, Cambridge, Mass., 1966.
- [19] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Mass., 1984.
- [20] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Trans. Commun.*, **COM-38**, pp. 1138-1144, 1990.
- [21] O. M. Collins and M. Hizlan, "Determinate state convolutional codes," *IEEE Trans. Commun.*, **COM-41**, pp. 1785-1794, 1993.
- [22] T. Woerz and J. Hagenauer, "Multistage decoding of coded modulation using soft output and source information," *Proc. of the 1993 IEEE Inform. Theory Workshop*, Susono-shi, Shizuoka, Japan, pp. 4.5.1-2, June 1993.
- [23] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proc. 1993 IEEE Int. Conf. on Comm.*, Geneva, Switzerland, pp. 1064-1070, 1993.
- [24] J. Snyders and Y. Be'ery, "Maximum-likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. Inform. Theory*, **IT-35**, pp. 963-975, 1989.
- [25] G. D. Forney, Jr., "Coset codes - Part II: binary lattices and related codes," *IEEE Trans. Inform. Theory*, **IT-34**, pp. 1152-1187, 1988.
- [26] S. Lin, "Low Complexity and High Performance Concatenated Coding Schemes for High Speed Satellite and Space Communications," *Technical Report 93-004* to NASA Goddard Space Flight Center, August 1993.
- [27] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, **IT-28**, pp. 55-67, 1982.
- [28] J. L. Massey, "Coding and modulation in digital communications," *Proc. 1974 Int. Zurich Seminar on Digital Comm.*, Zurich, Switzerland, pp. E2(1)-(4), March 1974.
- [29] L. C. Perez, "On the Performance of Multi-Dimensional Phase Modulated Trellis Codes," *Final Technical Report*, NASA Training Grant NGT-70109, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, October 1989.

- [30] A. J. Viterbi, J. K. Wolf, E. Zehavi, and R. Padovani, “A pragmatic approach to trellis-coded modulation,” *IEEE Commun. Mag.*, Vol. 27, pp. 11–19, July 1989.
- [31] G. D. Forney, Jr., “Coded modulation for band-limited channels,” *IEEE Information Theory Society Newsletter*, pp. 1–7, December 1990.
- [32] S. S. Pietrobon, R. H. Deng, A. Lafanechère, G. Ungerboeck and D. J. Costello, Jr., “Trellis-coded multidimensional phase modulation,” *IEEE Trans. Inform. Theory*, **IT-36**, pp. 63–89, 1990.
- [33] D. J. Costello, Jr., L. C. Perez, and F. Wang, “Bandwidth Efficient CCSDS Coding Standards Proposals,” *Semi-annual Status Report*, NASA Grant NAG 5-557, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, May 1992.
- [34] F. Q. Wang and D. J. Costello, Jr., “Probabilistic construction of large constraint length trellis codes for sequential decoding”, *To appear in the IEEE Trans. Commun.*.
- [35] H. Imai and S. Hirakawa, “A new multilevel coding method using error correcting codes,” *IEEE Trans. Inform. Theory*, **IT-23**, pp. 371–377, 1977.
- [36] A. R. Calderbank, “Multilevel codes and multistage decoding,” *IEEE Trans. Commun.*, **COM-37**, pp. 222–229, 1989.
- [37] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, “On multi-level block modulation codes,” *IEEE Trans. Inform. Theory*, **IT 37**, pp. 965-975, 1991.
- [38] S. S. Pietrobon, G. Ungerboeck, D. J. Costello, Jr., and L. C. Perez, “Rotationally invariant nonlinear trellis codes for two-dimensional modulation,” *To appear in the IEEE Trans. Inform. Theory*.
- [39] J. L. Massey, T. Mittelholzer, T. Riedel, and M. Vollenwieder, “Ring convolutional codes for phase modulation,” *Book of Abstracts, 1990 IEEE International Symposium on Information Theory*, San Diego, CA, pg. 176, January 1990.
- [40] H.-A. Loeliger and T. Mittelholzer, “Convolutional codes over groups,” *To appear in the IEEE Trans. Inform. Theory*.
- [41] G. D. Forney, Jr. and M. D. Trott, “The dynamics of group codes: state spaces, trellis diagrams, and canonical encoders,” *IEEE Trans. Inform. Theory*, **IT-39**, pp. 1491–1513, 1993.
- [42] L. F. Wei, “Rotationally invariant convolutional channel coding with expanded signal space. Part II: nonlinear codes,” *IEEE J. Select. Areas Commun.*, **SAC-2**, pp. 672-686, 1984.
- [43] L. F. Wei, “Trellis-coded modulation with multidimensional constellations,” *IEEE Trans. Inform. Theory*, **IT-33**, pp. 483-501, 1987.

Figure 1: Spectral Efficiency,  $\eta$ , versus  $E_b/N_0$  (dB)



# Random Time and Frequency Hopping for Infinite User Population

Sándor Csibi

Department of Telecommunications  
Technical University of Budapest  
Hungary

László Györfi

Department of Mathematics  
Technical University of Budapest  
Hungary

## Abstract

Multiaccess for a Poisson population is considered through a single multi-input, single-output collision channel without feedback for time hopping, and also through  $L$  such channels for frequency hopping. Synchronism as well as asynchronism are included. Upper bounds are given on the decoding error probability. Each of these consists of a first term due to overflow (above a threshold that is a design parameter), a second term due to identification error (under no overflow) and a third term due to collision (under neither overflow nor identification error). The third term, exponential in the message length, is obtained by Hoeffding's inequality for all cases, considering either the original model itself or a well-defined dominating model (with additional erasures generated appropriately). Proofs are given in the first case; just references in the second.

## I Thanks for Surprises

Let us start with two personal comments. Our research areas either are (L. Gy.), or have been for a long while (S. Cs.), nonparametric curve estimation, statistical pattern recognition, stochastic approximation and time series, weakly related to information theory, and where our main motivation is surprises formulated as good news and bad news. Although there were surprises in multiuser information theory, for example, that feedback increases the capacity, the corresponding techniques were not attractive for us. In the same time we enjoyed an everyday flourishing random access practice in a team of communication engineers; but in 1981 suddenly two beautiful surprises happened:

(i) Pippenger [11] has proved that by knowing the multiplicity of the collision the capacity of a multiaccess collision channel with feedback is 1, thus one may have the same rates as for time sharing.

(ii) Massey [8] has shown that the collision channel without feedback is not a useless object; it has approximately the same capacity as the supremum of the throughput of the (unstable) slotted ALOHA:  $e^{-1}$ . At that time it was a big surprise; nobody was brave enough to put the question, how to use a collision channel when the sender is not informed about the possible collision. Moreover, Jim gave a construction, the throughput of which is

the capacity. In Budapest maybe we were the first learning this result from Jim when he was lecturing on it. It was great. Formally, one of the main messages is the following:

**Theorem 1 ([9]):** *The symmetric capacity of the frame asynchronous  $M$ -user collision channel without feedback with  $M \geq 2$  is  $(1 - M^{-1})^{M-1}$ , no matter whether the slots are unsynchronized or synchronized. The common capacity obviously tends to  $e^{-1}$ , from above, as  $M \rightarrow \infty$ .*

Note that below this capacity, error-free transmission is possible. If there is no feedback and the population is not finite then error-free information transmission is impossible. In the sequel we show that below the same capacity a reliable communication is possible up to a fixed finite number of simultaneously active users even for an infinite user population.

Another point we would like to raise here is Jim's principle of "minimum mathematics". We both were uncompromising lovers of abstract mathematics ( $\sigma$ -algebra, reproducing kernel Hilbert space etc.), and therefore the result of this principle seemed to be fairly boring and primitive, which can be useful for undergraduate teaching, but cannot have a role in advanced study or in research. It turned out that this principle is extremely important for introducing students to research.

## II Frame Synchronous Time Hopping

The channel is a multiaccess channel without feedback, where the traffic to send over a common communications channel is in the form of "packets" of some fixed length that we assume take values in the finite field  $GF(Q)$  for some, in general large,  $Q$ . The time axis is assumed to be partitioned into slots whose duration corresponds to the transmission time for one packet; it is further assumed that all users know the slot boundaries. When a user transmits a packet, he must transmit it exactly within a slot. The common communications channel is assumed to be the collision channel without feedback. If, in a particular slot, none of the users are sending a packet (in which case we say each user "sends" the silence symbol), then the channel output in that slot is the silence symbol. If exactly one user is sending a packet in a particular slot, then the channel output in that slot is this packet value, which will be an element of  $GF(Q)$ . If two or more users are sending packets in a particular slot, then the channel output in that slot is the collision symbol. There is no feedback available to inform the senders of the channel outputs in previous slots. If the user population is finite then the coding can be done by a finite set of protocol sequences assigned in a one-to-one manner to the users, even in the frame asynchronous case ([1],[2],[8],[9],[12]).

Here infinite user population is considered such that the arrivals of the messages are according to a homogeneous Poisson process with intensity  $\lambda$ , if the time unit is the slot. A message is a  $k$  vector with components from  $GF(Q)$ .

All users know a common frame, which consists of a fixed number of slots. The messages arriving in a frame are encoded and sent in the next frame.

Coding: choose an integer  $n$  and assume that  $k < n < Q$ . Suppose that the frame length  $N$  is an integer multiple of  $n$ :

$$L = \frac{N}{n},$$

so a frame consists of  $n$  segments of length  $L$ . Each message is encoded by an  $(n, k)$  shortened Reed-Solomon code, and in each of the  $n$  segments a slot is chosen randomly (uniformly distributed, and memoryless). In these chosen  $n$  slots the  $n$  encoded packets are sent, so the  $n$  packets are sent according to a random protocol sequence (random time-hopping sequence) of length  $N$  and weight  $n$ .

For a message from a given user (called the tagged user) the decoding is possible if for the  $n$  slots chosen there are at most  $n - k$  collisions (erasures), and the decoder knows which are the  $n$  chosen slots for the actual message. In other words the decoder can separate its packets from the others. Such separation is possible by random addresses: each user generates a random address, uniformly over  $\{0, 1, \dots, L^K - 1\}$ . These addresses are stored in the head of the packet resulting in  $K \log(L)$  overhead. Note that the decoder should know the serial numbers of the packets sent, but in the frame synchronous case the serial number of a packet is equal to the serial number of the segment where it is sent.

If  $R_{sum}$  denotes the throughput then the main question of interest is the probability of decoding error for

$$R_{sum} < e^{-1}$$

and for infinite user population (Poisson arrivals). Because of an infinite user population, error-free transmission is impossible.

Since in a frame of length  $N$  the expected number of packets arrived is  $\lambda N k$ , which are encoded and sent in the next frame, the throughput is defined by

$$R_{sum} = \frac{\lambda N k}{N} = \lambda k.$$

**Theorem 2** ([3]): *Consider frame synchronous time hopping. If  $1 > \delta > 0$ ,  $\lambda n(1 + \delta) = 1$  and*

$$e^{-1} \geq (1 + \delta)R_{sum},$$

*then*

$$P(\text{decoding error}) \leq e^{-L \frac{\delta^2}{(1+\delta)} \frac{\ln 2}{2}} + L^{-(K-1)} + e^{-2n(e^{-1} - (1+\delta)R_{sum})^2}.$$

From the point of view of any single message this channel will be a memoryless erasure channel with varying erasure probability (varying from frame to frame). Therefore in the proof of Theorem 2 the following lemma has an important role, which is of independent interest.

**Lemma 1** ([3]): *Consider the memoryless erasure channel with input alphabet  $GF(Q)$ , erasure error probability  $p_e$  and capacity  $C = 1 - p_e$ . Apply a shortened  $(n, k)$  Reed-Solomon code over  $GF(Q)$  ( $n < Q$ ). Then for code rate*

$$R = \frac{k}{n} < C$$

*we have that*

$$P(\text{decoding error}) \leq e^{-2(C-R)^2 n}.$$

Observe that Lemma 1 is not an asymptotic result; it gives, for any finite value of  $n$  an exponential upper bound on the decoding error for  $R < C$ . It seems to be a constructive channel coding theorem saying that, by increasing the blocklength  $n$ , the decoding error can be small if  $R < C$ . However, one has to note the condition  $n < Q$ , so the bound cannot be, for a fixed  $Q$ , arbitrarily small. From a practical point of view it can be useful if, for example, a packet is of length about 100 bits then  $Q = 2^{100}$ , so practically we have no limits for  $n$ .

### III Frame Asynchronous Time Hopping

Consider first, for simplicity, frame asynchronous but slot synchronous time hopping. Even if the users have no common frame then the coding suggested is the same as before. The only difference is that the frame is initiated at the very next slot followed by the message arrival time. Unfortunately Lemma 1 cannot be applied here because a given user “sees” an erasure channel with memory.

Note that in this case the head of a packet should contain the serial number of the packet, too, which is an additional overhead of size  $\log(n)$ . Again

$$R_{sum} = \lambda k.$$

**Theorem 3 :** Consider frame asynchronous but slot synchronous time hopping. If  $1 > \delta > 0$ ,  $\lambda n(1 + \delta) = 1$  and

$$e^{-1} \geq (1 + \delta)R_{sum},$$

then

$$P(\text{decoding error}) \leq ne^{-L\frac{\delta^2}{(1+\delta)}\frac{\ln 2}{2}} + 2L^{-(K-1)} + 2e^{-(n-1)(e^{-1} - (1+\delta)R_{sum})^2}.$$

If we compare the bounds in Theorems 2 and 3, then the bound in Theorem 3 is looser (mainly due to  $n - 1$  instead of  $2n$  in the exponent in the third term), but still gives exponential upper bound in the same range:

$$e^{-1} > R_{sum}.$$

A main difference between the synchronous and asynchronous cases is that in the synchronous case the user had virtually a memoryless erasure channel, while in the asynchronous case a segment of a disturbing user can interfere with two consecutive segments of the actual user, so he has virtually an erasure channel with memory. However, specifically for a fixed frame front configuration of a given set of interfering users, the subsequences of each second channel events are already memoryless.

**Lemma 2 ([3]):** Let  $M$  be a Poisson random variable with parameter  $\mu$ . Then for  $0 < \delta < 1$

$$P(M > \mu(1 + \delta)) \leq e^{-\mu\delta^2\frac{\ln 2}{2}}.$$

For a real number  $x$  let  $x^+$  be the positive part of  $x$ , so  $x^+ = x$  if  $x \geq 0$  and 0 otherwise.

**Lemma 3** : Consider an erasure channel with memory. Let  $X_i$  be 1 if there is an erasure in the  $i$ -th slot, and 0 otherwise. Assume that  $X_1, X_3, \dots$  are independent, and  $X_2, X_4, \dots$  are independent with

$$P(X_i = 1) = p_i.$$

Apply a shortened  $(n, k)$  Reed-Solomon code over  $GF(Q)$  ( $n < Q$ ). Then for  $R = \frac{k}{n}$

$$\begin{aligned} P(\text{decoding error}) &\leq \exp \left( -2 \lfloor \frac{n}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (1 - p_{2i} - R) \right]^+ \right)^2 \right) \\ &+ \exp \left( -2 \lfloor \frac{n+1}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} (1 - p_{2i-1} - R) \right]^+ \right)^2 \right). \end{aligned}$$

Note that, in the special case of Lemma 1, the bound of Lemma 3 is a bit larger:

$$2e^{-(C-R)^2 n},$$

so there is a multiplier 2 here, and 2 is missing in the exponent, but it is still meaningful in the same range:  $R < C$ .

**Proof of Lemma 3:**

$$\begin{aligned} \{\text{decoding error}\} &= \left\{ \sum_{i=1}^n X_i > n - k \right\} \\ &= \left\{ \sum_{i=1}^n (X_i - E(X_i)) > \sum_{i=1}^n (1 - p_i - \frac{k}{n}) \right\} \\ &\subset \left\{ \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (X_{2i} - E(X_{2i})) > \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (1 - p_{2i} - R) \right\} \\ &\cup \left\{ \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} (X_{2i-1} - E(X_{2i-1})) > \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} (1 - p_{2i-1} - R) \right\}. \end{aligned}$$

Thus by the Hoeffding's inequality ([5])

$$\begin{aligned} P(\text{decoding error}) &\leq P \left( \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (X_{2i} - E(X_{2i})) > \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (1 - p_{2i} - R) \right) \\ &+ P \left( \frac{1}{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} (X_{2i-1} - E(X_{2i-1})) > \frac{1}{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} (1 - p_{2i-1} - R) \right) \\ &\leq \exp \left( -2 \lfloor \frac{n}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (1 - p_{2i} - R) \right]^+ \right)^2 \right) \\ &+ \exp \left( -2 \lfloor \frac{n+1}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} (1 - p_{2i-1} - R) \right]^+ \right)^2 \right). \end{aligned}$$

**Proof of the Theorem 3:** Let  $M_i$  be the number of messages arrived in  $[j_i - N, j_i]$  where  $j_i$  is the slot of  $i$ -th packet sent and  $M$  be the number of messages arrived in  $[j_1 - N, j_n]$ , then  $M_i$  is a Poisson random variable with parameter  $\lambda N = \lambda n L = L/(1+\delta)$  ( $i = 1, 2, \dots$ ). We cover the decoding error event by the union of three events: (i) there is an overload ( $\max_{i \leq n} M_i > L$ ); (ii) there is no overload, but the separation fails (there is a duplicate in the random addresses selected); (iii) there is no overload, the separation is correct, but the erasure correction fails, *i.e.* there are at least  $n - k + 1$  collisions. Let  $I_{[.]}$  be the indicator variable, then

$$\begin{aligned} P(\text{decoding error}) &\leq P(\max_{i \leq n} M_i > L) \\ &+ E[I_{[\max_{i \leq n} M_i \leq L]} P(\text{separation error} | M_i, i = 1, \dots, n, M)] \\ &+ E[I_{[\max_{i \leq n} M_i \leq L]} P(\text{decoding error} | \text{correct separation}, M_i, i = 1, \dots, n, M)]. \end{aligned}$$

For the first term of the right side apply the union bound and Lemma 2:

$$P(\max_{i \leq n} M_i > L) \leq n P(M_1 > L) \leq n e^{-L \frac{\delta^2}{(1+\delta)} \frac{\ln 2}{2}}.$$

For the second term observe that  $M \leq M_1 + M_n$ , then

$$\begin{aligned} I_{[\max_{i \leq n} M_i \leq L]} P(\text{separation error} | M_i, i = 1, \dots, n, M) \\ &\leq I_{[\max_{i \leq n} M_i \leq L]} (1 - (1 - L^{-K})^{M-1}) \\ &\leq I_{[\max_{i \leq n} M_i \leq L]} (1 - (1 - L^{-K})^{M_1 + M_n - 1}) \\ &\leq (1 - (1 - L^{-K})^{2L}) \\ &\leq 2L^{-(K-1)}. \end{aligned}$$

For the third term apply Lemma 3 with  $p_i = (1 - \frac{1}{L})^{M_i - 1}$ :

$$\begin{aligned} I_{[\max_{i \leq n} M_i \leq L]} P(\text{decoding error} | \text{correct separation}, M_i, i = 1, \dots, n, M) \\ &\leq I_{[\max_{i \leq n} M_i \leq L]} [\exp \left( -2 \lfloor \frac{n}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} ((1 - \frac{1}{L})^{M_{2i}-1} - \frac{k}{n}) \right]^+ \right)^2 \right) \\ &\quad + \exp \left( -2 \lfloor \frac{n+1}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} ((1 - \frac{1}{L})^{M_{2i-1}-1} - \frac{k}{n}) \right]^+ \right)^2 \right)] \\ &\leq \exp \left( -2 \lfloor \frac{n}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} ((1 - \frac{1}{L})^{L-1} - (1 + \delta)R_{sum}) \right]^+ \right)^2 \right) \\ &\quad + \exp \left( -2 \lfloor \frac{n+1}{2} \rfloor \left( \left[ \frac{1}{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} ((1 - \frac{1}{L})^{L-1} - (1 + \delta)R_{sum}) \right]^+ \right)^2 \right) \end{aligned}$$

$$\leq 2e^{-(n-1)(e^{-1} - (1+\delta)R_{sum})^2},$$

where we used that

$$\lim_{L \rightarrow \infty} \left(1 - \frac{1}{L}\right)^{L-1} = e^{-1}$$

in a monotone decreasing way. These three bounds imply the statement.

Applying some more sophisticated techniques (generating additional erasures appropriately and considering a well-defined dominating model introduced in this way) one replaces the coefficient  $n - 1$  in the exponent of the third term of Theorem 3 by  $2n$ :

**Theorem 4 ([4]):** Consider frame asynchronous but slot synchronous time hopping. If  $1 > \delta > 0$ ,  $\lambda n(1 + \delta) = 1$  and

$$e^{-1} \geq (1 + \delta)R_{sum},$$

then

$$P(\text{decoding error}) \leq ne^{-L\frac{\delta^2}{(1+\delta)}\frac{\ln 2}{2}} + 2L^{-(K-1)} + e^{-2n(e^{-1} - (1+\delta)R_{sum})^2}.$$

This improvement is the result of a dominating model where some additional collisions are virtually introduced such that the sequence of collisions is already memoryless ([4]).

**Remark 1.** There is an intermediate version of asynchronous time hopping where the difference of the frame fronts is constrained to any integer multiple of the segment duration. (This is instructive methodologically when similarities and distinctions between time and frequency hopping are to be understood more thoroughly.) Call this case frame asynchronism but segment synchronism. Then the erasure events are memoryless, and the upper bound of the decoding error probability can be derived by combining the first two terms of the bound in Theorem 3, and the third term of the bound in Theorem 2.

**Remark 2.** Obviously one has to investigate the real asynchronous case when there is no slot synchronism. (Notice that positions of the front end of any packet of the tagged message, received successfully, is known. Knowing this, the random address and the serial number of each such packet the decoding of the tagged message is possible. Assume, for simplicity, a draw over just the first  $L - 1$  slots of any segment, leaving the last slot always dummy. Assume any tagged packet erased any time if it is covered partially by a packet from at least one of the other users. Consider one of the following versions in this case:

(i) Use the model just introduced, drawing packet over  $L - 1$  slots per segment. Then any interfering user can have a collision with a tagged packet with probability  $\frac{2}{L-1}$  instead of  $\frac{1}{L-1}$ . Modifying the proof of Theorem 3 accordingly there is a change only for the third term such that under

$$e^{-2} \geq (1 + \delta)R_{sum}$$

the third term is

$$2e^{-(n-1)(e^{-2} - (1+\delta)R_{sum})^2}.$$

Thus the maximum of  $R_{sum}$  is  $e^{-2} = 0.1353\dots$

(ii) Insert a dummy slot after each slot, so double the frame length, and so half  $R_{sum}$ . Modifying the proof of Theorem 3 accordingly again there is a change only for the third term such that under

$$e^{-1} \geq 2(1 + \delta)R_{sum}$$

the third term is

$$2e^{-(n-1)(e^{-1}-2(1+\delta)R_{sum})^2}.$$

Thus the maximum of  $R_{sum}$  is  $e^{-1}/2 = 0.1839\dots$

(iii) Apply the celebrated idea of Massey and Mathys ([10]), when each 1 in the original hop sequence is replaced by 0 followed by  $m-1$  1's, and each 0 is replaced by  $m$  0's ( $m \geq 2$ ). For more see [5]. There it is shown by a reference to the proof of Theorem 3, again for the third term only such that the third term is still exponential in  $n-1$  but the maximum of  $R_{sum}$  is  $(1 - \frac{1}{m})e^{-1}$ .

**Remark 3.** It is of much interest that the coefficient  $n-1$  of the exponent in the third term of the upper bound can actually be increased to  $2n$  in cases (i), (ii) and (iii) of Remark 2 in a similar way as for frame synchronous but slot synchronous situation of Theorem 4. This fact can be proved by a more sophisticated approach, considering a well-defined dominating model, (with erasures generated appropriately under the circumstances in question). For more see [5].

## IV Frequency Hopping

Again the arrivals of the messages are according to a homogeneous Poisson process with intensity  $\lambda$ , if the time unit is the slot. In the model of frequency hopping assume that we have  $L$  frequency slots, which means that there are  $L$  parallel collision channels without feedback. Again each message is encoded by an  $(n, k)$  shortened Reed-Solomon code, and in each of the  $n$  time slots (also called dwell intervals in this case) a frequency slot is chosen randomly (uniformly distributed, and in memoryless way). In the chosen  $n$  time-frequency slots the  $n$  encoded packets are sent, so the  $n$  packets are sent in consecutive time slots according to a random frequency hopping sequence of length  $n$ , which can be interpreted as a time-frequency matrix. Obviously there is a one-to-one correspondence between the time hopping sequence of length  $N = nL$  and the frequency hopping matrix of size  $n \times L$ .

During the blocklength  $n$  the expected number of packets arrived is  $\lambda nk$ , which are sent via a time-frequency matrix of size  $n \times L$  so the throughput (with respect to this matrix) is defined by

$$R_{sum} = \frac{\lambda nk}{nL} = \frac{\lambda k}{L}.$$

The frame asynchronous but slot synchronous frequency hopping corresponds to the segment asynchronous time hopping, therefore according to Remark 1 we get that

**Theorem 5 :** If  $1 > \delta > 0$ ,  $\lambda n(1 + \delta) = L$  and

$$e^{-1} \geq (1 + \delta)R_{sum},$$

then

$$P(\text{decoding error}) \leq ne^{-L\frac{\delta^2}{(1+\delta)^2}\frac{\ln 2}{2}} + 2L^{-(K-1)} + e^{-2n(e^{-1}-(1+\delta)R_{sum})^2}.$$

In the proof of Theorem 5 we use a nonstationary extension of Lemma 1:

**Lemma 4 :** Consider a nonstationary memoryless erasure channel. Let  $X_i$  be 1 if there is an erasure in the  $i$ -th slot, and 0 otherwise such that

$$P(X_i = 1) = p_i.$$

Apply a shortened  $(n, k)$  Reed-Solomon code over  $GF(Q)$  ( $n < Q$ ). Then for  $R = \frac{k}{n}$

$$P(\text{decoding error}) \leq \exp\left(-2n\left(\left[\frac{1}{n} \sum_{i=1}^n (1 - p_i - R)\right]^+\right)^2\right).$$

**Proof of Theorem 5:** Let  $M_i$  be the number of messages arrived in  $[i-n, i]$  and  $M$  be the number of messages arrived in  $[1-n, n]$ , then  $M_i$  is a Poisson random variable with parameter  $\lambda n = L/(1 + \delta)$  ( $i = 1, 2, \dots$ ). Then

$$\begin{aligned} P(\text{decoding error}) &\leq P(\max_{i \leq n} M_i > L) \\ &+ E[I_{[\max_{i \leq n} M_i \leq L]} P(\text{separation error} | M_i, i = 1, \dots, n, M)] \\ &+ E[I_{[\max_{i \leq n} M_i \leq L]} P(\text{decoding error} | \text{correct separation}, M_i, i = 1, \dots, n, M)] \\ &\leq ne^{-L \frac{\delta^2}{(1+\delta)^2} \frac{\ln 2}{2}} + 2L^{-(K-1)} \\ &+ E[I_{[\max_{i \leq n} M_i \leq L]} \exp\left(-2n\left(\left[\frac{1}{n} \sum_{i=1}^n ((1 - \frac{1}{L})^{M_i-1} - \frac{k}{n})\right]^+\right)^2\right)] \\ &\leq ne^{-L \frac{\delta^2}{(1+\delta)^2} \frac{\ln 2}{2}} + 2L^{-(K-1)} + e^{-2n(e^{-1} - (1+\delta)R_{sum})^2}. \end{aligned}$$

**Remark 4.** It is of particular interest to investigate the real asynchronous case also in frequency hopping when there is no slot asynchronism. This can be done following essentially the same principles considered previously for time hopping with frame as well as slot asynchronism (see cases (i), (ii) and (iii) in Remark 2 and Remark 3). Note that for slot asynchronism the sequence of erasures will not be memoryless so apply again Lemma 3 instead of Lemma 4. Use the same notations as in Remark 2 for the corresponding cases:  
(i) Use the coding and accessing as before. Then any interfering user can have a collision with a tagged packet with probability  $\frac{2}{L}$  instead of  $\frac{1}{L}$ . Modifying the proof of Theorem 5 accordingly there is a change only for the third term such that under

$$e^{-2} \geq (1 + \delta)R_{sum}$$

the third term is

$$2e^{-(n-1)(e^{-2} - (1+\delta)R_{sum})^2}.$$

Thus the maximum of  $R_{sum}$  is  $e^{-2} = 0.1353\dots$  also in this case.

(ii) Insert a dummy slot (dwell interval) after each slot (*i.e.* dwell interval) also for frequency hopping, so double the frame length, and so half  $R_{sum}$ . Modify the proof of Theorem 5 accordingly. Again there is a change only for the third term such that under

$$e^{-1} \geq 2(1 + \delta)R_{sum}$$

the third term is

$$2e^{-(n-1)(e^{-1}-2(1+\delta)R_{sum})^2}.$$

Thus the maximum of  $R_{sum}$  is  $e^{-1}/2 = 0.1839\dots$

(iii) Apply the celebrated idea of Massey and Mathys ([10]), also for frequency hopping, when 1 in the original hop matrix is replaced by a 0 followed by  $m - 1$  1's, and each 0 is replaced by  $m$  0's ( $m \geq 2$ ). For more see [5]. There it is shown, by a slight change in the proof of Theorem 5 for the third term only, that the third term is still exponential in  $n - 1$  but the maximum of  $R_{sum}$  is  $(1 - \frac{1}{m})e^{-1}$ .

**Remark 5.** The coefficient  $n - 1$  of the exponent in the third term of the upper bound can be increased to  $2n$  in cases (i), (ii) and (iii) of Remark 4 in a similar way as in Remark 3. This can be proved by a more sophisticated approach, considering a well-defined dominating model, (with erasures generated appropriately under the circumstances in question). See [5].

## V Conclusions

Considering a Poisson (potential) user population, exponential upper bounds have been given on the decoding error probability contributions due to overflow, identification error (under no overflow) and due to hits (under neither overflow nor identification error).

All basic versions of synchronous and asynchronous access have been considered, including cases of actual as well as those of methodologic interest.

There is a rich literature available on error probability bounds, particularly for frequency hopping. A recent example of a key paper is by Frank and Pursley ([6]) in which even two-sided bounds are given. A quite recent approach to the analysis for various versions of frequency hopping is by Mohamed and Pap ([10]), referring to earlier papers of interest. However, the authors are not aware of any previous result on exponential bounds on the decoding error probability, considered in the present paper (and in [3], [4], [5]); *viz.*, for infinite user population, any finite source blocklength; and relying upon an asymptotically tight lower bound of the capacity of the erasure channel of the underlying original and that of the modified model, respectively.

A capacity equal to  $e^{-1}$  for basic kinds of multiaccess as the message length tends to infinity, pointed out by Massey, has been referred to at the outset of this paper. We enjoyed also finding a further important application to a witty approach by Massey and Mathys for reducing slot asynchronism to slot synchronism. Notice that our investigations, concerning the relevant fact that essentially the same upper bound can be given on the decoding error probability for all considered kinds of random time as well as frequency hopping, is completed (for both cases) by this reduction ([9]).

We conclude with these recollections, our presentation. Happy Birthday to you, Jim.

## References

- [1] N. Q. A, L. Györfi, and J. L. Massey, “Constructions of binary constant-weight cyclic codes and cyclically permutable codes” *IEEE Trans. on Information Theory*, 38, pp. 940-949, 1992.

- [2] L. A. Bassalygo and M. S. Pinsker, "Limited multiple-access of a nonsynchronous channel", (in Russian) *Problems of Information Transmission*, XIX, pp. 92-96, 1983.
- [3] S. Csibi and L. Györfi, "Coding for multiple access collision channel without feedback and with Poisson arrivals", *Proc. 6th Joint Swedish-Russian International Workshop on Information Theory*, pp. 72-75, Mölle, Sweden, August 22-27, 1993.
- [4] S. Csibi and L. Györfi, "Exponential bounds and dominating models for CDMA" *12th Prague Conference on Information Theory, Statistical Decision Functions*, 1994, (submitted).
- [5] S. Csibi and L. Györfi, "More on exponential bounds and dominating models for CDMA", *IEEE International Workshop on Information Theory*, Aksakovo-Moscow, Russia, 1994, (submitted).
- [6] F. D. Frank and M. B. Pursley, "On the statistical dependence of hits in frequency-hopping multiple access", *IEEE Trans. Communications*, COM-38, pp. 1483-1494, 1990.
- [7] W. Hoeffding, "Probability inequalities for sums of bounded random variables", *Journal of the Amer. Stat. Assoc.*, 58, pp. 13-30, 1963.
- [8] J. L. Massey, "The capacity of the collision channel without feedback", Abstracts of Papers. IEEE Int. Symp. on Info. Theory, p.101, 1982.
- [9] J. L. Massey and P. Mathys, "The collision channel without feedback", *IEEE Trans. on Information Theory*, IT-31, pp.192-204, 1985.
- [10] K. A. Mohamed and L. Pap, "Analysis of frequency-hopped packet radio networks with random signal levels" *IEEE Trans. Selected Area in Communications*, 1994 (to appear).
- [11] N. Pippenger, "Bounds on the performance of the protocols for a multiple-access broadcast channel", *IEEE Trans. on Information Theory*, IT-27, pp.145-151, 1981.
- [12] B. S. Tsybakov and N. B. Likhanov, "Packet communication on a channel without feedback", *Problems of Information Transmission*, XIX, pp. 69-84, 1983.

# On Repeated-Single-Root Constacyclic Codes

Valdemar C. da Rocha Jr.

Department of Electronics and Systems  
Federal University of Pernambuco  
50741-540 Recife PE BRASIL

## Abstract

A new derivation is presented for the minimum Hamming distance of a class of  $p^r$ -ary maximum distance separable constacyclic codes ( $n = p, k, d = p - k + 1$ ), where  $p$  is a prime and  $r$  is a positive integer, introduced by Massey, Costello, and Justesen in 1973. These are repeated-single-root constacyclic codes generated by the polynomial  $g(x) = (x - \alpha)^{p-k}$ ,  $1 \leq k < p$ ,  $\alpha \neq 0$ ,  $\alpha \in GF(p^r)$ . As a by-product of the derivation for the minimum Hamming distance these codes are shown to be equivalent to shortened generalized Reed-Solomon codes. An application of these codes is suggested for secret-key cryptosystems.

## I Introduction

In 1973 Massey, Costello, and Justesen [1] wrote a paper presenting algebraic constructions of various classes of both block and convolutional codes, based on the *weight-retaining* property of the polynomials  $(x - \alpha)^i$ ,  $i = 0, 1, 2, \dots$ , where  $\alpha$  is any nonzero element of  $GF(q)$ , that any linear combination of these polynomials with coefficients in  $GF(q)$  has Hamming weight not less than that of the minimum degree polynomial included. We note that the above mentioned constructions made exclusive use of repeated-single-root polynomials.

In Section II we give an alternative derivation of the minimum Hamming distance of a class of  $p^r$ -ary repeated-single-root constacyclic maximum distance separable (MDS) codes first given in [1], making use of the parity-check matrix for these codes, along the lines of [3]. In Section III we suggest an application of such MDS constacyclic codes for secret-key cryptosystems where the secret key consists of a private  $p^r$ -ary random source and a permutation  $P$  of  $p$ -ary  $nr$ -tuples.

## II Algebraic Characterization

A polynomial  $g(x)$  over  $GF(p^r)$  of degree  $n - k$ , which divides  $x^n - \alpha$ , where  $\alpha \neq 0$ ,  $\alpha \in GF(p^r)$ , generates a  $(n, k)$  *constacyclic* code [2, p.303], i.e., a code whose  $p^{rk}$  codewords are all the multiples of  $g(x)$  reduced modulo  $x^n - \alpha$ . If  $\mathbf{c} = (c_0, c_1, \dots, c_{n-1})$  is a codeword of such a code then its constacyclic shift,  $(\alpha c_{n-1}, c_0, c_1, \dots, c_{n-2})$ , is also a codeword. The code is cyclic if and only if  $\alpha = 1$  and is *negacyclic* [2, p.211] if and only if  $\alpha = -1$ .

In [3] Massey and three of his students developed further the theory of cyclic codes with repeated roots. One particularly interesting result in that paper was the *correct* construction of a parity-check matrix for  $p^r$ -ary repeated-root cyclic codes. When we refer to the *correct* construction we mean a construction that does not fail when some root of the code generator polynomial  $g(x)$  has multiplicity  $p$  or greater. Curiously enough, apart from presenting its construction, no further use of the parity-check matrix of repeated-root cyclic codes was made in [3]. For our convenience we will phrase these results in terms of constacyclic codes and will essentially reproduce the proof given in [3] for the sake of completeness. As we show below, the use of a parity-check matrix for a class of repeated-single-root constacyclic codes allows a simple derivation of their minimum Hamming distance.

**Theorem 1** *Let  $g(x)$  generate a  $p^r$ -ary  $(n, k)$  constacyclic code  $\mathbf{C}$ . Let  $GF(p^r)$  be the splitting field of  $g(x)$ . Then the matrix  $H$  having as rows the  $n$ -tuples*

$$\left[ \binom{n-1}{j} \alpha^{n-1}, \binom{n-2}{j} \alpha^{n-2}, \dots, \binom{1}{j} \alpha, \binom{0}{j} \right] \quad (1)$$

where  $\alpha$  is in the root set of  $g(x)$  with multiplicity  $e$  and  $0 \leq j < e$ , is a parity-check matrix for  $\mathbf{C}$ .

**Proof:** The  $p^r$ -ary polynomial  $c(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \dots + c_0$  is a code polynomial if and only if, given that  $\alpha$  is a root of  $g(x)$  with multiplicity  $e$ , its  $j^{\text{th}}$  Hasse derivative  $c^{[j]}(x)$  [3],  $0 \leq j < e$ , is zero for  $x = \alpha$ , i.e.,

$$c^{[j]}(\alpha) = 0, \quad \text{for } 0 \leq j < e \quad (2)$$

where

$$c^{[j]}(x) = \sum_i \binom{i}{j} c_i x^{i-j}. \quad (3)$$

The left side of (2) can be expressed in matrix form as

$$\begin{aligned} & [c_{n-1}, c_{n-2}, \dots, c_0] \cdot \\ & \left[ \binom{n-1}{j} \alpha^{n-1}, \binom{n-2}{j} \alpha^{n-2}, \dots, \binom{1}{j} \alpha, \binom{0}{j} \right]^T, \end{aligned} \quad (4)$$

where the superscript T denotes transpose. Therefore (2) has the meaning that the  $n$ -tuples

$$\left[ \binom{n-1}{j} \alpha^{n-1}, \binom{n-2}{j} \alpha^{n-2}, \dots, \binom{1}{j} \alpha, \binom{0}{j} \right], \quad \text{for } 0 \leq j < e,$$

are orthogonal to every codeword of  $\mathbf{C}$ . That the matrix  $H$  is indeed a parity-check matrix for  $\mathbf{C}$  follows from the fact that  $[c_{n-1}, c_{n-2}, \dots, c_0]$  is a codeword of  $\mathbf{C}$  if and only if (2) holds for every root  $\alpha$  of  $g(x)$ , with multiplicity  $e$ , and for every  $j$ ,  $0 \leq j \leq e - 1$ .  $\square$

The theorem below was first stated and proved by Massey *et al* [1]. They introduced a class of  $p^r$ -ary repeated-single-root constacyclic codes with the observation that the cyclic codes in this class were given earlier by Assmus and Mattson [4] and by Berman [5]. We shall now present an alternative proof of this result which is based on a parity-check matrix

for these codes. For this particular class of codes, since their blocklength is  $n = p$ , the multiplicity of any root of a generator polynomial is necessarily less than  $p$  for nontrivial codes and thus a parity-check matrix based on formal derivatives could be used. We prefer however to use a form of parity-check matrix which is always correct, independent of the root multiplicity.

**Theorem 2 ([1, Theorem 5])** *The polynomial  $g(x) = (x - \alpha)^{p-k}$  for  $1 \leq k < p$  generates a  $p^r$ -ary ( $n = p, k, d$ ) constacyclic code with  $d = p - k + 1$  (i.e., a maximum distance separable code), where  $\alpha \in GF(p^r)$  and  $\alpha \neq 0$ .*

**Proof:** We note first that  $g(x) = (x - \alpha)^{p-k}$  for  $1 \leq k < p$  divides  $x^p - \alpha^p = (x - \alpha)^p$  and thus generates a constacyclic code of blocklength  $n = p$ . By Theorem 1, a parity-check matrix  $H'$  for such a code is given by

$$H' = \begin{bmatrix} \alpha^{n-1} & \alpha^{n-2} & \cdots & \alpha & 1 \\ \binom{n-1}{1}\alpha^{n-1} & \binom{n-2}{1}\alpha^{n-2} & \cdots & \binom{1}{1}\alpha & 0 \\ \binom{n-1}{2}\alpha^{n-1} & \binom{n-2}{2}\alpha^{n-2} & \cdots & \binom{1}{2}\alpha & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \binom{n-1}{e-1}\alpha^{n-1} & \binom{n-2}{e-1}\alpha^{n-2} & \cdots & \binom{1}{e-1}\alpha & 0 \end{bmatrix},$$

where we adopt the convention that  $\binom{i}{j} = 0$  if  $i < j$ . We can write  $H' = X'Y$ , i.e., as a product of two matrices, as follows

$$X'Y = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ \binom{n-1}{1} & \binom{n-2}{1} & \cdots & \binom{1}{1} & 0 \\ \binom{n-1}{2} & \binom{n-2}{2} & \cdots & \binom{1}{2} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \binom{n-1}{e-1} & \binom{n-2}{e-1} & \cdots & \binom{1}{e-1} & 0 \end{bmatrix} \begin{bmatrix} \alpha^{n-1} & & & & \\ & \alpha^{n-2} & & & \\ & & \alpha^{n-3} & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}.$$

We note that  $Y$  is nonsingular since  $\alpha \neq 0$ ,  $\alpha \in GF(p^r)$ , thus implying  $\alpha^i \neq 0$ ,  $0 \leq i \leq n-1$ . Also, by *Theorem A.1* (in the Appendix), since  $n = p$ , it follows that any  $e$  columns of  $X'$  are linearly independent, i.e., the determinant of any square matrix formed by selecting any  $e$  columns of  $X'$  is nonzero. We thus conclude that any  $e$  columns of  $H'$  are linearly independent and consequently that the minimum Hamming distance of this code satisfies  $d \geq e + 1 = p - k + 1 = n - k + 1$ . However, from the Singleton bound [6, p.33] we know that  $d \leq n - k + 1$ . We thus conclude that  $d = n - k + 1$ .  $\square$

Using the notation developed in the Appendix, we can write  $H'$  as  $H' \prod_m (m!) =$

$AXY$ ,  $1 \leq m \leq e - 1$ , where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1e} \\ a_{21} & a_{22} & \cdots & a_{2e} \\ \vdots & \vdots & & \vdots \\ a_{e1} & a_{e2} & \cdots & a_{ee} \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 \\ n-1 & n-2 & \cdots & 2 & 1 & 0 \\ (n-1)^2 & (n-2)^2 & \cdots & 2^2 & 1^2 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ (n-1)^{e-1} & (n-2)^{e-1} & \cdots & 2^{e-1} & 1^{e-1} & 0^{e-1} \end{bmatrix}.$$

We note that  $A$  is nonsingular because  $a_{ii} = 1$ ,  $1 \leq i \leq e$  and  $a_{ij} = 0$  if  $i < j$ . Therefore, an equally good parity-check matrix for these codes is the matrix  $H = XY$  [6, p.335]. Clearly  $H = XY$ , as described in [6, pp.333-334], is a parity-check matrix of a shortened generalized Reed-Solomon code. Therefore, the codes given in [1, *Theorem 5*] are equivalent to shortened  $p^r$ -ary generalized Reed-Solomon codes.

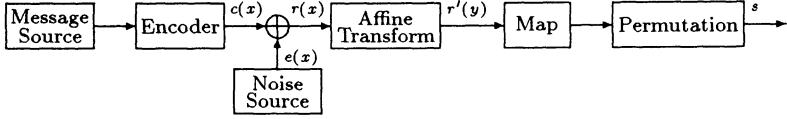
### III An Application to Cryptography

In this section we consider a possible application of repeated-single-root constacyclic codes to secret-key cryptography. We note that a  $p^r$ -ary constacyclic code  $\mathbf{C}$  ( $n = p, k, d = n - k + 1$ ), with generator polynomial  $g(x) = (x - \alpha)^{p-k}$  is transformed into a *degenerate*  $p^r$ -ary code  $\mathbf{C}'$  ( $n = p, k, d' = 1$ ) by the affine transformation  $y = x - \alpha$ . The codewords of  $\mathbf{C}'$  are  $p^r$ -ary  $n$ -tuples of the form  $(c'_{p-1}, c'_{p-2}, \dots, c'_{p-k}, 0, 0, \dots, 0)$ , i.e., their lowest order  $p - k$  coordinates are zeros. Clearly,  $\mathbf{C}'$  has no useful error-correcting power. Summarizing, the affine transformation  $y = x - \alpha$  changes a MDS  $p^r$ -ary constacyclic code  $\mathbf{C}$ , with generator polynomial  $g(x) = (x - \alpha)^{p-k}$ , into a code  $\mathbf{C}'$  with minimum Hamming distance one. For a given finite field  $\text{GF}(p^r)$ , according to Theorem 2, we have  $p^r - 1$  distinct codes  $\mathbf{C}$  ( $n = p, k, d = p - k + 1$ ), i.e., we have  $p^r - 1$  distinct choices for  $\alpha \neq 0$  in  $g(x) = (x - \alpha)^{p-k}$ , and all such codes are mapped to the same code  $\mathbf{C}'$  ( $n = p, k, d = 1$ ) by the respective affine transformation  $y = x - \alpha$ .

Suppose that  $c(x)$  is a codeword of code  $\mathbf{C}$  and suppose that  $e(x) = \sum_i e_i x^i$ ,  $0 \leq i \leq n - 1$ , is a nonzero random error pattern with at most  $t$  nonzero  $p^r$ -ary components, i.e., a pattern of Hamming weight at most  $t$ ,  $0 < t \leq \lfloor (p - k + 1)/2 \rfloor$ , where  $\lfloor x \rfloor$  denotes the integer part of  $x$ . Let  $r(x) = c(x) + e(x)$ , with addition as defined for the additive group of  $\text{GF}(p^r)$ . By applying the affine transformation  $y = x - \alpha$  to  $r(x)$  we obtain  $r'(y) = c'(y) + e'(y)$ , where  $c'(y)$  is a codeword of code  $\mathbf{C}'$  and  $e'(y) = \sum_i e_i (y + \alpha)^i$ . Let  $i_{\min}$  denote the minimum value of  $i$ ,  $0 \leq i \leq n - 1$ , for which  $e_i \neq 0$ . Then, by Theorem 6.1 of [1, p.109] and by Lemma 1 [1, p.102] we have for the Hamming weight  $W_H[e'(y)]$  of  $e'(y)$  that

$$W_H[e'(y)] \geq W_H[(y + \alpha)^{i_{\min}}] = i_{\min} + 1. \quad (5)$$

We interpret (5) as meaning that very often we will have  $W_H[e(x)] < W_H[e'(y)]$ .



The above considerations suggest the use of repeated-single-root constacyclic codes for secret-key cryptosystems where the secret-key would consist of  $\alpha$  (in principle), a noise source (private random source) and a randomly chosen (but fixed) permutation  $P$  applied to  $p$ -ary  $nr$ -tuples as illustrated by the above block diagram.

We assume that the source emits (preferably statistically independent and uniformly distributed) symbols (*plaintext*) from  $GF(p^r)$  which are fed to the encoder for code **C**. The noise source (private random source) adds a nonzero random error pattern of Hamming weight at most  $t = \lfloor (p - k + 1)/2 \rfloor$  to the codeword  $c(x)$ . The resulting  $n$ -tuple,  $r(x)$ , is changed into  $r'(y)$  by the affine transformation  $y = x - \alpha$ . Each symbol of the  $n$ -tuple  $r'(y)$  is then mapped one-to-one to  $r$   $GF(p)$  symbols. Finally, a permutation  $P$  is applied to the resulting  $p$ -ary  $nr$ -tuple. The permuted sequence  $S$ , of length  $nr$ , represents the cryptogram which is then ready for transmission or storage.

The deciphering operation by an authorized party consists of inverting the permutation  $P$ , then reverse map the  $n$   $p$ -ary  $r$ -tuples into  $n$   $p^r$ -ary symbols and finally applying the algebraic decoding algorithm developed in [1, *Theorem 5*] for these codes to correct the error pattern  $e(x)$ , i.e., to remove the effect of the private random source, and reproduce the original cleartext. We remark that this decoding algorithm deals directly with  $r'(y)$  thus avoiding the need for an inverse affine permutation. The reason for considering the noise source as a private random source should now be clear because the decoder does not need to know in advance which error pattern was added to the transmitted codeword, in order to recover the cleartext.

A *chosen plaintext attack* by an enemy cryptanalyst, i.e., by someone who does not know either  $e(x)$  nor  $P$ , may succeed in finding  $\alpha$  and the location of the information positions in the intercepted ciphertext as we now explain. That is why  $\alpha$  does not need to be kept secret in principle. However, the disclosure of  $\alpha$  by the enemy cryptanalyst is not yet enough to break the system.

Let  $m_0 \in GF(p^r)$  and let  $m_0 = m_{0,r-1}\beta^{r-1} + m_{0,r-2}\beta^{r-2} + \dots + m_{0,1}\beta + m_{0,0}$  be the  $p$ -ary representation of  $m_0$ , where  $\beta$  is a primitive element of  $GF(p^r)$  and  $m_{0,i} \in GF(p)$ ,  $0 \leq i \leq r-1$ . By repeatedly feeding chosen messages  $m(x) = m_0$  (which correspond to codewords  $m_0(x - \alpha)^{p-k}$ ) such that only one of the coordinates of  $m_0$ , say  $m_{0,i}$ , is equal to 1,  $0 \leq i \leq r-1$ , and all other coordinates  $m_{0,j}$ ,  $j \neq i$ , are equal to zero, the enemy cryptanalyst identifies  $m_0y^{p-k}$  in the intercepted ciphertext by majority voting. In an analogous manner, the enemy cryptanalyst continues repeatedly feeding messages  $m(x) = m_1x$  and, again using majority voting, identifies  $\alpha$  and the location of  $y^{p-k+1}$  in the ciphertext, since  $m(x)g(x) = m_1x(x - \alpha)^{p-k}$  is changed by the affine transformation into  $m_1(y + \alpha)y^{p-k} = m_1y^{p-k+1} + m_1\alpha y^{p-k}$  and the location of  $y^{p-k}$  in the ciphertext is known from the previous step. In this manner, by repeatedly feeding  $m(x) = m_i x^i$ ,  $0 \leq i \leq k-1$ , the enemy cryptanalyst finds  $\alpha$  and the positions in the ciphertext occupied by sums of information digits, i.e., the location in the ciphertext of the powers of  $y$  obtained from  $\sum_i m_i(y + \alpha)^i$ ,  $0 \leq i \leq k-1$ . The resulting linear system of equations would be solvable if noise were not present ( $k$  equations

in  $k$  unknowns). The presence of noise makes it necessary for the enemy cryptanalyst to identify the portion of the permutation  $P$  affecting the  $r(p - k)$   $p$ -ary parity-check symbols in the received  $nr$ -tuple in order to be able to employ the known error-correcting code. We note, however, that these  $r(p - k)$  places do not leak information on how they were permuted since in all codewords  $c'(y)$  they are always occupied by zeros, no matter which message is chosen for the attack. We are assuming that the system parameters are chosen in such way that exhaustive searches are not feasible. We leave open for the potential enemy cryptanalyst the challenging task of finding an efficient way to process the received  $p$ -ary  $nr$ -tuple in order to extract the cleartext.

## IV Acknowledgement

The author wishes to take this opportunity to acknowledge with great pleasure the extremely useful experience of learning more about coding and cryptography through many *little discussions* with Jim Massey, who would always find time in spite of being ever so overcommitted to various activities, during the period spent at the Signal and Information Processing Laboratory of the Swiss Federal Institute of Technology-Zurich.

This work received partial support from the Brazilian National Council for Scientific and Technological Development (CNPq) under the grant No.304214/77-9.

## A Appendix

We note that over the integer field the binomial coefficient  $\binom{x}{m}$ ,  $x \geq m$ , can always be written as a degree  $m$  polynomial in  $x$ , since by definition

$$\binom{x}{m} = x(x-1)(x-2)\cdots(x-m+1)/m! , \quad (6)$$

where  $m!$  denotes the factorial of  $m$ , i.e.,  $m! = m(m-1)(m-2)\cdots 2 \cdot 1$ , with the convention that  $0! = 1$ . Performing the product on the right of (6) we obtain

$$m! \binom{x}{m} = x^m + a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0.$$

Therefore, it follows that any square matrix  $X'$  having as rows

$$\left[ \binom{x_1}{j}, \binom{x_2}{j}, \dots, \binom{x_e}{j} \right], \quad 0 \leq j \leq e-1,$$

where  $x_1, x_2, \dots, x_e$  are integers, after some elementary row operations can be written as the product of a constant and the Vandermonde matrix having as rows

$$\left[ x_1^j, x_2^j, \dots, x_e^j \right], \quad 0 \leq j \leq e-1. \quad (7)$$

As it is well known, the value of the determinant associated with the above Vandermonde matrix is given by  $\prod_{i>j}(x_i - x_j)$ ,  $0 \leq i, j \leq e-1$ , and of course is nonzero if and only if  $x_i \neq x_j$ ,  $i > j$ ,  $0 \leq i, j \leq e-1$ . Over a finite field  $GF(p)$ , this result is equivalently stated as  $\prod_{i>j}(x_i - x_j) \neq 0$ ,  $0 \leq i, j \leq e-1$ , if and only if  $x_i \not\equiv x_j \pmod{p}$ ,  $i > j$ ,  $0 \leq i, j \leq e-1$ , where  $x_i \in GF(p)$ ,  $0 \leq i \leq e-1 < p$ . The following theorem summarizes these facts.

**Theorem A.1** Over a finite field  $GF(p)$ , where  $p$  is a prime, the determinant of any square matrix  $X'$  having as rows

$$\left[ \binom{x_1}{j}, \binom{x_2}{j}, \dots, \binom{x_e}{j} \right], \quad 0 \leq j \leq e-1 < p,$$

where  $x_1, x_2, \dots, x_e \in \{0, 1, 2, \dots, p-1\}$ , is nonzero if and only if  $x_i \not\equiv x_j \pmod{p}$ ,  $i > j$ ,  $0 \leq i, j \leq e-1 < p$ .

## References

- [1] J.L. Massey, D.J. Costello, Jr., and J. Justesen, “Polynomial weights and code constructions”, *IEEE Trans. Inform. Theory*, vol. IT-19, pp.101-110, 1973.
- [2] E.R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
- [3] G. Castagnoli, J.L. Massey, P.A. Schoeller, and N. von Seemann, “On repeated-root cyclic codes”, *IEEE Trans. Inform. Theory*, vol. IT-37, pp.337-342, 1991.
- [4] E.F. Assmus, Jr. and H.F. Mattson, Jr., “New 5-designs”, *J. Combinatorial Theory*, vol.6, pp.122-151, 1969.
- [5] S.D. Berman, “On the theory of group codes”, *Kibernetika*, vol.3, pp.31-39, 1967.
- [6] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*. New-York: North-Holland, 1978.

# Orthogonal Checksets in the Plane and Enumerations of the Rationals mod $p$ .

Peter Elias  
Massachusetts Institute of Technology  
Cambridge, MA

## Abstract

Binary linear codes with wordlength  $p^2$  for prime  $p$  and minimum distance at least  $2s$  and at most  $2p$  are constructed using as check sets all  $sp$  of the  $p$ -bit lines in  $G(p)$ , a finite plane geometry mod  $p$ , which have any one of  $s$   $p$ -distinct slopes. Minimum distances greater than  $2s$  seem attainable for some sets of  $s$  slopes, but  $2p$  is a firm upper bound. A norm on the rationals mod  $p$  allows choices of  $s$   $p$ -distinct slopes which remain  $p'$ -distinct and keep minimum distance for all prime  $p' > p$ .

## I Introduction

This paper is dedicated to Jim Massey on his 60th birthday. Like Massey in “Threshold decoding” [3] it constructs codes with a large number of checksets having an orthogonality property: each pair has at most one bit in common. It uses the finite geometries introduced to coding theory by Marcel Golay, another distinguished code constructor who divided his time between the United States and Switzerland, in 1949, the second year of the information theory era. [1].

Some years before Massey’s thesis I designed a code using checksets having the orthogonality property. Unlike his more elegant algebraic construction, mine was brute force. I added a new dimension for each new group of orthogonal checksets, leading to arbitrarily reliable communication at a positive rate over a binary symmetric [2] or binary erasure [4] channel but requiring huge codewords. This is a preliminary report on codes using more than two orthogonal families of checksets in the plane, preserving orthogonality by using a finite geometry. For simplicity and ease of visualisation I deal only with geometries modulo a prime  $p$  and square arrays. To save space I give comments rather than complete proofs of most results: like a proof, a comment is terminated by  $\blacksquare$ . The exploration is incomplete, and at best cannot lead to planar codes with very good minimum distance properties for large  $p$  (see Lemma 5 below), but it does lead to some pretty pictures and interesting relations between geometries mod  $p$  and discrete, finite Euclidean structures.

## II Geometry Modulo an Odd Prime $p$ and Finite Euclidean Structures

Consider the field over the set  $J_p$  of nonnegative integers less than an odd prime  $p$ . The cartesian product  $J_p \times J_p = J_p^2$  is represented by the square of black dots in Figure 1 for  $p = 7$ . A pair  $\pi = (i, j)$  in  $J_p^2$  is a point in that square and is also a vector, from the origin  $(0, 0)$  to  $\pi$ , so addition of pairs and their multiplication by constants in  $J_p$  ( $\text{mod } p$ ) are defined. We consider the geometry of the points and lines in  $J_p^2$ , which constitute a coordinatized finite affine geometry [6] denoted by  $G(p)$ , and also a discrete Euclidean structure  $E(p)$  defined on the same domain.

Equivalence in  $G(p)$  is denoted by  $\equiv_p$ , as in  $-2 \equiv_p p - 2$ ,  $1/2 \equiv_p (p+1)/2$ .  $p$ -equivalence is extended to set  $j/i \equiv_p \infty$  if  $i \equiv_p 0, j \not\equiv_p 0$ . The slope of the vector from the origin to  $(i, j)$  is the rational  $j/i$ : the slope of the null vector  $(0, 0)$  is not defined.

In Figure 1 circles surround 7 points in the lattice  $J_7^2$  of black dots. They are connected by line segments of rational slopes  $i/j$ , where  $0 < i, j < p$ . It is not the line segment itself but the set of points it connects which is of interest here and is (all or part of) a line: the visible line segment serves only to identify that set of points.

If we take the figure to represent the discrete Euclidean structure denoted by  $E(7)$  on  $J_7^2$ , it contains twelve sets of two or more circled points each, one or more sets for each of the 7 slopes in  $S = \{5, -2, 1/3, 3/2, -1/4, 4/5, -3/5\}$ .

Instead we may take the figure to represent the 7 circled dots which constitute the single line whose equation is  $j \equiv_7 5i$  in  $G(7)$ , the finite plane geometry mod 7 on  $J_7^2$ . In  $G(7)$ , in fact, all seven circles satisfy the equation  $j \equiv_7 \sigma i$ , where  $\sigma$  is any one of the rational slopes in  $S$ : each  $\sigma \in S$  yields the same 7 points in  $J_7^2$ .

Finally, we may take the figure to represent the lower left corner of  $J_p^2$ , an array of much larger prime dimension  $p' > 41$ . While all of the 7 rational slopes in  $S$  are 7-equivalent, by Lemma 7 below no two of them are  $p'$ -equivalent for  $p' > 41$ , so the figure has the same interpretation in both  $E(7)$  and, for example,  $E(43)$  and  $G(43)$ .

Some obvious but useful properties of  $G(p)$  and  $E(p)$  are given for reference in Lemma 1. A definition is useful first. Lines of  $p$ -equivalent slope in  $G(p)$  and of equal slope in  $E(p)$  are *parallel*. The set of all parallel lines is a *family*, and may be called a  $\sigma$ -family if its lines have slope  $p$ -equivalent (in  $G(p)$ ) or equal (in  $E(p)$ ) to  $\sigma$ .

**Lemma 1** *Let  $G(p)$  be a finite affine geometry and  $E(p)$  a finite Euclidean structure on  $J_p^2$  for some prime  $p$ . Let  $J_p = \{0, 1, \dots, p-1\}$ ,  $J_p^+ = J_p \cup \{\infty\}$ ,  $R_p = \{j/i : 0 < i < p, -p < j < p\}$ , and  $R_p^+ = R_p \cup \{\infty\}$ .*

- i. *Each pair  $(\sigma, \pi)$  of a slope in  $R_p^+$  and a point in  $J_p^2$  defines unique lines  $\lambda_{G(p)}(\sigma, \pi)$  and  $\lambda_{E(p)}(\sigma, \pi)$ , both of slope  $\sigma$ , both including  $\pi$ , given by*

$$\begin{aligned}\lambda_{G(p)}(\sigma, (i, j)) &= (i, j) \cup \{(i', j') \in J_p^2 - (i, j) : (j' - j)/(i' - i) \equiv_p \sigma\}, \\ \lambda_{E(p)}(\sigma, (i, j)) &= (i, j) \cup \{(i', j') \in J_p^2 - (i, j) : (j' - j)/(i' - i) = \sigma\}.\end{aligned}$$

- ii. *Each pair  $\{\pi = (i, j), \pi' = (i', j')\}$  of distinct points in  $J_p^2$  defines a slope  $\sigma = (j' - j)/(i' - i)$  and unique lines  $\lambda_{G(p)}(\sigma, \pi)$ ,  $\lambda_{E(p)}(\sigma, \pi)$ , which contain both  $\pi$  and  $\pi'$ .*

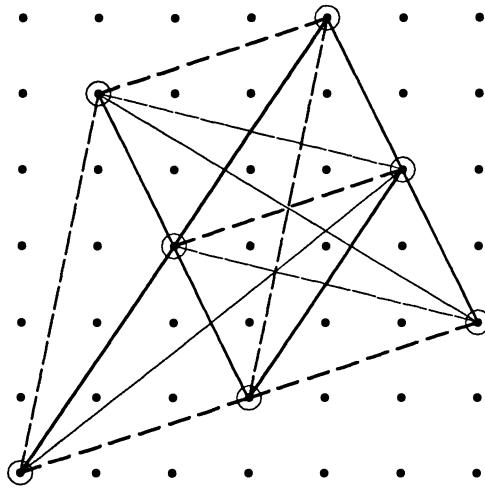


Figure 1: The dots show  $J_p^2$  for  $p = 7$ . There are circles at the 7 points in the line  $j \equiv_7 5i$  in  $G(7)$ , two of which fall on each of the two dashed lines of that slope in the Euclidean plane. The other lines show the points lying on the 7-equivalent slopes  $-2, 1/3, 3/2, -1/4, 4/5, -3/5$  in the Euclidean plane.

iii. Parallel lines have no points in common, in either  $G(p)$  or  $E(p)$ . In  $G(p)$  a pair  $\lambda_{G(p)}(\sigma, \pi), \lambda_{G(p)}(\sigma', \pi')$  of lines which are not parallel define a unique point  $(i, j)$  of intersection. Since they are not parallel, it is possible to label the pair so that  $\sigma \not\equiv_p \infty, \sigma' \not\equiv_p 0$ . Then

$$(i, j) \equiv_p \left( \frac{i' + (j - j' - \sigma i)/\sigma'}{1 - \sigma/\sigma'}, \frac{j + \sigma(i' - i - j'/\sigma')}{1 - \sigma/\sigma'} \right)$$

In  $E(p)$ , two lines which are not parallel intersect at most once, but possibly not at all, in  $J_p^2$ .

iv. In  $G(p)$  there are  $p + 1$  families, one for each  $p$ -equivalence class of slopes in  $R_p^+$ . Each family partitions  $J_p^2$  into  $p$  disjoint parallel lines, each line containing  $p$  points. Thus there are  $p^2$  points and  $p(p + 1)$  lines in  $G(p)$ .

In  $E(p)$  there are  $4|F_{p-1}| + 1$  families, one for each distinct slope in  $R_p^+$ , where  $F_n$  is the size of the  $n$ th Farey series.

**Comments.** In (i) the functions  $\lambda_{G(p)}, \lambda_{E(p)}$  are many-one: lines have many names, especially in  $G(p)$ . Replacing  $\pi$  by any  $\pi'$  in either line does not change the points in that line, and changing  $\sigma$  to any  $\sigma' \equiv_p \sigma$  does not change the value of  $\lambda_{G(p)}$ . This (nonstandard) notation is useful in dealing simultaneously with  $G(p)$  and  $E(p)$ .

The discrete structure  $E(p)$  is not a plane geometry in the sense of Euclidean plane geometry or  $G(p)$ , since nonparallel lines need not intersect and some lines have only one point: for example a line through  $(0,0)$  with a negative slope.

In (iv) the size of the Farey series is the number of irreducible positive proper fractions with denominators at most  $n$  [5]. The factor 4 arises since both positive and negative proper fractions and their reciprocals are included in  $R_p$ : the added 1 is the infinite slope in  $R_p^+$ . ■

### III Codes Using $\sigma$ -families as Checksets: Thickness and Minimum Distance

A linear code can be constructed by using as parity check sets all lines in  $s$  families with  $p$ -distinct slopes. Such a code with  $s = 2$ , whose checksets are for example the rows and the columns of  $J_p^2$ , obviously has minimum distance 4. Its  $2p$  check vectors are not linearly independent: since each family partitions  $J_p^2$ , each set of constraints requires that the parity of all  $p^2$  bit values in  $J_p^2$  be even. Lemma 2 gives an independent subset, which can be put in systematic form.

**Lemma 2** *Let  $S$  be a set of  $s$   $p$ -distinct slopes. Delete one line from each of the  $s$  families with slopes in  $S$ , construct the checksets for all of the remaining  $s(p - 1)$  lines and add the global checkset which checks all bits in  $J_p^2$ . Then those  $1 + s(p - 1)$  checkvectors are linearly independent for all  $s \leq p + 1$ .*

**Comment.** By Lemma 1(iii) the  $p + 1$  lines intersecting in  $(i, j)$  are disjoint except for their common intercept. Since  $p + 1$  is even, the corresponding  $p^2$ -bit check vectors sum mod 2 to the vector with 1's everywhere except at  $pi + j$ . Adding that sum to the all-1 check vector gives just the  $(pi + j)$ th basis vector, showing that the given  $p^2$  check vectors span the space. ■

Although  $s$  families with  $p$ -distinct slopes are orthogonal in the sense of Lemma 1(iii), minimum distance does not grow like  $2^s$ . The distance properties of codes with  $s > 2$  families of checksets are not obvious. The minimum distance of these codes is closely related to *thickness*, a geometric property of point sets.

A set  $T$  of  $t$  points in  $J_p^2$  is said to be *thick* (or *even*) in  $G(p)$  or  $E(p)$  for slope  $\sigma$  if it is not empty and every line of slope  $\sigma$  through a point in  $T$  contains at least two such points (or a positive even number of points). Note from Lemma 1 that if  $T$  is thick or even in  $G(p)$  for  $\sigma$  then it is thick or even in  $G(p)$  for any  $\sigma' \equiv_p \sigma$ .

In Figure 1, the illustrated set  $T$  of seven circled dots lies on a single line in  $G(7)$  of slope 5.  $T$  is thick (and not even) in  $G(7)$  for slope  $\sigma = 5$ , and for the six other slopes  $p$ -equivalent to 5, but no others. In  $E(7)$   $T$  is thick (and not even) only for  $\sigma = 1/3$ : two lines in that family contain two of the seven points in  $T$  each and one contains the other three. The families with the other six slopes in  $R_7^+$  each contain at least one line which includes only one point in  $T$ , so  $T$  is neither even nor thick in  $E(7)$  for any of the other 7-equivalent slopes.

Define  $N_{G(p)}(S)$  (and  $N_{E(p)}(S)$ ) as the number of points in the smallest subset of  $J_p^2$  that is thick in  $G(p)$  (or in  $E(p)$ ) for  $S$ . Such a  $T$  is said to be *minimal* in  $G(p)$  (or  $E(p)$ ) for  $S$ .

**Lemma 3** Let  $C$  be a linear code whose words satisfy parity checks over subsets equal to all lines with slopes in  $S$ . Let  $d_{\min}$  be the minimum distance between codewords in  $C$ . Then  $d_{\min} = N_{G(p)}(S)$ .

**Comment.** Lemma 3 is easily proved using the fact that the minimum distance of a code is the size of the smallest set of erasures which cannot be corrected. ■

The codes we have discussed and will discuss further are constructed in  $G(p)$ .  $E(p)$  is introduced to prove results about the thickness in  $G(p)$  of sets of points for sets of slopes. Lemma 4 summarizes some relations between thickness in  $E(p)$ , in  $E(p')$  for a larger prime  $p'$ , and in  $G(p)$ .

**Lemma 4** Let  $S \subseteq R_p^+$  be a set of slopes, and let  $p'$  be prime and larger than  $p$ . Then  $N_{E(p)}(S) \geq N_{G(p)}(S)$  and  $N_{E(p)}(S) \geq N_{E(p')}(S)$ .

**Comment.** The definitions of  $R_p^+$ ,  $\lambda_{G(p)}$  and  $\lambda_{E(p)}$  in Lemma 1(i) imply the inclusions

$$(1) \quad R_{p'}^+ \supset R_p^+, \quad \lambda_{E(p)}(\sigma, \pi) \subseteq \lambda_{E(p')}(\sigma, \pi), \quad \lambda_{E(p)}(\sigma, \pi) \subseteq \lambda_{G(p)}(\sigma, \pi),$$

which imply that a  $T$  thick for  $S$  in  $E(p)$  is thick for  $S$  in  $G(p)$  and in  $E(p')$ . ■

## IV Sizes of Planar Sets Thick in $G(p)$ and $E(p)$

Since by Lemma 2 all codes with the same number  $s$  of  $p$ -distinct slopes in  $S$  have the same rate, a code designer is interested in finding which set of  $s$  slopes maximizes  $N_{G(p)}(S)$ , and therefore  $d_{\min}$ , and how big that maximum is. Let  $n_p^{\min}(s), n_p^{\max}(s)$  denote the extrema of  $N_{G(p)}(S)$  over all  $S$  which contain  $s$   $p$ -distinct slopes. The minimum and maximum agree for  $s \leq 3$ , but seem not to do so at  $s = 4$ .

Lemma 5 gives weak but general upper and lower bounds to the sizes of point sets thick in  $G(p)$  for  $s$   $p$ -distinct slopes.

**Lemma 5** Let  $S$  contain  $s$   $p$ -distinct slopes. Then  $2p \geq n_p^{\max}(s) \geq n_p^{\min}(s) \geq s + 1$ .

**Proof.** If  $T$  of size  $t$  is thick for  $S$  then each of the  $s$  lines with slopes in  $S$  through a particular point in  $T$  also goes through at least one of the other  $t - 1$  points. Then  $s \leq t - 1$ , since by Lemma 1(iii) no two of these lines are parallel.

Let  $T'$  be all  $2p$  points on two lines in a  $\sigma$ -family. That family has  $p > 2$  of those points on each of two lines and no points on its other lines. Every other line in  $G(p)$  includes two points of  $T'$  by Lemma 1(iii), so  $T'$  is thick for all slopes in  $R_p^+$ . ■

Lemma 6 lower bounds the sizes of point sets thick in  $E(p)$  for a set of  $s$  slopes.

**Lemma 6** Let  $T \subseteq J_p^2$  be a set of  $t$  points thick in  $E(p)$  for  $S \subseteq R_p^+$ , a set of  $s$  slopes, and let  $\bar{T}$  be  $T$ 's convex hull. Then  $t \geq 2s$ . If  $t = 2s$  then  $\bar{T} = T$  and each vertex in  $\bar{T}$  is connected to each of the  $s$  vertices an odd number of sides away by a line with a different one of the  $s$  slopes in  $S$ . Lines connecting vertices an even number of sides apart have slopes not in  $S$ .

**Comment.** A set  $T$  of  $t = 2$  points is thick in  $E(p)$  for the slope of the line they define by Lemma 1(ii), and is its own convex hull. For more slopes, each of the two lines with slope in  $S$  tangent to  $\bar{T}$  must have two points on  $\bar{T}$ , since if it had only one the second point needed by a  $T$  thick for that slope would lie outside  $\bar{T}$ , which it cannot do by convexity. ■.

Lemma 6 gives useful hints for the construction of  $T$  thick in  $E(p)$  for  $S \subseteq R_p^+$  with  $t = 2s$ . Figure 2 show such sets for  $s = 1, 2, 3, 4$  and 6. By Lemma 4 those  $T$  are also thick in  $G(p)$  and in  $E(p')$  for the same set of slopes. It turns out that the sets in Figure 2 are also thick in  $G(p')$  for the same set of slopes for all sufficiently large  $p'$ .

Lemma 7 specifies for each  $p$  a set  $S_p$  of rational slopes which (i) are  $p$ -distinct and (ii) have the *inclusion* property that if  $p' > p$  then  $S_{p'} \supseteq S_p$ . The set of slopes used in Figure 2 has both properties, as does the set  $J_p^+$ , which is also of the maximal size  $p + 1$ . But choosing slopes from  $J_p^+$  makes sketching convex hulls, so useful to the intuition (and the reason for bringing in Euclidean structures) difficult: the figures are very tall and thin, and lean to the right. Sets of distinct representatives from the  $p + 1$  classes of the  $p$ -equivalence relation used in the examples in Figure 2 are more suitable for drawing figures on small pieces of paper, and share the inclusion property. They are also maximal for small primes, and nearly maximal for much larger ones.

The set of eight slopes in Figure 2 is derived by defining a norm on the rationals mod  $p$ , which assigns the same value to both members of each of the pairs  $\{0/1, 1/0\}$  and  $\{1/1, -1/1\}$ , and to the four members of each tetrad  $\{i/j, -i/j, j/i, -j/i\}$ , and orders slopes by their norms. All slopes with norms less than  $p$  are  $p$ -distinct, and the set is invariant under reflection about the  $x$  and  $y$  axes and rotation by 90 degrees, so it allows pictures of convex hulls which have many slopes but are not tall and thin or leaning to the right.

**Lemma 7** Define a (positive or negative, proper or improper, extended) rational  $i/j$  to be reduced if  $i = 0, j = 1$  or  $i = 1, j = 0$  or  $j > 0$  and  $\gcd(|i|, j) = 1$ . Define the norm  $\nu$  on reduced  $i/j$  by  $\nu(i/j) = i^2 + j^2$ . Then all reduced rationals with  $\nu(i, j) < p$  are  $p$ -distinct.

**Comment.** If  $i/j + m/n \equiv_p 0$  then either  $in + mj = 0$  and the two are  $p$ -distinct negatives, or  $in + mj = kp$ ,  $k > 0$ . Then

$$(2) \quad p^2 > \nu(i/j)\nu(m/n) \geq (i^2 + j^2)(n^2 + m^2) - (mi - nj)^2 = (in + mj)^2 \geq p^2,$$

a contradiction. ■

Lemma 7 does not tell how close the number of  $p$ -distinct extended rationals with norms less than  $p$  is to the maximal number  $p + 1$ . In fact, for  $3 \leq p \leq 43$ , taking the first  $p + 1$  rationals in order of increasing norm gives a maximal  $p$ -distinct set. For larger  $p$ , among the first 2000 primes about 95 percent of the  $p + 1$   $p$ -distinguishable rational slopes have norms less than  $p$ , and the fraction does not decrease with  $p$ .

Lemma 8 was used in constructing the sets  $T_1, T_2, \dots, T_8$  in Figure 2. Define the *translate* of a set  $T$  of points in  $G(p)$  by the vector  $\pi = (i, j)$  as the set  $T + \pi = \{\pi' + \pi : \pi' \in T\} = \{(i' + i, j' = j) : (i', j') \in T\}$ . Translation preserves families, permuting their lines, so it preserve evenness and thickness in  $E(p)$ .

**Lemma 8** Let  $T \in J_p^2$  be thick in  $E(p)$  for a set  $S$  of  $s$  slopes with norms less than  $p$ . Let  $\pi$  be a vector of slope  $\sigma$  not in  $S$ , let  $S' = S \cup \sigma$ ,  $T' = T + \pi$ ,  $T'' = T \cup T'$ ,  $T''' = T \oplus T' =$

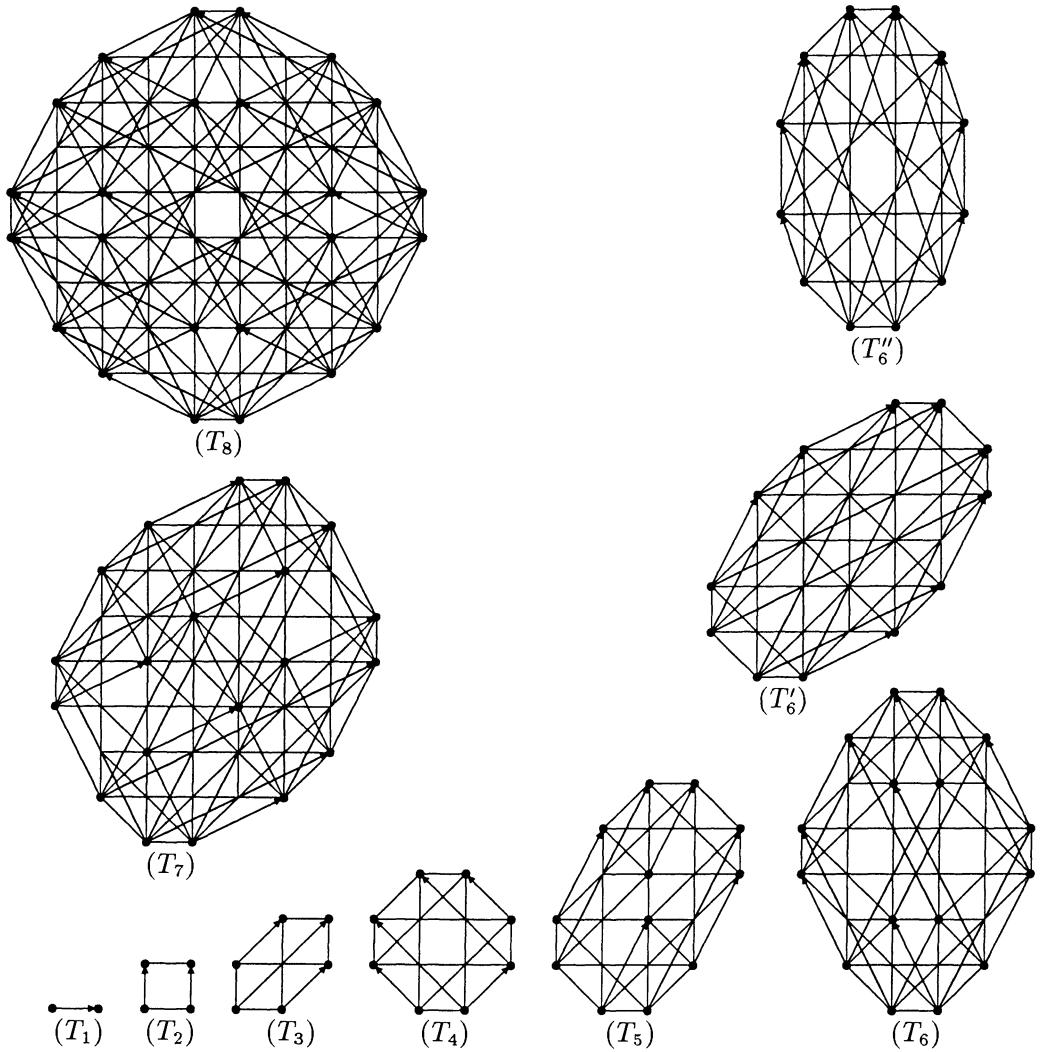


Figure 2: A sequence of 8 sets thick in  $E(p)$  for the first one ( $T_1$ ) to eight ( $T_8$ ) slopes in the set  $\{0, \infty, 1, -1, 2, -2, 1/2, -1/2\}$ . Each set is constructed using Lemma 8, adding the last set mod 2 to its shift in the new direction by an amount which cancels the maximum number of points in the sum. Arrows demonstrate thickness for the most recently added slope. Two other sets are also shown:  $(T'_6)$ , of size  $t = 2s = 12$ , smaller than  $(T_6)$ , and thick for the asymmetric set  $\{0, \infty, 1, -1, 2, 1/2\}$ , and  $(T''_6)$ , also of size 12, thick for the symmetric set  $\{0, \infty, 1, -1, 3, -3, 1/3, -1/3\}$ , shown at half the scale of the others.

$T \cup T' - T \cap T'$  and let  $p' \geq p$  be large enough so that both  $T'' \subseteq J_{p'}^2$  and  $\nu(\sigma) < p'$  for all  $\sigma \in S'$ . Then  $T''$  is thick in  $G(p')$  for  $S'$ , and if  $T$  is even in  $E(p)$  then  $T'''$  is also even in  $G(p')$  for  $S'$ .

**Comment.** Let  $\pi'_j = \pi_j + \pi$ , the translate of each  $\pi_j$  in  $T$ . Since the line  $(\pi_j, \pi'_j)$  has slope  $\sigma$ , if  $T$  is thick for  $S \subseteq R_p^+$  and  $\sigma$  is in  $R_p^+ - S$  then  $T'' = T \cup (T + \pi)$  is thick for  $S' = S \cup \{\sigma\}$ . The first result in Lemma 8 follows from Lemmas 4, 7. It remains to show that deleting points in the intersection  $T \cap T'$  leaves an even set when  $T$  is even.

If  $\pi_j$  is in that intersection then for each  $\sigma \in S'$ , let  $U(\sigma, \pi_j)$  contain the points in  $T$  on a line of slope  $\sigma$  through  $\pi_j$ , and let  $U'(\sigma, \pi_j)$  contain the points in  $T'$  on the same line.

For  $\sigma \in S$ , by the evenness of  $T$  and  $T'$ ,  $|U|$  and  $|U'|$  are even, so  $U \oplus U'$  has an even number  $|U| + |U'| - 2|U \cap U'|$  of members. For the new slope  $\sigma = S' - S$ ,  $|U|$  and  $|U'|$  need not be even but are equal, so  $|U \oplus U'| = 2|U| - 2|U \cap U'|$  is even. And  $T'''$  is empty iff  $T = T'$ , in which case  $T$  is already thick for  $\sigma'$ , a contradiction, so  $T''$  is even for  $\sigma'$  and thus for  $S'$ . ■

A final comment. Constructions of thick sets, like those in Figure 2 for  $s \leq 8$ , give firm upper bounds to  $n_p^{max}$  and  $n_p^{min}$ . However I have found no definitive proof that  $n_p^{max} > 2s$  for any large  $p$ , though it is suggested by (nonexhaustive) simulations for  $s = 4$ . Firm results for  $s \leq 4$  include these:

- $n_p^{min}(s) = n_p^{max}(s) = 2s$ ,  $p \geq 3$ ,  $1 \leq s \leq 3$ .
- $n_3^{min}(4) = n_3^{max}(4) = 6$ .
- $n_p^{min}(4) \leq 8$ ,  $p \geq 5$ ;  $n_5^{max}(4) \leq 8$ ,  $n_7^{max}(4) \leq 9$ ,  $n_{11}^{max}(4) \leq 10$ .

It seems likely that all inequalities bounding maxima and minima above are equalities.

## References

- [1] M.J.E. Golay, “Notes on Digital Coding”, Proc. IRE June 1949, v.37 p. 657.
- [2] P. Elias, “Error-Free Coding”, PGIT Transactions, v. 4, Sept. 1954 pp. 29-37.
- [3] J. L. Massey, “Threshold Decoding”, MIT Press, Cambridge Mass., 1963.
- [4] P. Elias, “The Noisy Channel Coding Theorem for Erasure Channels”, American Mathematical Monthly v.81 no.8, Oct. 1974, pp.853-862.
- [5] Hardy and Wright, Chapter III p. 23, in An Introduction to the Theory of Numbers, Third Edition, Clarendon Press, Oxford, 1954.
- [6] R. Lidl and H. Niederreiter, pp. 496-508 in Finite Fields, vol. 20 in *Encyclopedia of Mathematics and its Applications*, Addison Wesley, 1983.

# Spherical Codes From the Hexagonal Lattice

Thomas Ericson  
Linköping University  
Dept. of Electrical Engineering  
S-581 83 Linköping, Sweden

Victor Zinoviev  
\*University of Puerto Rico  
Dept. of Mathematics  
Rio Piedras, PR 00931, USA

## Abstract

Using subsets of the hexagonal lattice as alphabets we present two classes of constructions of spherical codes. Some of the codes obtained are optimal or best known.

## I Introduction

A spherical code  $X$  is a subset of the unit sphere  $\Omega_n$  in  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ . We characterize any such code with a triple  $(n, \rho, M)$  of parameters where  $n$  denotes dimension,  $\rho$  is squared minimum distance

$$\rho = \rho(X) = \min_{x \neq y} \|x - y\|^2; \quad x, y \in X$$

and  $M$  is cardinality,  $M = |X|$ . In the present investigation we will build spherical codes by concatenation techniques, using various subsets of the hexagonal lattice as alphabets. For convenience we will represent the alphabets as points in the complex plane  $\mathbb{C}$ , which we identify with the plane  $\mathbb{R}^2$  using the obvious map  $\mathbb{C} \rightarrow \mathbb{R}^2$ ;  $z = x + jy \rightarrow (x, y)$ .

## II First Construction

Denote by  $F_3^N$  the set of all  $N$ -tuples of elements from the ternary field  $F_3$ ; denote by  $F_3^{N,w}$  the subset consisting of all  $N$ -tuples with Hamming weight  $w$ . In our first construction a spherical code  $X$  will be obtained as a map of a pair of ternary codes: one constant weight code  $C \subseteq F_3^{N,w}$  and one unrestricted code  $A \subseteq F_3^w$ . Notice that the length  $w$  of the code  $A$  equals the weight of the code  $C$ .

Let  $\Theta = e^{2\pi j/3}$  be a complex cubic root of unity and for each codeword  $c = (c_1, c_2, \dots, c_N) \in C$  let the function  $\phi_c : \{1, 2, \dots, N\} \rightarrow \{1, 2, \dots, w\}$  be defined as

$$\phi_c(i) = \sum_{j=1}^i w_H(c_j).$$

---

\*On leave from the Institute for Problems on Information Transmission, Ermolova Str. 19, GSP-4, Moscow, 101447, Russia.

In words:  $\phi_c(i)$  denotes the weight of the first  $i$  components in  $c = (c_1, c_2, \dots, c_N) \in C$ . For a given pair  $(c, a)$  of codewords  $c = (c_1, c_2, \dots, c_N) \in C$  and  $a = (a_1, a_2, \dots, a_w) \in A$  we define  $x = (x_1, x_2, \dots, x_N)$  as follows:

$$x_i = \frac{1}{\sqrt{3w}}(1 - \Theta^{c_i})\Theta^{\alpha_{\phi_c(i)}}; \quad i = 1, \dots, N;$$

where the exponents of  $\Theta$  are defined in the natural way:

$$\Theta^0 = 1; \quad \Theta^1 = \Theta = e^{2\pi j/3}; \quad \Theta^2 = e^{4\pi j/3}.$$

Notice that the code  $A$  is employed only in those coordinates where the codeword  $c$  from the code  $C$  is nonzero.

The set of points so generated form a spherical code  $X$  with parameters as follows:

$$n = 2N$$

$$\rho \geq \frac{1}{w} \min \{d_C, 3d_A\}$$

$$M = M_A M_C$$

where  $d_A$  and  $d_C$  denote the minimum Hamming distance in  $A$  and  $C$  respectively, and where  $M_A$  and  $M_C$  are the cardinalities of  $A$  and  $C$ . In Figure 1 the enumeration of points in the complex plane  $C$  induced by our construction is indicated. We illustrate the construction with the aid of several examples.

**Example 1:**  $C = \{012, 201, 120\}$ ;  $A = F_3^2$ . Clearly we have  $d_C = 3$ ,  $M_C = 3$ ;  $d_A = 1$ ,  $M_A = 9$ , which gives us  $(n, \rho, M) = (6, 3/2, 27)$ . This code is known as the Hessian polytope. It meets the Levenshtein bound [1] and is therefore an optimal configuration.

**Example 2:**  $C = \{111, 222\}$ ;  $A = F_3^3$ . Here we have  $d_C = 3$ ,  $M_C = 2$ ;  $d_A = 1$ ,  $M_A = 27$ . We get  $(n, \rho, M) = (6, 1, 54)$ .

**Example 3:**  $C = \{100, 200, 010, 020, 001, 002\}$ ;  $A = \{0, 1, 2\}$ . Here we have  $d_C = 1$ ,  $M_C = 6$ ;  $d_A = 1$ ,  $M_A = 3$  and we obtain  $(n, \rho, M) = (6, 1, 18)$ .

Examples 2 and 3 do not produce codes which are individually strong. However, by a proper adjustment of coordinates the union produces a code with parameters  $(n, \rho, M) = (6, 1, 72)$  which is the best known configuration for contact number in six dimensions (Conway-Sloane [2], p.23).

**Example 4:**  $C = \{0111, 1012, 1201, 1120, 0222, 2021, 2102, 2210\}$ ;  $A = F_3^3$ .

$$d_C = 3, \quad M_C = 8; \quad d_A = 1, \quad M_A = 27 \quad \Rightarrow (n, \rho, M) = (8, 1, 216).$$

**Example 5:**  $C = \{1000, 0100, 0010, 0001, 2000, 0200, 0020, 0002\}$ ;  $A = F_3$ .

$$d_C = 1, \quad M_C = 8; \quad d_A = 1, \quad M_A = 3 \quad \Rightarrow (n, \rho, M) = (8, 1, 24).$$

Again, none of the two last examples is very strong individually, but an appropriate union gives us  $(n, \rho, M) = (8, 1, 240)$  which again is optimal (Conway-Sloane [2], p.23).

### III Second Construction

The simplest construction of a spherical code from a ternary code is obtained by a direct mapping of the ternary field  $F_3$  onto the corners of an equilateral triangle centered around the origin in the complex plane.

Let  $A \subseteq F_3^N$  and let  $a = (a_1, a_2, \dots, a_N) \in A$  be a codeword. The corresponding complex-valued vector  $x = (x_1, x_2, \dots, x_N)$  is given by

$$x_i = \frac{1}{\sqrt{N}} \Theta^{a_i}; \quad i = 1, 2, \dots, N.$$

For good ternary codes this simple map often yields reasonably good spherical codes, but often improvements can be obtained by simple modifications. We will describe one such possibility.

Let  $\alpha = e^{\pi j/3}$  and let  $B \subset F_2^{N,w}$  be a binary constant weight code with parameters  $(n_B = N, w, d_B, M_B)_2$ . Further, let  $C \subset F_3^w$  be an unrestricted ternary code with parameters  $(n_c = w, d_C, M_C)_3$ . The parameters of the first ternary code  $A$  are  $(n_A = N, d_A, M_A)_3$  and the corresponding spherical code is denoted  $X$ . Let  $\phi_b$  be defined as

$$\phi_b(i) = \sum_{j=1}^i w_H(b_j)$$

and define a map  $B \times C \rightarrow \mathbb{C}^N$  by the rule

$$y_i = \frac{1}{\sqrt{w}} b_i \alpha^{1+2c_{\phi_b(i)}}; \quad i = 1, \dots, N$$

It is easy to see that the vectors  $y = (y_1, y_2, \dots, y_N) \in \mathbb{C}^N$  formed in this way constitute a spherical code  $Y$  with parameters

$$n = 2N$$

$$\rho \geq \min \left\{ \frac{d_B}{w}, 3 \frac{d_C}{w} \right\}.$$

However, the construction we like to emphasize is one obtained by a properly selected union. Choosing  $N = 4w$  we have

$$x_i = \frac{1}{2\sqrt{w}} \alpha^{2a_i}$$

$$y_i = \frac{b_i}{\sqrt{w}} \alpha^{1+2c_{\phi_b(i)}}$$

$$i = 1, 2, \dots, N = 4w.$$

The various points constituting the alphabets in the two mappings are shown in Figure 2. An easy calculation reveals that any pair  $x \in X; y \in Y$  are at squared distance at least  $3/2$ . Thus, the union  $Z = X \cup Y$  is a spherical code with parameters

$$n = 8w$$

$$\rho \geq \min\left\{\frac{3d_A}{4w}, \frac{d_B}{w}, 3\frac{d_C}{w}, 3/2\right\}$$

$$M = M_A + M_B M_C.$$

We give two examples.

**Example 6:**

$$\begin{aligned} (n_A, d_A, M_A)_3 &= (4, 2, 27)_3 \\ (n_B, w, d_B, M_B)_2 &= (4, 1, 2, 4)_2 \\ (n_C, d_C, M_C)_3 &= (1, 1, 3)_3 \end{aligned}$$

(the existence of codes with these parameters is obvious). We obtain  $(n, \rho, M) = (8, 3/2, 39)$ .

**Example 7:**

$$\begin{aligned} (n_A, d_A, M_A)_3 &= (12, 6, 3^6)_3 \\ (n_B, w, d_B, M_B)_2 &= (12, 3, 6, 4)_2 \\ (n_C, d_C, M_C)_3 &= (3, 2, 3^2)_3. \end{aligned}$$

The first code is the extended ternary Golay code; the two other codes are trivial. We obtain  $(n, \rho, M) = (24, 3/2, 765)$ .

## References

- [1] V.I. Levenshtein, “Bounds for packings of metric spaces and some of their applications”, Probl. Cybern., Vol.40, Nauka, Moscow 1983, pp. 43- 110 (in Russian).
- [2] J.H. Conway and N.J.A Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, Sec.Ed. 1993.

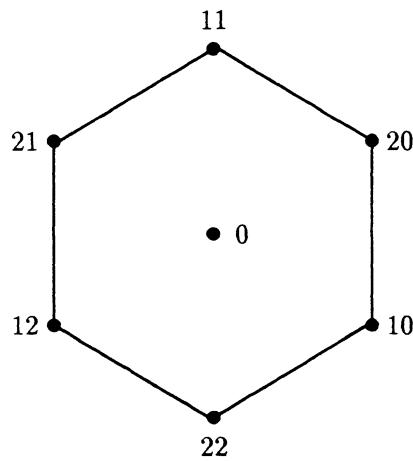


Figure 1: Enumeration of alphabet in first construction

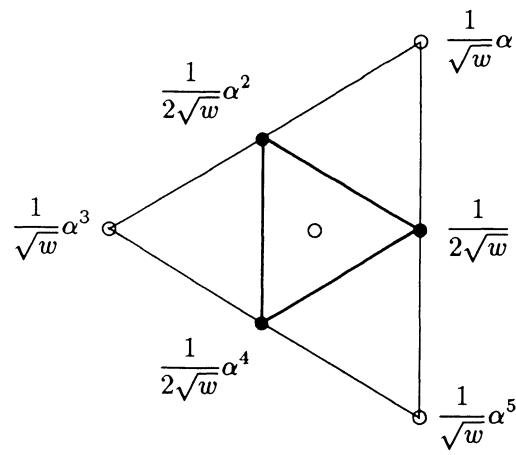


Figure 2: Alphabets used in second construction

# Trellises Old and New

G. David Forney, Jr.  
Motorola Codex  
Mansfield, MA 02048 USA

## Abstract

The concept of a trellis and the proof of the optimality of the Viterbi algorithm grew out of early work at Codex Corporation, here published for the first time. A recently observed flaw in this proof is noted. Trellises for block codes and lattices are of current interest. An absolutely minimal trellis is exhibited for the  $E_8$  lattice. This trellis gives a succinct summary of the algebraic, geometrical, and dynamical structure of  $E_8$  and its sublattices.

## Preface

It is said that for a wedding one should bring “something old, something new, something borrowed, something blue.” Birthdays, too?

## I Something Old

Codex Corporation was founded in 1962 to exploit the error-correcting coding patents of Jim Massey (threshold decoding) [1] and Bob Gallager (low-density parity-check codes) [2]. Threshold decoders were the main product of the company in the 1960s.

When I came to Codex in 1965, the company was just putting into production its first standard product, the TD-12 error corrector, a Massey-type burst-error-correcting (!) threshold decoding system with constraint length 12. The principal components of the system were 36 flip-flops, 12 in the encoder and 24 in the decoder. At that time a flip-flop was implemented as a quite massive 1” x 1” x 2” epoxy-encased module.

My first practical coding achievement was to alter the threshold decoding equations so that two flip-flops could be saved in the decoder. After excited phone calls between Arthur Kohlenberg and Jim Massey to confirm this breakthrough, the call went out to halt production and commence redesign to realize these impressive savings! I felt that my career at Codex was off to a good start.

Most of my coding work in the 1960s was for space communications, initially for the NASA Pioneer project. After first investigating threshold decoders, we made the leap all the way to sequential decoding and  $R_0$ . (Coding had already been declared dead at M.I.T., because we knew how to achieve  $R_0$ , or  $R_{comp}$  as it was then called.) We were able to program a commercial Honeywell minicomputer with a clock speed of about 1 MHz to decode a rate-1/2 convolutional code at speeds from 8 b/s to 512 b/s.

All of our experience at Codex suggested strongly that convolutional codes were essentially superior to block codes in terms of performance versus complexity. I wanted to understand why, at a fundamental level, and enjoyed the great good fortune of being able to discuss these topics regularly with Bob Gallager, Arthur Kohlenberg, and Jim Massey.

In an appendix to our final report to the NASA Ames Research Center [3], I wrote “a rather complete review of what is known about the performance of random tree codes ... with optimum decoders.” This appendix is frequently cited for its introduction of the trellis concept and for the first proof that the Viterbi algorithm [4] is optimum. As it has never been published, this seems like an appropriate occasion to reproduce two pertinent sections. One may note that “error events” and “string decoders” (the “Minty algorithm” [5]) appear here also for the first time in the coding literature.

### The Merging Concept in the Analysis of Tree Codes [3, pp. A17-A22]

The central concept to be grasped in the analysis of tree codes is that of *merging*. Two source sequences  $s$  and  $s'$  are said to be *merged* at branch  $i$  if the two source sequences have the same  $\nu + 1$  letters in the positions  $i - \nu$  through  $i$ :

$$s'_{i'} = s_{i'}, \quad i - \nu \leq i' \leq i. \quad (36)$$

By the definition of constraint length [number of memory elements  $\nu$ ], if  $s$  is merged with  $s'$  at branch  $i$ , the input letters in the branches  $x_i = x'_i$  of the corresponding code words must be identical. For example, in a terminated tree code, all source sequences with the same first letter are merged at the first branch, and all with the same last letter are merged at the  $(K + \nu)$ th (last) branch.

A corollary concept is that of the *unmerged span*, which is defined with respect to two particular source sequences, say  $s$  and  $s'$ . The first unmerged span  $U_1$  contains the indexes of the source letters in the span from the first letter in which  $s$  differs from  $s'$  to the last letter of the first subsequent string of  $\nu$  consecutive letters in which  $s$  agrees completely with  $s'$ . Thus over the first unmerged span  $s$  and  $s'$  are unmerged, but they are about to merge at the end of the span. The  $j$ th unmerged span  $U_j$  then contains the indexes of the source letters in the span from the first letter not in  $U_{j-1}$  in which  $s$  differs from  $s'$  to the last letter of the first subsequent string of  $\nu$  consecutive letters in which  $s$  agrees completely with  $s'$ . It is obvious that:

- The unmerged spans are disjoint;
- The unmerged spans contain the indexes of all branches for which  $s$  and  $s'$  are unmerged.

Thus with the concept of unmerged span we partition the set of branches at which  $s$  and  $s'$  are unmerged into disjoint subsets. Each consists of a consecutive string of at least  $\nu + 1$  branches; the spans may or may not be separated by merged segments.

The principal utility of this concept lies in the observation that the log likelihood ratio of the code words  $\mathbf{x}$  and  $\mathbf{x}'$  corresponding to the two source sequences is equal to the sum of the log likelihoods over the unmerged spans. For let  $\mathbf{y}$  be the received word and  $y_i$ , the part of the received word corresponding to the  $i$ th branch  $x_i$  of the transmitted word; then

$$\begin{aligned}\ln Pr(\mathbf{y} \mid \mathbf{x})/Pr(\mathbf{y} \mid \mathbf{x}') &= \sum_i \ln Pr(y_i \mid x_i)/Pr(y_i \mid x'_i) \\ &= \sum_j \sum_{i \in U_j} \ln Pr(y_i \mid x_i)/Pr(y_i \mid x'_i),\end{aligned}\quad (37)$$

since the branch log likelihood ratio is zero for all  $i$  for which  $\mathbf{s}$  and  $\mathbf{s}'$  are merged and therefore  $x_i = x'_i$ .

Now let us suppose that the code word actually sent is  $\mathbf{x}$ , but that a maximum likelihood decoder chooses some code word  $\mathbf{x}' \neq \mathbf{x}$ . From (37) we have immediately

$$\sum_j \sum_{i \in U_j} \ln Pr(y_i \mid x_i)/Pr(y_i \mid x'_i) \leq 0. \quad (38)$$

But the fact that  $\mathbf{x}'$  has likelihood greater than all other sequences  $\mathbf{x}''$  implies the stronger statement

$$\sum_{i \in U_j} \ln Pr(y_i \mid x_i)/Pr(y_i \mid x'_i) \leq 0, \quad \text{all } j. \quad (39)$$

For suppose the inequality of (39) is not satisfied for some  $U_j$ ; that is, over the  $j$ th unmerged span the correct code word  $\mathbf{x}$  has greater likelihood than  $\mathbf{x}'$ . Consider then the source sequence  $\mathbf{s}''$  which is equal to  $\mathbf{s}$  over the span  $U_j$  and to  $\mathbf{s}'$  elsewhere; the corresponding code word  $\mathbf{x}''$  will equal  $\mathbf{x}$  over  $U_j$  and  $\mathbf{x}'$  elsewhere, and will therefore have greater likelihood than  $\mathbf{x}'$  if (39) is not satisfied.

Consequently, when errors occur with tree codes, it is natural and convenient to think of these as separate error events, each one corresponding to a particular unmerged span  $U_j$ . Error events may be of any length and contain any number of symbol errors, but do have a definite beginning and end. Intuitively, the number of error events will be proportional to the code length, and one should think not of the probability of error per block, but the probability of error per branch, defined as the probability of an error event starting (or ending) at any particular branch.

When we consider terminated tree codes as block codes, however, we shall be interested in block probability of error. Then the following lemma, which follows directly from the above discussion, will be useful:

**Lemma 1** There will be a decoding error with a terminated tree code if and only if some incorrect word which differs from the correct code word only over a single unmerged span has greater likelihood than the correct word.

For suppose the word actually decoded has  $J$  unmerged spans with respect to the correct word; by (39) the likelihood ratio over each of these spans will be less than one, so that each of the  $J$  words differing from the correct word over a single one of these spans will have greater likelihood than the correct word.

The merging concept suggests a change in the way we picture tree codes of finite constraint length. Customarily tree codes are depicted graphically as trees, with  $q^i$  possible transmitted branches at branch  $i$ . Taking into account that sequences do merge with each other, however, we arrive at a structure more like a trellis, with no more than  $q^{v+1}$  possible transmitted branches at any one branch. The difference is illustrated in Figure 5, for a binary tree code of constraint length 2 terminated after five information bits. In both the tree and the trellis pictures, the representation of any of the 32 particular source sequences of 5 information bits followed by 2 zeroes is the path through the graph labelled by the corresponding bits; in both pictures darkened lines indicate the representation of 0100000. The code would be totally specified by labelling each branch with the appropriate channel inputs. The trellis picture recognizes that since 0100000 merges with 0000000 after the fourth branch, the corresponding transmitted branches must be the same for the two code words.

Figure 5. Terminated binary ( $q = 2$ ) tree code with  $v = 2, K = 5$ .

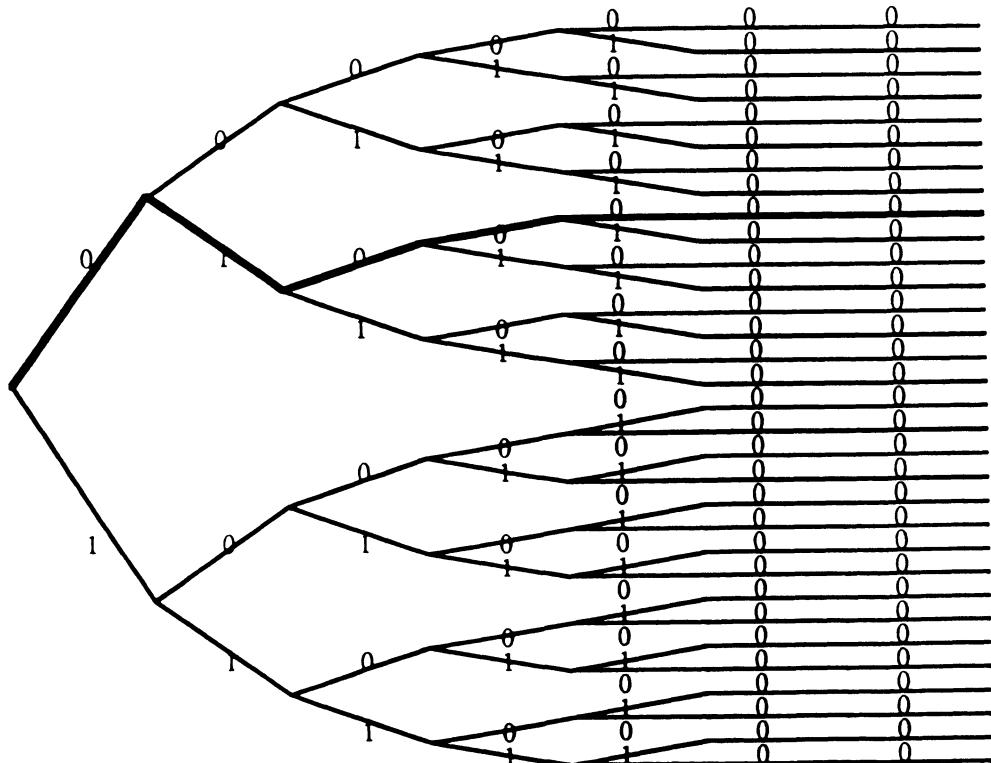


Figure 5a. Tree picture.

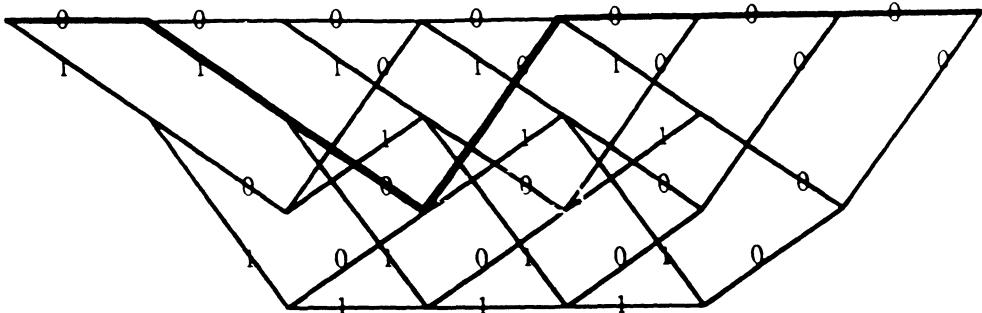
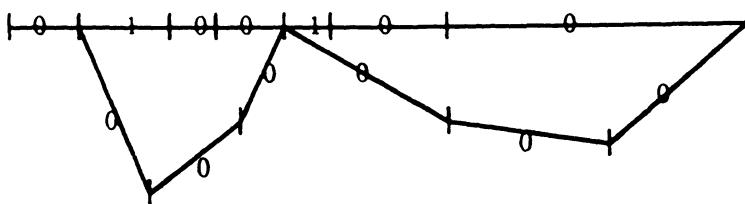


Figure 5b. Trellis picture (taking account of merging).

These pictures suggest making up a maximum likelihood decoder out of string, as follows. For each branch, cut out a piece of string of length  $-\ln Pr(y_i | x_i)$ , where  $y_i$  is what was received for branch  $i$ , and  $x_i$  is what would have been transmitted if the actual code word included that branch. For a tree decoder, attach all of these pieces of string together as in the tree picture of Figure 5a. Hold the resulting bundle at the tree origin, let all the strings hang down, and pick out the terminal node which is highest; this will be the node corresponding to the shortest path through the tree, thus to the minimum value of

$$-\sum \ln Pr(y_i | x_i) = -\ln Pr(\mathbf{y} | \mathbf{x}),$$

and thus to the most likely code sequence. Similarly, for a trellis decoder, attach the pieces of string together according to the trellis topology, as in Figure 5b. Pick up the resulting bundle at the two end points and pull them apart until some path becomes taut; this is again the path of minimum length and hence of maximum likelihood. All less likely paths will dangle down from the most likely path in loops corresponding to unmerged spans. Suppose, for example, that the code word corresponding to 0000000 is actually sent, but that 0100100 is the sequence of greatest likelihood. These two paths alone might look in the string decoder like



The discussion above simply pointed out that the overall block decoding error ought to be thought of as two distinct error events, one corresponding to each loop, and that the incorrect sequence has to be more likely than the correct one over each event individually. It is obvious from the picture that both 0100100 and 0000100 are also more likely than 0000000.

The Viterbi Algorithm [3, pp. A30-A34]

We now show that the structure of tree codes permits a decoding algorithm which gives precisely the same results as maximum likelihood decoding and is therefore optimum, but which has complexity proportional only to  $q^\nu$ , rather than to the  $q^K$  required by maximum likelihood decoding of a terminated tree code. That something like this was possible was foretold by our string decoders, where the number of pieces of string required was proportional to  $q^K$  for the tree structure, but only to  $q^\nu$  for the trellis.

The critical observation to be made is the following. Let  $\mathbf{x}$  and  $\mathbf{x}'$  be two input words which agree for the span of  $\nu$  consecutive letters ending in  $i$ ; that is, which are either merged at branch  $i$  or which have an unmerged span terminating at  $i$ . Let  $\mathbf{x}$  be the one which has greater likelihood up to branch  $i$ :

$$\sum_{1 \leq i' \leq i} \ln Pr(y_{i'} | x_{i'}) / Pr(y_{i'} | x'_{i'}) \geq 0 \quad (71)$$

Then  $\mathbf{s}'$  cannot be the choice of a maximum likelihood decoder. For  $\mathbf{x}'$  will always have less likelihood than the input sequence which agrees with  $\mathbf{x}$  up to branch  $i$  and with  $\mathbf{x}''$  thereafter, since the corresponding code word will be identical to  $\mathbf{x}'$  beyond branch  $i$ , and from (71) will therefore continue to have greater likelihood forever.

Let us rephrase this observation in terms of our trellis structure, and our trellis string decoder. Suppose we set up the structure only out to branch  $i$ , and pulled on one of the  $q^\nu$  nodes at that depth and the initial node, thus finding the shortest path to that node. Clearly any other path dangling down in a loop from the shortest path is not going to be a part of the path finally chosen; thus we can take out scissors and cut away any such loops without any possibility of discarding the best possible path. If we do this for all  $q^\nu$  nodes, we shall be left with only one path from the initial node to each of these  $q^\nu$  nodes, which set will form a true tree, with many branches missing, but no loops. This operation can be repeated for each branch in sequence, until one arrives at the final node, at which time only one path will remain, necessarily the shortest one of all and thus the optimum choice of a maximum likelihood decoder. This is the Viterbi algorithm.

Returning to the world where decoders are built with computer components rather than string, let us restate the algorithm. The decoder operates in a series of steps, one for each branch past branch  $\nu$ . At the step associated with branch  $i$ , for each of the  $q^\nu$  possible strings of source letters of length  $\nu$ , it chooses the one input sequence which, among all those which have that particular string of  $\nu$  letters ending at branch  $i$ , has the greatest likelihood up to branch  $i$ . At the end of any step, therefore, it retains a list of only  $q^\nu$  possible sequences up to branch  $i$ ; this determines the length of the decoder memory. It follows that at the next step, there will be only  $q$  sequences to be compared in each class, so only  $q^{\nu+1}$  likelihoods to be computed in all. It is therefore evident that the

complexity of the algorithm is indeed asymptotically proportional to  $q^\nu$ , if the length of the sequences does not grow exponentially with  $\nu$ .

This algorithm may never make a final choice until the tree code is terminated, although a choice can be imposed after a while without any loss in performance, as we shall discuss later. Should the tree code be terminated, however, the algorithm automatically converges on a single choice, since it need not consider words which do not agree at the end with the  $\nu$ -letter synchronizing sequence, and therefore at the step associated with the branch  $K + \nu$  it chooses the one input sequence which, among all those ending with the synchronizing sequence, has the greatest likelihood over the whole block. This sequence must be the same sequence as would be chosen by a maximum likelihood decoder, since the only words ever discarded are those which could not possibly be the choice of a maximum likelihood decoder. The algorithm is therefore optimum — Viterbi too modestly called it suboptimum but asymptotically optimum — and consequently we know the error probability which can be obtained with its use.

Expressions which give the probability of error  $Pr(E)$  and decoding complexity  $G$  when random tree codes are used with the Viterbi algorithm are therefore

$$\begin{aligned} Pr(E) &\cong \begin{cases} q^{-\nu\rho(r)}, & r \geq R_1; \\ q^{-\nu R_1/r}, & r \leq R_1; \end{cases} \\ G &\cong q^\nu, \end{aligned} \quad (72)$$

where we have reproduced (57) and (58). The probability of error is therefore given directly in terms of the decoding complexity by

$$Pr(E) \cong \begin{cases} G^{-\rho(r)}, & r \geq R_1; \\ G^{-R_1/r}, & r \leq R_1; \end{cases} \quad (73)$$

Again, the probability of error decreases only algebraically with decoding complexity, as with maximum likelihood decoding of block codes. However, because the complexity is less with tree codes, the exponent is greater. For a direct comparison of exponents, compare a block code of rate  $R$ , error exponent  $E(R)$ , and therefore [from (22)] complexity exponent  $E(R)/R$ , with the terminated tree code of the right tree code rate  $r_R$  and the right [synchronization rate loss]  $\lambda$  to give an optimum block code of rate  $R$  [that is, if  $R = E'_0(\rho), r_R = R_\rho$ ], which will have complexity exponent  $\rho(r_R)$ ; this comparison is made graphically in Figure 7. Even with the allowance for synchronization loss, the comparison strongly favors the tree code, the more so the closer the rate approaches capacity. Obviously the comparison would be even more favorable to the tree code if we let the block length increase indefinitely so that  $r$  approached the block code rate  $R$ , since the complexity and probability of error depend only on  $\nu$  and  $r$ , and not on the block length; then we would get an exponent equal to  $\rho(R)$ , the  $\rho$  for which  $R = R_\rho$ , as is also illustrated in Figure 7. Again, we are led to feel that although random tree codes can be used as optimum block codes, it is better not to terminate them, or at least to let the blocks between synchronizing

sequences become very long. Though it is true that in this case the equivalent block code exponents go to zero, one is usually more interested in probability of error versus complexity than versus block length.

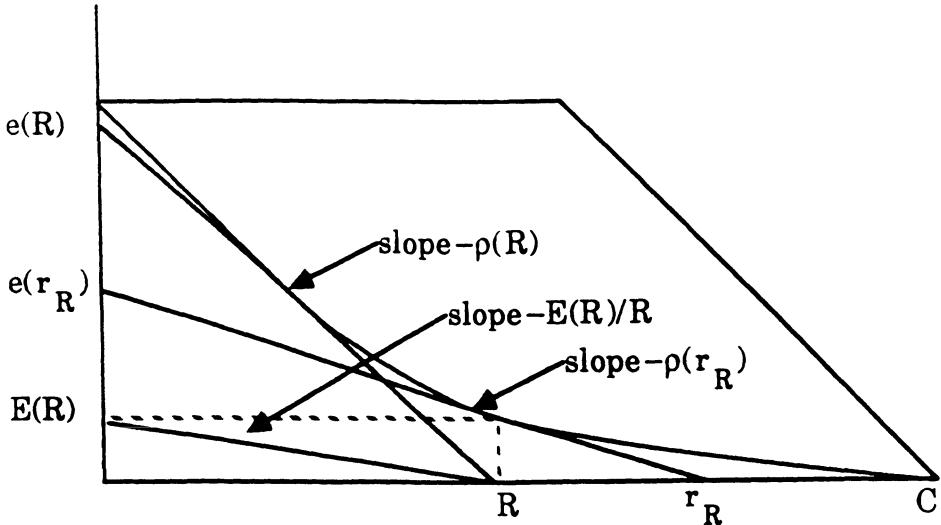


Figure 7. Comparison of block and tree code exponents.

It would be pleasant to report that Codex went on to exploit this discovery by implementing the Viterbi algorithm in its products. Unfortunately, knowing that sequential decoding could achieve  $R_0$ , the “practical capacity,” we did nothing of the kind.

In a talk many years later [6], Andy Viterbi declared that when he invented the algorithm, it was only as a proof technique; Dave Forney realized that it was optimum; but it was Jerry Heller who realized that it was practical. That is likely to be close to the historical truth.

## II Something New

Convolutional encoders have been characterized as linear sequential circuits over finite fields since their invention by Elias [7]. However, Massey and Sain [8]-[9] were the first to establish substantial bridges between convolutional coding theory and linear system theory, and to focus on the code itself rather than on encoders.

Building on this work, I developed a “minimal encoder” for an arbitrary linear convolutional code over a field [10]. This work showed that linear convolutional codes have well-defined minimal state spaces and trellis diagrams.

In 1978, Wolf [11] and Massey [12] proposed describing block codes by trellis diagrams, and gave methods of constructing trellis diagrams for linear block codes over fields. In [13, Appendix], I showed that, given a definite coordinate ordering, linear block codes and lattices have well-defined minimal state spaces and trellis diagrams. Both the Wolf and the Massey trellis diagrams for linear block codes have been shown to be minimal [14].

Massey and his students ran into difficulties in trying to develop comparable results for ring codes [15]. Resolving these difficulties required the development of a general dynamical theory of “group codes” [16], [17], which shows that any code that has a componentwise group property has a well-defined minimal trellis diagram.

A beautiful recent application of this theory is to lattices. An  $n$ -dimensional lattice is a discrete subgroup of  $\mathbf{R}^n$  as an additive group. In any given coordinate system, a lattice is a group code and has a well-defined minimal trellis diagram.

One may seek the coordinate system for a given lattice that yields the minimal trellis diagram with the smallest possible state spaces — the “absolutely minimal trellis” — just as one may seek the coordinate permutation of a given block code that yields the smallest state spaces.

The Gosset lattice  $E_8$  has many remarkable properties [18]. It is the densest possible lattice in eight dimensions; furthermore, the densest lattices in all lower dimensions — namely,  $\mathbf{Z}$ ,  $A_2$ ,  $D_3$ ,  $D_4$ ,  $D_5$ ,  $E_6$  and  $E_7$  — are cross-sections of  $E_8$ .

We now give a construction of  $E_8$  that yields an “absolutely minimal” trellis diagram [19], and that exhibits explicitly the densest lattices in all lower dimensions as cross-sections. Indeed, the proof of minimality is essentially that smaller state spaces would imply the existence of denser lower-dimensional lattices, which is known not to be possible [18]. In this construction,  $E_8$  is the set of all integer linear combinations of the following set of eight generators:

$$\begin{array}{cccccccc} 2/\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/\sqrt{2} & 3/\sqrt{6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2/\sqrt{6} & 4/\sqrt{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6/\sqrt{12} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3/\sqrt{12} & 1/2 & 1/2 & 3/\sqrt{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4/\sqrt{12} & 2/\sqrt{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3/\sqrt{6} & 1/\sqrt{2} \end{array}$$

The minimum Euclidean (squared) norm of any nonzero vector in this version of  $E_8$  is 2. The volume of  $\mathbf{R}^8$  associated with each lattice point is 1. The inner product of any pair of lattice vectors is an integer; i.e., this version of  $E_8$  is “self-dual.” This implies that all norms are even integers.

The state spaces of this version of  $E_8$  have sizes  $\{1, 2, 3, 4, 4, 4, 3, 2, 1\}$ . Its trellis diagram is shown in Figure 1. The notation  $(\mathbf{Z}/m\mathbf{Z})/\sqrt{m}$  associated with each coordinate means that the projection of this version of  $E_8$  onto that coordinate is  $m^{-1/2}\mathbf{Z}$  and the cross-section in that coordinate is  $m^{1/2}\mathbf{Z}$ . A branch labelled by an integer  $a$ ,  $0 \leq a \leq m - 1$ , corresponds to the coset  $m^{1/2}(\mathbf{Z} + a)$  of  $m^{1/2}\mathbf{Z}$  in  $m^{-1/2}\mathbf{Z}$ .

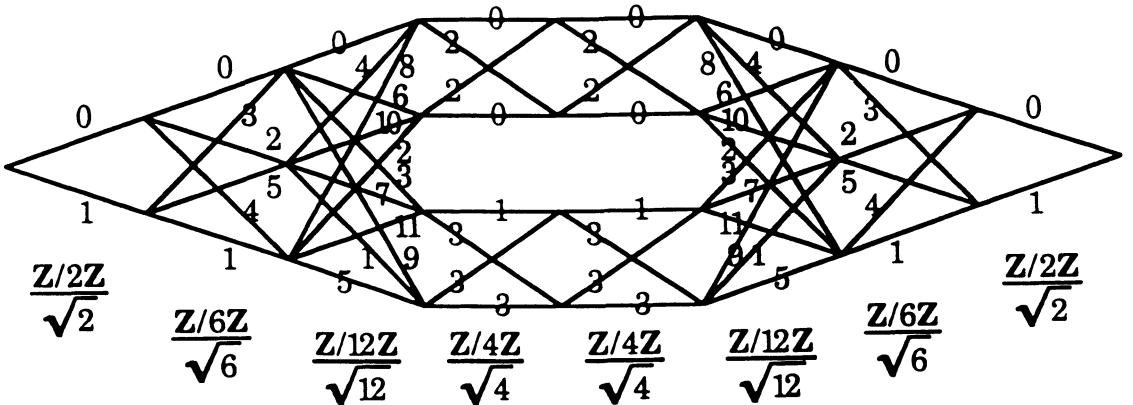


Figure 1. Absolutely minimal trellis diagram for  $E_8$ .

The cross-section in the first two coordinates is a version of the hexagonal lattice  $A_2$  with minimum norm 2. Similarly, the cross-sections in the first  $3, 4, \dots, 7$  coordinates are versions of  $D_3, D_4, D_5, E_6$  and  $E_7$  with minimum norm 2.

Furthermore, it follows from the self-duality of this version of  $E_8$  that the corresponding projections are the dual lattices  $A_2^\perp$ ,  $D_3^\perp D_4^\perp$ ,  $D_5^\perp$ ,  $E_6^\perp$  and  $E_7^\perp$  [19]. Since a cross-section is a sublattice of the corresponding projection, the cross-sections are each sublattices of their duals — i.e., they are “integral lattices” [18]. The state spaces are the corresponding quotient groups (“glue groups”):

$$\begin{aligned}(\mathbf{Z}/2\mathbf{Z})/\sqrt{2} &\cong \mathbf{Z}_2; \\ A_2^\perp/A_2 &\cong \mathbf{Z}_3; \\ D_3^\perp/D_3 &\cong \mathbf{Z}_4; \\ D_4^\perp/D_4 &\cong (\mathbf{Z}_2)^2; \\ D_5^\perp/D_5 &\cong \mathbf{Z}_4; \\ E_6^\perp/E_6 &\cong \mathbf{Z}_3; \\ E_7^\perp/E_7 &\cong \mathbf{Z}_2.\end{aligned}$$

Thus, in a single common coordinate system, this version of  $E_8$  explicitly exhibits versions of the densest lattices in dimensions  $1 \leq i \leq 8$  as integral lattices  $\Lambda_i$  with even norms, their duals  $\Lambda_i^\perp$ , and their glue groups  $\Lambda_i^\perp/\Lambda_i$  (as state spaces).

For decoding using the Viterbi algorithm, the number of trellis branches and the number of two-way comparisons are more important measures of complexity than the numbers of states. The total number of branches in this trellis diagram is 56, and the number of two-way comparisons is 33. By contrast, there is a well-known trellis diagram for  $E_8$  with state spaces of sizes  $\{1, 2, 4, 8, 4, 8, 4, 2, 1\}$  [13] in which the total number of branches is only 44, and the number of comparisons is only 11. From this perspective, the “absolutely minimal” trellis diagram of Figure 1 is of more theoretical than practical interest.

The norms of lattice vectors in this coordinate system can be easily read from the trellis. Figure 2 shows the minimum norm for each branch that contributes to a minimum-norm

lattice vector. Branches for which there are a pair of parallel minimum nonzero-norm transitions have been darkened.

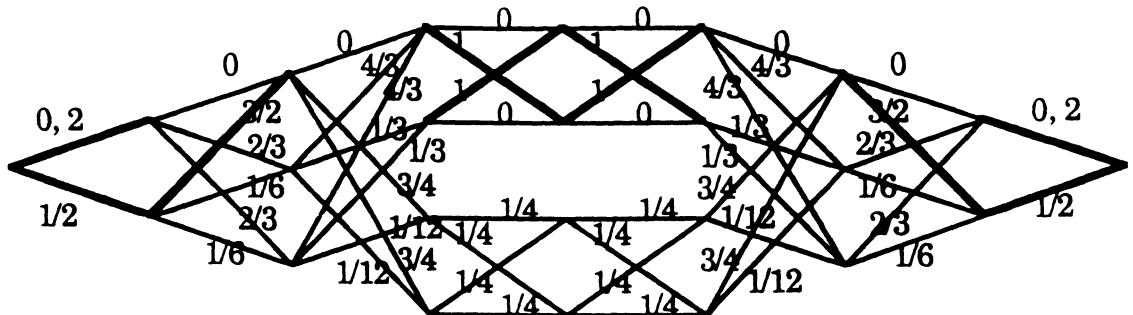


Figure 2. Minimum norms of branches associated with minimum-norm paths.

From this diagram, the number of norm-2 paths of each “shape” terminating at each time  $i$ ,  $1 \leq i \leq 8$ , can be easily determined. They are tabulated in Table 1.

| shape                                      | number terminating at time $i$ |   |   |    |    |    |    |     | total |
|--|--------------------------------|---|---|----|----|----|----|-----|-------|
|  | 1                              | 2 | 3 | 4  | 5  | 6  | 7  | 8   |       |
| (2, 0, 0, 0, 0, 0, 0, 0)                   | 2                              | 0 | 0 | 0  | 0  | 0  | 0  | 0   | 2     |
| (1/2, 3/2, 0, 0, 0, 0, 0, 0)               | 0                              | 4 | 0 | 0  | 0  | 0  | 0  | 0   | 4     |
| (0, 2/3, 4/3, 0, 0, 0, 0, 0)               | 0                              | 0 | 2 | 0  | 0  | 0  | 2  | 0   | 4     |
| (1/2, 1/6, 4/3, 0, 0, 0, 0, 0)             | 0                              | 0 | 4 | 0  | 0  | 0  | 0  | 0   | 4     |
| (0, 2/3, 1/3, 1, 0, 0, 0, 0)               | 0                              | 0 | 0 | 4  | 4  | 0  | 8  | 0   | 16    |
| (1/2, 1/6, 1/3, 1, 0, 0, 0, 0)             | 0                              | 0 | 0 | 8  | 8  | 0  | 0  | 16  | 32    |
| (0, 0, 0, 1, 1, 0, 0, 0)                   | 0                              | 0 | 0 | 0  | 4  | 0  | 0  | 0   | 4     |
| (0, 2/3, 1/3, 0, 0, 1/3, 2/3, 0)           | 0                              | 0 | 0 | 0  | 0  | 0  | 4  | 0   | 4     |
| (1/2, 1/6, 1/3, 0, 0, 1/3, 2/3, 0)         | 0                              | 0 | 0 | 0  | 0  | 0  | 8  | 8   | 16    |
| (1/2, 1/6, 1/3, 0, 0, 1/3, 1/6, 1/2)       | 0                              | 0 | 0 | 0  | 0  | 0  | 0  | 16  | 16    |
| (0, 0, 3/4, 1/4, 1/4, 3/4, 0, 0)           | 0                              | 0 | 0 | 0  | 0  | 8  | 0  | 0   | 8     |
| (0, 2/3, 1/12, 1/4, 1/4, 3/4, 0, 0)        | 0                              | 0 | 0 | 0  | 0  | 8  | 8  | 0   | 16    |
| (0, 2/3, 1/12, 1/4, 1/4, 1/12, 2/3, 0)     | 0                              | 0 | 0 | 0  | 0  | 0  | 8  | 0   | 8     |
| (1/2, 1/6, 1/12, 1/4, 1/4, 3/4, 0, 0)      | 0                              | 0 | 0 | 0  | 0  | 16 | 0  | 16  | 32    |
| (1/2, 1/6, 1/12, 1/4, 1/4, 1/12, 2/3, 0)   | 0                              | 0 | 0 | 0  | 0  | 0  | 16 | 16  | 32    |
| (1/2, 1/6, 1/12, 1/4, 1/4, 1/12, 1/6, 1/2) | 0                              | 0 | 0 | 0  | 0  | 0  | 0  | 32  | 32    |
| Total                                      | 2                              | 4 | 6 | 12 | 16 | 32 | 54 | 114 | 240   |

Table 1. Number, shape, and span of norm-2 points.

The cumulative totals yield the “kissing numbers”  $\{2, 6, 12, 24, 40, 72, 126, 240\}$  of the densest lattices  $\Lambda_i$  in dimensions  $1 \leq i \leq 8$ .

### III Something Borrowed

Remarkably, it took more than 20 years for someone to notice that the string decoder described above is not in fact equivalent to the Viterbi algorithm.

In 1991, Andi Loeliger [20] and Jim Massey observed that while the Minty algorithm solves the shortest-path problem for a graph with undirected edges, it may give the wrong solution when the edges are directed, as they are in trellis diagrams. For instance, it is conceivable that it could decode the darkened path in Figure 3 as the most likely path in the trellis of Figure 5b, which is nonsense. So the Minty algorithm is not a maximum likelihood decoder for trellises after all.

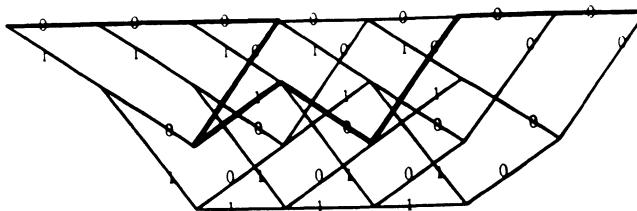


Figure 3. Possible decoded path with a string decoder (Minty algorithm).

### IV Something Bluesy

I love to see  
That old man Jim come 'round.  
I love to see  
Professor Jim go 'round.  
'Cause he loves life  
Much as any man in town.

He's got a good woman;  
She loves him every day.  
Got plenty of students;  
They love him every way.  
Got millions of friends who  
Should all be here today.

He says he's 60!  
I can't believe it, no way.  
He can't be 60 –  
Was 30 yesterday.  
Well, *tempus* may *fugit*;  
We'll toast him anyway:

*Chorus:*

Here's to Jim, to your health, to long life, to negentropy!  
To your talk, to your song, to your joyful society! (IT!)  
To life, to friends, let's drink to capacity! (or  $R_0 \dots$ )

## References

- [1] J. L. Massey, *Threshold Decoding*. Cambridge, MA: MIT Press, 1963.
- [2] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1962.
- [3] G. D. Forney, Jr., "Final report on a coding system design for advanced solar missions," Contract NAS2-3637, NASA CR73176, NASA Ames Research Center, Moffett Field, CA, Dec. 1967.
- [4] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260-269, 1967.
- [5] G. J. Minty, "A comment on the shortest-route problem," *Oper. Res.*, vol. 5, p. 724, 1957.
- [6] A. J. Viterbi, "From proof to product," *1990 IEEE Communication Theory Workshop*, Ojai, CA, 1990.
- [7] P. Elias, "Error-free coding," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 29-37, 1954.
- [8] J. L. Massey and M. K. Sain, "Codes, automata, and continuous systems: Explicit interconnections," *IEEE Trans. Auto. Control*, vol. AC-12, pp. 644-650, 1967.
- [9] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Computers*, vol. C-17, pp. 330-337, 1968.
- [10] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720-738, 1970.
- [11] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76-80, 1978.
- [12] J. L. Massey, "Foundation and methods of channel coding," *Proc. Intl. Conf. on Inform. Theory and Systems*, Berlin, NTG-Fachberichte, vol. 65, pp. 148-157, 1978.
- [13] G. D. Forney, Jr., "Coset codes — Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1152-1187, 1988.
- [14] A. D. Kot and C. Leung, "On the construction and dimensionality of linear block code trellises," *Proc. 1993 IEEE Intl. Symp. Inform. Theory*, San Antonio, TX, p. 291, 1993.
- [15] J. L. Massey, T. Mittelholzer, T. Riedel, and M. Vollenweider, "Ring convolutional codes for phase modulation," *Proc. 1990 IEEE Intl. Symp. Inform. Theory*, San Diego, CA, p. 176, 1990.
- [16] G. D. Forney, Jr. and M. D. Trott, "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1491-1513, 1993.

- [17] H.-A. Loeliger and T. Mittelholzer, “Convolutional codes over groups,” submitted to *IEEE Trans. Inform. Theory*, 1992.
- [18] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 2d ed. New York: Springer-Verlag, 1992.
- [19] G. D. Forney, Jr., “Density/length profiles and trellis complexity of lattices,” in preparation.
- [20] R. C. Davis and H.-A. Loeliger, “A nonalgorithmic maximum likelihood decoder for trellis codes,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 1450-1453, 1993.

# An Inequality on the Capacity Region of Multiaccess Multipath Channels

Robert G. Gallager

Massachusetts Institute of Technology  
Cambridge, MA 02139 USA

## Dedication

This paper is dedicated to James L. Massey on the occasion of his 60th birthday. His works on coding, on multiaccess, and on information theory have all had an important impact on the ideas here. May his 60's be as much fun and as much an inspiration to all of us as his earlier decades.

## Abstract

The effects of multiaccess sources and time varying multipath are considered in analyzing a system in which multiple sources communicate with a fixed base station. We discuss detection and the use of stripping in a multiaccess multipath environment. We finally derive a capacity for these systems. It turns out that CDMA type systems are inherently capable (theoretically) of higher rates than systems such as slow frequency hopping that maintain orthogonality between users.

## I Introduction

Wireless communication has become a very active research area in recent years. This activity is stimulated partly by the success of the Qualcomm code division multiple access (CDMA) system [4,7] and the European GSM system [5], partly by the rapidly growing wireless market, and partly by the ability of VLSI technology to implement very sophisticated systems inexpensively. While it is not difficult to learn about the particular systems being implemented today, it is quite difficult to contrast the basic merits of these approaches. This difficulty comes partly from the inherent complexity of these systems and partly from some degree of sales orientation in much of the technical literature in the area. Our purpose in this paper is to provide some perspective on the concepts and theory underlying this area.

The systems of interest here are those with a set of nonstationary subscribers communicating with a fixed base station. The subscriber sets could be in vehicles or be hand held, the communication could be voice or data, and the area of interest could be urban, rural, or inside a building. Systems with multiple base stations, and the inherent problems, first, of handoff from one base station to another and, second, of interference between the cells associated with different base stations are important, but they will not be addressed here.

## II Characterization of Fading Multipath Channels

The major distinguishing characteristics of wireless communication are the fading effects and the multiaccess effects. In this section, we discuss the characterization of fading multipath channels in the context of point to point communication; in the next section, we include multiaccess effects. Since the subscriber location is arbitrary and not chosen for its propagation characteristics, multiple propagation paths typically exist from transmitter to receiver. We denote the strength of the  $i$ th such propagation path by  $a_i$  and the propagation delay by  $\tau_i$ . These strengths and delays vary with time, and thus can be characterized as  $a_i(t)$  and  $\tau_i(t)$ . If a signal  $x(t)$  is transmitted, it is received as

$$y(t) = \sum_i a_i(t)x(t - \tau_i(t)) \quad (1)$$

The observed waveform at the output is then  $y(t) + n(t)$  where  $n(t)$  is additive noise. It can be seen that the effect of the multipath is the same as the effect of a linear filter (although the filter here is time varying), and thus we can represent the effect of the multipath as a filter impulse response,  $g(\tau, t)$  given by

$$g(\tau, t) = \sum_i a_i(t)\delta(\tau - \tau_i(t)) \quad (2)$$

$$y(t) = \int x(t - \tau)g(\tau, t)d\tau \quad (3)$$

Note that  $g(\tau, t)$  can be interpreted as the response at time  $t$  to an impulse  $\tau$  seconds earlier at the input. One benefit of representing the multipath as a time varying impulse response  $g(\tau, t)$  is that  $a_i(t)$  might be frequency selective, and  $g(\tau, t)$  can incorporate this effect. The major benefit, however, is that (3) reduces the problem of point to point communication through a multipath channel to a very familiar problem—communication through a linear filter channel with additive noise.

Now let us assume that the input is limited to some band  $W$  of frequencies around some center frequency  $f_0$ . Then we can express the input and output to the channel in terms of the baseband complex equivalent waveforms  $u(t)$  and  $v(t)$ ,

$$x(t) = Re[u(t) \exp(j2\pi f_0 t)] = Re[u(t)] \cos(2\pi f_0 t) - Im[u(t)] \sin(2\pi f_0 t) \quad (4)$$

$$y(t) = Re[v(t) \exp(j2\pi f_0 t)] = Re[v(t)] \cos(2\pi f_0 t) - Im[v(t)] \sin(2\pi f_0 t) \quad (5)$$

Substituting (4) and (5) into (1),

$$\begin{aligned} Re[v(t) \exp(j2\pi f_0 t)] &= \sum_i a_i(t) Re[u(t - \tau_i(t))] \exp[j2\pi f_0(t - \tau_i(t))] \\ &= Re \sum_i a_i(t) \exp[-j2\pi f_0 \tau_i(t)] u(t - \tau_i(t)) \exp[j2\pi f_0 t] \end{aligned} \quad (6)$$

Defining  $\alpha_i(t) = a_i(t) \exp[-j2\pi f_0 \tau_i(t)]$  and  $h(\tau, t) = \sum_i \alpha_i(t) \delta(\tau - \tau_i(t))$  as the complex baseband equivalent of the path strengths and filter response, we get the baseband equation

$$v(t) = \sum_i \alpha_i(t) u(t - \tau_i(t)) = \int u(t - \tau) h(\tau, t) d\tau \quad (7)$$

The baseband received waveform is then  $r(t) = v(t) + z(t)$  where  $z(t)$  is the baseband noise, i.e.,  $n(t) = \text{Re}[z(t) \exp[j2\pi f_0 t]]$  is the noise in the bandwidth  $W$  around  $f_0$ .

Note that  $\alpha_i(t)$  is complex, and its phase changes with  $\tau_i(t)$ . As an example of what these equations mean, consider Figure 1 in which a vehicle traveling at 60 km/hour is transmitting to a base station behind it, and there is both the direct path and a path with a reflection from a smooth wall in front of the vehicle. Path 1, the direct path, is getting longer, and the propagation delay on this path is increasing at 55 nsec per second. Path 2 is getting shorter at 55 nsec per second. Thus the baseband path strengths,  $\alpha_i(t) = a_i(t) \exp[-j2\pi f_0 \tau_i(t)]$ , are rotating in opposite directions as shown in Figure 1b. These linearly changing phases correspond to Doppler shifts in the received frequency. The important feature here is that different paths have different Doppler shifts, and that the amount of shift depends not only on the speed of the transmitter or receiver but also on the location of the reflector. These different Doppler shifts give rise to frequency selective fading over the transmitted waveform.

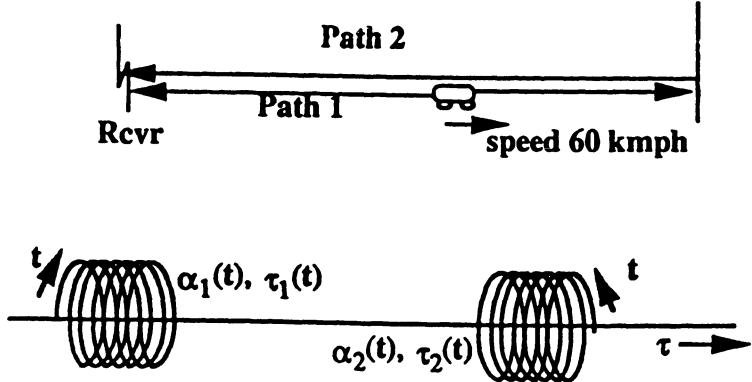


Figure 1: Path Delays and Doppler

It is often useful to view (7) in sampled form. By assumption, the bandpass waveforms have a bandwidth of  $W$ , with  $W/2$  on each side of the center frequency  $f_0$ . Thus  $u(t)$  and  $v(t)$  are constrained to bandwidth  $W/2$  and are determined by their samples  $u(i/W)$  and  $v(i/W)$  for integer  $i$ . Equation (7) then becomes

$$v\left(\frac{i}{W}\right) = \sum_k u\left(\frac{i-k}{W}\right) \tilde{h}[k, i]$$

$$\tilde{h}[k, i] = \int \frac{\sin(\pi(k - \tau W))}{\pi(k - \tau W)} h(\tau, i/W) d\tau \quad (8)$$

$\tilde{h}[k, i]$  provides a tapped delay line model for the channel, giving the complex value of the  $k$ th tap at time  $i/W$ . These taps depend on the bandwidth of the input, but suffice to provide the input output relation on the channel given the bandwidth constraint. Note that the output  $v(t)$  can be spread out over a somewhat larger bandwidth than the input because of the time varying filter. M. Medard has carried out this sampling argument carefully and

shown that  $W$  need be no larger than the input bandwidth plus the bandwidth of  $h(\tau, t)$  viewed as a function of  $t$  (*i.e.*, essentially the largest Doppler shift).

For the example of Figure 1, if we take  $W$  to be  $10^6$  and  $f_0$  to be  $10^9$ , we see that it takes about 18 seconds for a path to move from one tap to the next, and the tap strengths rotate at about 55 H. This means that, relative to input sample times, the taps are rotating slowly and the paths are moving even more slowly relative to the taps.

Typical multipath situations are far more complex than that of Figure 1. There are often many more paths, and these paths often separate into a large number of subpaths due to reflection from rough surfaces and other anomalies. The overall delay spread between different paths ranges from less than 100 nsec to more than 15 msec [11]. The tap gains range from being relatively constant to having Rayleigh, Rician, or Nakagami distributions. As we will see later, the detailed way that taps vary is not critical, because viable communication systems must deal with all these cases, and typically must track the tap gains. The speed with which taps vary is important, but we have already seen how Doppler shifts cause the major part of this variation.

### III Point to Point Detection

Suppose that the transmitter selects the  $m$ th of  $M$  waveforms,  $u_1(t), \dots, u_M(t)$  for transmission in a given interval and suppose there is no multipath. Then the received baseband waveform is  $r(t) = u_m(t) + z(t)$ , and if the noise is white over the signalling bandwidth, it is well known that the optimal detector consists of passing  $r(t)$  through  $M$  matched filters of impulse responses  $u_1^*(-t), \dots, u_M^*(-t)$  (assume that the time reference is chosen so that these filters are realizable; the complex conjugates are necessary because the signals are complex baseband signals). The decision is then made from the outputs sampled at time 0.

Next suppose there is multipath but  $h(\tau, t)$  is known. This can be viewed as the same detection problem as above, replacing  $u_m(t)$  with the filtered signal  $v_m(t) = \int u_m(t - \tau)h(\tau, t)d\tau$ ,  $1 \leq m \leq M$ , and replacing the matched filters  $u_m^*(-t)$  with the matched filters  $v_m^*(-t)$ . Finally, consider the case where the multipath  $h(\tau, t)$  is unknown. The matched filters  $v_m^*(-t)$  are then unknown, but  $v_m^*(-t)$  is the convolution of  $u_m^*(-t)$  with  $h^*(-\tau, -t)$ . A rake receiver [9,10] can be viewed as a receiver that first passes  $r(t)$  through the matched filters  $u_m^*(-t)$ , uses the outputs to help estimate  $h(\tau, t)$ , and then completes the matched filtering with the estimated  $h^*(-\tau, -t)$  (see Figure 2).

This becomes particularly simple in a CDMA system where the signals  $u_m(t)$  are broadband noise-like signals. If  $u_m(t)$  is transmitted, then the output  $w_{m,m'}(t)$  of the matched filter  $u_{m'}^*(-t)$  is

$$w_{m,m'}(t) = \int r(t - \tau)u_{m'}^*(-\tau)d\tau =$$

$$\int \left[ \int u_m(t - \tau - \phi)h(\phi, t - \tau)d\phi + z(t - \tau) \right] u_{m'}^*(-\tau)d\tau$$

$$\approx \int R_{m,m'}(t - \phi)h(\phi, t)d\phi + z_{m'}(t)$$

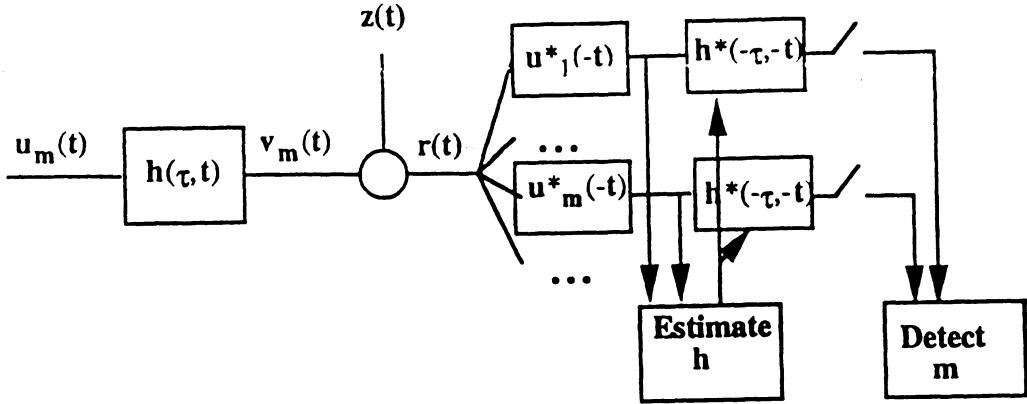


Figure 2: Rake Receiver

where  $R_{m,m'}(t) = \int u_m(t - \tau)u_{m'}^*(-\tau)d\tau$  is the cross-correlation of the input signals and  $z_{m'}(t)$  is the response of the matched filter to the noise  $z(t)$ . The approximation here is in assuming that  $h(\tau, t)$  is slowly varying in  $t$  over the multipath spread. For CDMA signals, this cross-correlation function is approximately zero for  $m \neq m'$ . For  $m = m'$ , it is approximately zero for  $t \neq 0$  and is equal to  $S$ , the energy of the signal at  $t = 0$ . Thus  $w_{m,m}(\phi) \approx Sh(\phi, t) + z_m(\phi)$ , and for  $m \neq m'$ ,  $w_{m,m'} \approx z_{m'}(\phi)$ . Assuming one of these  $M$  signals is transmitted each signalling interval, and assuming that  $h(\tau, t)$  changes slowly with  $t$ , the outputs of the matched filters can be used to update the estimate of the tap values  $\tilde{h}(\tau, t)$ , and this can be used for the final part of the matched filter, as shown in Figure 2.

The optimality of the matched filter above is predicated on one of  $M$  signals being transmitted in isolation. If the interval between successive signals is greater than the multipath spread, then successive signal outputs from the matched filters  $u_m^*(-t)$  are isolated. Conversely, if the interval between signals is less than the multipath spread, one should either use a different set of signals on adjacent signalling intervals or should somehow combine the above receiver with adaptive equalization [8].

The receiver described above is called a coherent rake receiver. One must estimate both the phase and magnitude of each tap of  $\tilde{h}$ . Since the phases typically change faster than the magnitudes, it is sometimes better to use a noncoherent rake receiver in which one estimates the magnitude of  $\tilde{h}$  and uses the corresponding matched filter on the magnitude of the outputs of the matched filters  $u_m^*(-t)$ .

For narrow-band signals, successive signals are separated by only one or two sampling intervals. These sampling intervals are typically long with respect to the multipath spread so the sample tap channel  $\tilde{h}$  typically has one large tap. Unfortunately other taps are large enough to cause significant intersymbol interference, necessitating adaptive equalization. In slow frequency hopping systems, this is usually accomplished by sending a known string of symbols in the middle of a hop and training the equalizer on those symbols. It appears that the broadband system is able to “resolve” the channel better than the narrowband system because it evaluates a large number of taps and is able to add the responses from these taps coherently. We look at this point of view more carefully later, and show that it is slightly

misleading.

Although the rake approach to measuring a channel is very different from the narrow-band approach, we see that each, if successful, succeeds in measuring the response of the channel to signals in the given bandwidth and then uses the measured response as if it were the true channel impulse response. To the extent that this measurement can be done precisely, the channel is simply a time varying, but known, channel with additive white Gaussian noise. In what follows, except for occasional comments and warnings, we assume that the channel can be measured precisely. This is a reasonable assumption for a channel with multipath spread on the order of  $10\mu\text{sec}$  or less and Doppler shifts of  $100 \text{ Hz}$  or less, but it appears that such measurements are difficult to make in practice. One possible approach to improving the channel measurement is to take advantage of the fact that the phase rotation of individual paths is due to path length variations, which often tend to be approximately linear in time.

## IV Multiuser Detection

Suppose now that there are  $K$  different subscribers using the same bandwidth  $W$  to communicate with a base station. We will ignore the reverse problem of a base station communicating with multiple subscribers; this reverse problem is important practically, but is less interesting conceptually since there is no interesting problem of interference between the different transmissions. We also restrict attention to broadband (CDMA) transmission here, since it would not make much sense to share a narrow-band between several subscribers at the same time. Let  $u^{(k)}(t)$  denote the transmitted waveform from the  $k$ th source. Let  $h^{(k)}(\tau, t)$  be the output response at time  $t$  to an impulse from source  $k$  that was sent  $\tau$  seconds earlier. Then the received waveform at the base station is given by

$$r(t) = \sum_k v^{(k)}(t) + z(t), \quad \text{where } v^{(k)}(t) = \int u^{(k)}(t - \tau) h^{(k)}(\tau, t) d\tau \quad (9)$$

Optimal detection is quite difficult in such a system, even in the absence of multipath [13]. Figure 3 illustrates the difficulty. Suppose that Source 1 is sending a sequence of data carrying waveforms,  $c_1, c_2, \dots$ , and Source 2 is sending another sequence  $d_1, d_2, \dots$  with an offset in timing. Detection of  $c_1$  is improved by knowledge about  $d_1$  and  $d_2$ , but knowledge about  $d_1$  is similarly improved by knowledge about  $c_0$  and  $c_1$ , and knowledge about  $d_2$  is improved by  $c_1$  and  $c_2$ . Thus detection of  $c_1, c_2, \dots$  and  $d_0, d_1, \dots$  are all linked together.

In CDMA systems, it is common, when detecting Source 1, to model the reception due to Source 2 as white Gaussian noise over the band  $W$ . This makes a certain amount of sense, since pseudo-noise waveforms are used to send data. Unfortunately, if, say, Source 2 has much more power than Source 1, this conventional approach is quite inferior. A more powerful approach is first to decode the symbols of Source 2, treating Source 1 as noise. Then (still assuming no multipath),  $u^{(2)}(t)$  can be subtracted (stripped) from the received waveform  $r(t)$ , and the weak source can then be decoded with no interference from the strong source. This is somewhat counter-intuitive, since it asserts that a source can be detected better in the presence of a very strong interfering source than in the presence of a weaker interfering source. Techniques such as stripping that allow weak data sources to be decoded in the presence of known strong data sources are called near-far resistant.

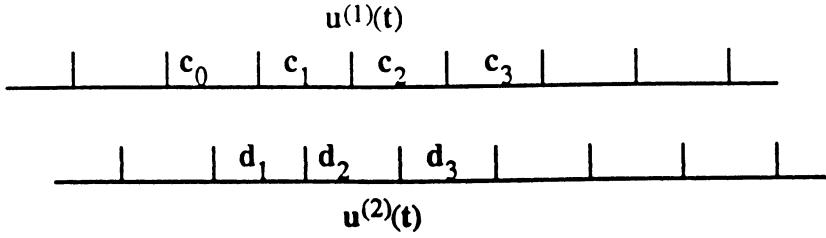


Figure 3: Asynchronous Sources

The situation becomes more complicated in the presence of multipath. Assuming that the multipath can be measured exactly, one can determine  $v^{(2)}(t)$  from knowledge of  $h^{(2)}(\tau, t)$  and from decoding the symbols in  $u^{(2)}(t)$ . However, under the assumption that  $v^{(2)}(t)$  has much higher power than  $v^{(1)}(t)$ , small errors in measuring  $h^{(2)}(\tau, t)$  can leave significant interference in the received signal  $r(t)$  after stripping the estimate of  $v^{(2)}(t)$  from  $r(t)$ .

It is unclear from a practical point of view, and even from a theoretical point of view, how effective stripping high power sources from  $r(t)$  is in the presence of multipath. [12] gives one indication that something akin to stripping might be effective by showing that a noncoherent receiver in Gaussian noise can achieve a form of near-far resistance.

## V Multiuser Information Theory

The idea of decoding strong sources and then stripping their effect from the received waveform is the crucial idea in the multiaccess coding theorem [1,6]. In order to avoid concealing the central ideas in what follows, the development will be heuristic. First, assume that each baseband source waveform  $u^{(k)}(t)$  goes through a linear time invariant filter with impulse response  $h^{(k)}(\tau)$ , so that the received baseband waveform is  $r(t) = \sum_k \int u^{(k)}(t-\tau)h^{(k)}(\tau)d\tau + z(t)$  where  $z(t)$  is a sample function of white Gaussian noise of spectral density  $N_0/2$ . The multiaccess coding theorem then asserts that if source  $k$ ,  $1 \leq k \leq K$  ( $k \geq 2$ ) has rate  $R_k$  and is constrained to a power spectral density  $S_k/W$  over the given band of width  $W$ , then arbitrarily small error probability can be achieved for all sources if for each subset  $A$  of the integers  $\{1, 2, \dots, K\}$ ,

$$\sum_{k \in A} R_k < \int_{-W/2}^{W/2} \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(f)|^2}{WN_0} \right] df \quad (10)$$

where  $H^{(k)}(f)$  is the Fourier transform of  $h^{(k)}(t)$  (see [2] and [3]). If the left side of (10) is larger than the rightside for any subset  $A$ , then arbitrarily small error probability cannot be achieved. It is also true that the encoders need not know the impulse responses  $h^{(k)}(t)$ , although the decoders do need this knowledge. The quantity on the right side of (10) is the conditional average mutual information per unit time,  $I(A)$ , between the inputs in  $A$  and

the output, conditional on the inputs not in  $A$ ; this assumes that the inputs are independent stationary white Gaussian noise (WG) processes over the bandwidth  $W$ .

Next, suppose the baseband linear filters are time varying with impulse responses  $h^{(k)}(\tau, t)$  as before. Let  $T_0$  be the multipath spread of the channels; *i.e.*, for all  $k$  and  $t$ ,  $h^{(k)}(\tau, t) = 0$  for  $\tau < 0$  and for  $\tau > T_0$ . Let  $W_0$  be the Doppler spread of the channels and assume that  $W_0 \ll 1/T_0$ . Consider a bandwidth  $W$  in the range  $W_0 \ll W \ll 1/T_0$ . Such a  $W$  was referred to above as narrowband, but  $W$  is also large enough that we can neglect the effect of in-band signals moving out of band because of Doppler shifts. Let  $H^{(k)}(f, t) = \int h^{(k)}(\tau, t) \exp(-j2\pi f\tau) d\tau$ . Since  $W$  is much less than the coherent bandwidth,  $1/T_0$ , of each filter, we can approximate  $H^{(k)}(f, t)$  as being constant for  $-W/2 \leq f \leq W/2$ . With this approximation, we can simply evaluate the right side of (10) as a rate per unit time of generation of average conditional mutual information,  $I(A, t)$ ,

$$I(A, t) = W \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(0, t)|^2}{WN_0} \right] \quad (11)$$

We now want to view  $H^{(k)}(0, t)$  as a stochastic process in  $t$ , but must recall that (11) is then conditioned on  $H^{(k)}$  for each  $k$ , and  $I(A, t)$  is a random variable that depends on the sample values of  $H^{(k)}(0, t)$  for each  $k$ . We assume that the processes  $\{H_k(0, t); -\infty < t < \infty\}$  are ergodic. We discuss this somewhat unrealistic assumption later. We now claim that if

$$\sum_{k \in A} R_k < E \left\{ W \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(0, t)|^2}{WN_0} \right] \right\} \quad (12)$$

for all sets  $A$ , then reliable communication is possible at the rates  $R_1, \dots, R_K$ . The argument is that for large enough  $T$ , the inequalities

$$\sum_{k \in A} R_k < \frac{1}{T} \int_0^T W \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(0, t)|^2}{WN_0} \right] dt \quad (13)$$

will all be satisfied with high probability, and codes exist over this block-length for which the error probability is then small. In the same way, if one of the inequalities in (12) is reversed, reliable communication is not possible. We now use (12) to compare two scenarios. In the first,  $K$  users each use a separate band of width  $W$ . In the second,  $K$  users each use all  $K$  bands, again each of width  $W$ . Thus, in the second scenario, each user uses the entire bandwidth  $KW$ . We assume that  $W \ll 1/T_0$ , but not that  $KW < 1/T_0$ . Let  $H^{(k)}(i, t)$  denote the response of the  $i$ th of these bands to user  $k$ . For Scenario 1, suppose user  $k$  uses channel  $k$ , so that (12) becomes

$$R_k < E \left\{ W \ln \left[ 1 + \frac{S_k |H^k(k, t)|^2}{WN_0} \right] \right\} \quad (14)$$

Reliable communication for each user is not possible (given the assumed use of bandwidth and power) if each of these inequalities is reversed. Equaton (14) is equivalent to the larger set of equalities over all subsets  $A$ :

$$\sum_{k \in A} R_k < E \left\{ W \sum_{k \in A} \ln \left[ 1 + \frac{S_k |H^k(k, t)|^2}{WN_0} \right] \right\} \quad (15)$$

For Scenario 2, assuming that user  $k$  uses power spectral density  $S_k/KW$  in each band, (12), summed over all bands, becomes

$$\sum_{k \in A} R_k \leq E \left\{ \sum_i W \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(i, t)|^2}{KWN_0} \right] \right\} \quad (16)$$

Assuming that each process  $\{H^{(k)}(i, t); t \geq 0\}$  is statistically identical over all bands  $i$ , this can be rewritten as

$$\sum_{k \in A} R_k \leq E \left\{ KW \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(0, t)|^2}{KWN_0} \right] \right\} \quad (17)$$

Again, reliable communication, given the assumed allocation of power to frequency bands, is possible if (16) and thus (17), is satisfied for all  $A$ ; reliable communication is impossible if any of the inequalities is reversed. Now denote the right side of (15) for a given  $A$  as  $I_{NB}(A)$  and denote the right side of (17) as  $I_{WB}(A)$ . Because of the concavity of the logarithm, we have

$$I_{NB}(A) \leq E \left\{ |A| W \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(k, t)|^2}{|A| W N_0} \right] \right\} \quad (18)$$

where  $|A|$  is the number of sources in  $A$ . The inequality is strict unless  $S_k |H^{(k)}(k, t)|^2$  is the same for all  $k$  with probability 1. Since the sources are assumed to have separate paths, we assume these terms are not all the same, and thus we assume that (18) is satisfied with strict inequality for all  $A$  such that  $|A| \geq 2$ . Since  $x \ln(1 + b/x)$  is increasing in  $x$  for any  $b > 0$ , (18) can be further bounded by

$$I_{NB}(A) < E \left\{ KW \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(k, t)|^2}{KWN_0} \right] \right\} \quad (19)$$

with strict inequality unless  $|A| = K$ . Finally, since the channels are statistically identical,

$$I_{NB}(A) < E \left\{ KW \ln \left[ 1 + \frac{\sum_{k \in A} S_k |H^{(k)}(0, t)|^2}{KWN_0} \right] \right\} = I_{WB}(A) \quad (20)$$

Thus  $I_{NB}(A)$ , for each  $A$ , is strictly less than  $I_{WB}(A)$ . Thus the rates achievable under the wide-band power assumptions of (17) are strictly larger than those achievable under the narrowband assumptions of (15).

Now we observe that slow frequency hopping, hopping between bands of width  $W$ , is also constrained by (15) because of the assumption that the different frequency bands are statistically identical. That is, frequency hopping achieves diversity, allowing an individual user to obtain averaging over channels, but does nothing to enhance average mutual information, which is necessary to achieve high data rates.

Conversely, CDMA is constrained by (17). It also appears, although some proof would be required, that reliable communication is possible using CDMA (in conjunction with long constraint length, low rate, error correcting codes) if (17) is satisfied. In other words, CDMA is theoretically capable of higher data rates than slow frequency hopping.

This result is rather surprising. To make it a little more understandable, we look at a simple example with  $K = 2$ ,  $S = 1$ ,  $W = 1$ ,  $N_0 = 1$ . We also take  $H^k(0, t)$  to be 0 or 1, each with probability 1/2, so that in scenario 1, the received signal power on a single channel is 0 or 1 with equal probability. For Scenario 2, the received signal power on a single channel is 0 with probability 1/4 (if both sources are faded), is 1/2 with probability 1/2 (if one source is faded) and is 1 with probability 1/4 (if neither source is faded). The right side of (15) is then  $\ln 2 = 0.69$ , and the right side of (16) is  $\ln(3/2) + (1/2)\ln(2) = 0.75$ . Since the fading is independent between sources, the use of both sources together tends to partially average the received signal power, which is the effect that increases average mutual information.

One peculiar effect of this result is that  $K$  users, distributed in space, can send more data to a base station than a single user with  $K$  times as much power. Another peculiar effect is that if one achieves orthogonality in any way between the  $K$  users, this lowers the achievable data rate over that possible without orthogonality.

One should not assume that this result implies that CDMA is “better” than slow frequency hopping. In order to approach the rates promised by this result, one must code, or interleave, over a long time period, and one must jointly decode all the sources. The usual technique of viewing the other sources as noise in decoding a given source might give up more than what has been gained here. It is possible that stripping could be used to achieve joint decoding, but as pointed out before, it is not clear that the channel can be measured well enough to make this effective. The best power levels for stripping have been worked out in [14], but multipath effects were not taken into account. With the relatively rapid changes in channel strength due to multipath, one might guess that even if one attempted to keep received power levels the same, actual differences might be enough to make use of stripping. We have also neglected inter-cell interference from our model, and this is an important aspect for practical systems.

We have assumed that the channel can be accurately measured, and it is not clear how dependent our result is on that assumption. Finally, we have assumed in that there is no feedback from base station to sources. If feedback were available, a source could in principle transmit only when the channel is good (or choose a good channel to transmit on). Current systems use feedback to try to maintain constant received signal power from each source, but do not attempt to transmit only when the channel is good, so that the assumptions here are more or less reasonable for these systems.

## References

- [1] R. Ahlswede, “Multi-way communication channels,” *Proc. 2nd Int’l. Symp. Inform. Th.*, Tsahkadsor, Armenian SSR, 1971.
- [2] R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, New York, 1968.
- [3] R. G. Gallager, “A perspective on multiaccess channels”, *IEEE Trans. on Information Theory*, Vol. IT-31, No. 2, 1985.

- [4] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, and C. E. Wheatley, “On the capacity of a cellular CDMA system”, *IEEE Trans. Vehic. Tech.*, Vol. 40, No. 2, pp. 303-312.
- [5] L. Hanzo and J. Stefanov, *The Pan-European Digital Cellular Mobile Radio System—Known as GSM*, Chapter 8 in *Mobile Radio Communications*, R. Steele, Ed., Pentech Press, London, 1992.
- [6] H. Liao, “A coding theorem for multiple access communications”, *Proc. Int. Symp. IT*, Asilomar, CA, 1972; also *Multiple Access Channels*, Ph.D thesis, Dept. EE, Univ. Hawaii.
- [7] Qualcom, “An Overview of the Application of CDMA to Digital Cellular System and Personal Communications Networks”, submission to the *Cellular Telecommunications Industry Association*, 1991.
- [8] S. U. H. Quershi, “Adaptive equalization”, *Proc. IEEE*, Vol. 73, pp. 1349-87, 1985.
- [9] R. Price, “Optimum detection of random signals in noise with applications to scatter-multipath communication”, *IRE Trans. on Information Theory*, Vol. IT-2, pp. 125-135, 1956.
- [10] R. Price and P. Green, “A communication technique for multipath channels”, *Proc. IRE*, Vol. 46, pp. 555-570, 1958.
- [11] G. L. Turin, “Spread spectrum antimultipath techniques”, *Proc. IEEE*, Vol. 68, pp. 328-353, 1980.
- [12] M. K. Varanasi, “Noncoherent detection in asynchronous multiuser channels”, *IEEE Trans. Information Theory*, Vol. IT-39, No. 1, pp. 157-176., 1993.
- [13] S. Verdu, “Minimum probability of error for asynchronous multiuser channels”, *IEEE Trans. Information Theory*, Vol. IT-32, No. 1, pp. 890-897, 1986.
- [14] A. J. Viterbi, “Very low rate convolutional codes for maximum theoretical performance of spread-spectrum multiple-access channels”, *IEEE JSAC*, Vol. 8, No. 4, 1990.

# A Finite Field Fourier Transform for Vectors of Arbitrary Length

Christoph G. Günther  
Ascom Tech Ltd, Gewerbepark  
5506 Mägenwil, Switzerland

## Abstract

Finite field Fourier transforms are of great interest in coding and cryptography. They are, in particular, used for describing BCH and RS codes in the spectral domain and for representing the solutions of recurrence equations used in stream ciphers. So far, finite field Fourier transforms have only been defined on vectors that have a length which is relatively prime to the characteristic of the field. The aim of the paper is to generalize this definition to arbitrary lengths. Many properties get a simpler interpretation with this approach.

## I Introduction

Consider a vector  $\underline{v}$  of length  $N$  whose elements are from a finite field  $\text{GF}(q)$ . Let  $p$  denote the characteristic of that field and assume that  $p$  does not divide  $N$ . Then there is a smallest integer  $\mu$  such that  $N|q^\mu - 1$ , and we can find an  $N$ -th root of unity  $\alpha$  in the extension field  $\text{GF}(q^\mu)$ . In this situation, the finite field Fourier transform from  $\text{GF}(q)$  into the extension field  $\text{GF}(q^\mu)$  can be defined by (see for example [1]):

$$V_k := \sum_{x=0}^{N-1} v_x \alpha^{kx}, \quad \text{in } \text{GF}(q^\mu)$$

The original vector  $\underline{v}$  can be recovered from  $\underline{V}$  through the relation (Fourier theorem)

$$v_x = \frac{1}{N} \sum_{k=0}^{N-1} V_k \alpha^{-kx}, \quad \text{in } \text{GF}(q).$$

The finite field Fourier transform has all properties of a Fourier transform: linearity, unitarity (Parseval's theorem) and the convolution property. Furthermore, it has a very important algebraic property, called Blahut's theorem, which is that the linear complexity of the sequence  $\underline{v}$  is equal to the Hamming weight of  $\underline{V}$ . That property was key in Blahut's description [2] of the decoding of Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes, which was the first description of these codes that was intuitively understandable.

All of the above statements hold in the case  $\gcd(N, p) = 1$ . Now assume that

$$N = n p^\nu,$$

with  $\gcd(n, p) = 1$ , then

$$z^N - 1 = z^{np^\nu} - 1 = (z^n - 1)^{p^\nu},$$

i.e., every  $N$ -th root of unity is also an  $n$ -th root of unity. This means that the powers of  $\alpha$  do not represent a complete set of functions in  $(\text{GF}(q))^N$  anymore. In Section III, we will introduce additional functions in order to restore completeness. This will lead to a generalization of the finite field Fourier transform. Similar generalizations have been obtained independently by Mathys [3] in 1990, by Blackburn [4] in 1994, and by the author [5] in 1987. The paper is a more detailed exposition of [5], with most of the results being now derived from a single powerful lemma, called convolution lemma.

The generalized finite field Fourier transform will be used to describe the solution of linear recursions (Section IV) and the decoding of Reed-Muller codes (Section V). Both results are known in a different form but will get a simpler interpretation in the present context.

This paper focusses on new insights into existing results. Jim Massey, to whom I dedicate the paper on the occasion of his 60th birthday, always finds ways to derive and express results in a way that shows much symmetry and beauty. I hope that he will enjoy this modest contribution.

## II Binomial Coefficients in Finite Fields

Binomial coefficients in fields  $\text{GF}(p)$ ,  $p$  prime, play a crucial role in the generalization of the finite field Fourier transform. The present section introduces those coefficients and summarizes their most important properties.

The binomial coefficient in the finite field  $\text{GF}(p)$  is defined by the recursion

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi - 1 \\ \eta \end{bmatrix} + \begin{bmatrix} \xi - 1 \\ \eta - 1 \end{bmatrix} \quad (\text{mod } p), \quad (1)$$

and the conditions

$$\begin{bmatrix} \xi \\ 0 \end{bmatrix} = 1, \quad \begin{bmatrix} \xi \\ \xi \end{bmatrix} = 1. \quad (2)$$

Although the recursion and the initial conditions are the same as in the rational case ( $\text{mod } p$  is missing in that case), we have used square brackets  $[]$  instead of round brackets  $()$  to indicate that the present coefficients have additional properties, which we now summarize.

The first major property was already discovered during the last century by Lucas [6]. It says that the coefficients have a recursive structure very similar to Hadamard matrices:

**Lemma 1** (Lucas' lemma [6]) *Let  $\xi = \sum_{j=0}^{\nu-1} \xi_j p^j$ , and let  $\eta = \sum_{j=0}^{\nu-1} \eta_j p^j$ , then*

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \prod_{j=0}^{\nu-1} \begin{bmatrix} \xi_j \\ \eta_j \end{bmatrix} \quad (\text{mod } p) \quad (3)$$

Lucas's Lemma has three important implications: The first one is concerned with the exchange of the first and second argument of the coefficients. It will be used in the context of the convolution theorem:

**Corollary 2** For any numbers  $\xi, \eta \in \{0, 1, \dots, p^\nu - 1\}$ , we have:

$$\begin{bmatrix} p^\nu - 1 - \eta \\ p^\nu - 1 - \xi \end{bmatrix} = (-1)^{\xi+\eta} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \pmod{p}.$$

The second implication gives the period of the binomial coefficients:

**Corollary 3** The minimal period of the binomial coefficients  $\begin{bmatrix} \xi \\ \eta \end{bmatrix}$  with respect to  $\xi$ , i.e., the smallest integer  $T$  such that  $\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi+T \\ \eta \end{bmatrix}$  for all  $\xi$ , is given by

$$p^{\lceil \log_p \eta \rceil}. \quad (4)$$

Both results are proved in the appendix. The last implication is due to Massey, Costello, and Justesen [14]. It gives the Hamming weight of the vectors of binomial coefficients. Let

$$\begin{bmatrix} \cdot \\ \eta \end{bmatrix} = \left( \begin{bmatrix} 0 \\ \eta \end{bmatrix}, \begin{bmatrix} 1 \\ \eta \end{bmatrix}, \dots, \begin{bmatrix} p^\nu - 1 \\ \eta \end{bmatrix} \right), \quad (5)$$

and let  $\omega_{MCJ}(\eta)$  denote the Massey-Costello-Justesen weight of the index  $\eta := \sum_{j=0}^{\nu-1} \eta_j p^j$ , which is defined by

$$\omega_{MCJ}(\eta) := \prod_{j=0}^{\nu-1} (p - \eta_j), \quad (6)$$

then the following holds:

**Lemma 4** (Massey-Costello-Justesen lemma [14]) The Hamming weight of the  $\eta$ -th vector of binomial coefficients, i.e., the number of values of  $\xi$  for which  $\begin{bmatrix} \xi \\ \eta \end{bmatrix}$  does not vanish, is given by:

$$\omega_H\left(\begin{bmatrix} \cdot \\ \eta \end{bmatrix}\right) = \omega_{MCJ}(\eta). \quad (7)$$

This lemma will be used in the discussion of Reed-Muller codes.

The second major property of the binomial coefficients over finite fields is to our knowledge new. It is the basis for most of the succeeding results.

**Lemma 5** (Convolution lemma) For any value of  $\xi, \kappa, \lambda \in \{0, 1, \dots, p^\nu - 1\}$ , we have:

$$\begin{aligned} \sum_{\eta=0}^{p^\nu-1} (-1)^\eta \begin{bmatrix} \kappa \\ \eta \end{bmatrix} \begin{bmatrix} \xi + \eta \\ \lambda \end{bmatrix} &= \begin{cases} (-1)^\kappa \begin{bmatrix} \xi \\ \lambda - \kappa \end{bmatrix} & \text{if } \lambda \geq \kappa, \\ 0 & \text{otherwise} \end{cases} \\ \sum_{\eta=0}^{p^\nu-1} (-1)^\eta \begin{bmatrix} \kappa \\ \eta \end{bmatrix} \begin{bmatrix} \xi - \eta \\ \lambda \end{bmatrix} &= \begin{cases} \begin{bmatrix} \xi - \kappa \\ \lambda - \kappa \end{bmatrix} & \text{if } \lambda \geq \kappa, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The proof of this very powerful lemma is surprisingly simple. It is given in the appendix.

### III Generalized Finite Field Fourier Transform

The generalized finite field Fourier transform can be defined in several ways. We will focus on a form which was originally constructed through matrix representations of the cyclic group of order  $N$  in extension fields of  $\text{GF}(p)$ . Alternative approaches are proposed by Mathys in [3] and Blackburn in [4]. Let us begin by introducing a suitable representation of the numbers from the index set  $\{0, 1, \dots, N - 1\}$ . Since  $\gcd(n, p) = 1$ , the equation

$$x = \tilde{x} n + x' p^\nu, \quad (8)$$

defines a unique mapping from  $x \in \{0, \dots, N - 1\}$  to  $(\tilde{x}, x') \in \{0, 1, \dots, p^\nu - 1\} \times \{0, 1, \dots, n - 1\}$ :

$$\tilde{x} = xn^{-1} \pmod{p^\nu}, \quad x' = xp^{-\nu} \pmod{n}. \quad (9)$$

A similar representation holds for  $k$ .

With these notations, the generalized finite field Fourier transform of the vector  $\underline{v}$  becomes:

$$V_k := \sum_{x=0}^{N-1} v_x (-1)^{\tilde{x}} \begin{bmatrix} \tilde{k} \\ \tilde{x} \end{bmatrix} \alpha^{k'x'}, \quad \text{in } \text{GF}(q^\mu), \quad (10)$$

where  $\mu$  is the smallest integer such that  $N|q^\mu - 1$ . The definition of the generalized transform is justified by the following important result, which states that the original vector can be uniquely recovered from the coefficients

**Theorem 6** (Fourier theorem) *Let  $V_k \in \text{GF}(q^\mu)$  be given by Equation (10) then*

$$v_x = \frac{1}{n} \sum_{k=0}^{N-1} V_k (-1)^{\tilde{k}} \begin{bmatrix} \tilde{x} \\ \tilde{k} \end{bmatrix} \alpha^{-k'x'}, \quad \text{in } \text{GF}(q). \quad (11)$$

The proof follows directly from Lemma 5 ( $\xi = 0$ ) and from the property  $\begin{bmatrix} 0 \\ n \end{bmatrix} = \delta_{n,0}$ .

Note that, after a reordering of the vector  $\underline{v}$  and of the Fourier coefficients  $\underline{V}$ , the Equations (10) and (11) also hold without primes and tildes. Since the derivations of the various results all use the prime and tilde variables, however, we found it more convenient to include the decomposition in the definition itself.

In the following, we summarize some simple properties of the generalized transform:

- In the case  $\nu = 0$ , i.e.,  $\gcd(N, p) = 1$ , the generalized finite field Fourier transform becomes the traditional one.
- Both the Fourier transform  $\underline{V}$  and the inverse transform obtained from Equation (11) are periodic with period  $N$ . This is implied by Corollary 3 and  $\alpha^n = 1$ .
- Any technique for the fast computation of finite field Fourier transforms, like Cooley-Tukey or Good-Thomas (see [1]), can also be applied to the generalized transform. The defining Equation (10) already splits the computation into a traditional transform and one with respect to the binomial base functions. The latter one can be computed using the representation from Lucas' lemma (Lemma 1), which reduces the complexity of that part from  $p^\nu$  to  $p\nu$  operations per coefficient.

- The following functions of  $x \in \{0, 1, \dots, N - 1\}$

$$e_x^{(k)} = \begin{bmatrix} \tilde{x} \\ k \end{bmatrix} \alpha^{-k'x'} \quad (12)$$

with  $k \in \{0, 1, \dots, N - 1\}$ , form a complete set of linearly independent functions.

- The property  $v_x \in \text{GF}(q)$  implies

$$v_x = v_x^q = \frac{1}{n} \sum_{k=0}^{N-1} V_k^q (-1)^{\tilde{k}} \begin{bmatrix} \tilde{x} \\ \tilde{k} \end{bmatrix} \alpha^{-k'x'q}, \quad (13)$$

and using the independence of the functions (12):

$$V_{\tilde{k}, qk'} = V_{\tilde{k}, k'}, \quad (14)$$

where  $V_{\tilde{k}, k'} := V_{\tilde{k}n+k'p^\nu}$ . Define  $\mu_{k'}$  to be the smallest integer such that  $\alpha^{-k'} \in \text{GF}(q^{\mu_{k'}})$ , then an iterated use of Equation (14) (and  $\alpha^{-k'q^{\mu_{k'}}} = \alpha^{-k'}$ ) leads to the conclusion that  $V_k \in \text{GF}(q^{\mu_{k'}})$ . The relations (14) are called conjugacy constraints. They induce a splitting of the sum into individual sums, each one running over a cyclotomic set. Making this structure explicit is very useful in certain cases.

Let us now discuss a property which will be of central importance to Section IV and to some extent also to Section V. Let us begin by defining a reordering of the vectors  $\underline{v}$  and  $\underline{V}$ :

$$v_{\tilde{x}, x'}^* = v_{p^\nu-1-\tilde{x}, x'} \quad (15)$$

$$V_{\tilde{k}, k'}^* = V_{p^\nu-1-\tilde{k}, k'} \quad (16)$$

then Corollary 2 implies that the reordered vectors in sequence and Fourier space are related by

$$V_k^* = (-1)^{\tilde{k}} \sum_{x=0}^{N-1} v_x^* \begin{bmatrix} \tilde{x} \\ \tilde{k} \end{bmatrix} \alpha^{k'x'} \quad (17)$$

$$v_x^* = (-1)^{\tilde{x}} \frac{1}{n} \sum_{k=0}^{N-1} V_k^* \begin{bmatrix} \tilde{k} \\ \tilde{x} \end{bmatrix} \alpha^{-k'x'}.$$

These reordered vectors allow a simple formulation of the following important result:

**Theorem 7** (Convolution theorem) *Let  $\underline{U}^*$  and  $\underline{u}^*$  as well as  $\underline{V}$  and  $\underline{v}$  be related by (17) and (10), respectively, then:*

$$\sum_{y=0}^{N-1} u_y^* v_{x+y} = \frac{1}{n} \sum_{k'=0}^{n-1} \sum_{\tilde{k}_2=0}^{p^\nu-1} \sum_{\tilde{k}_1=0}^{\tilde{k}_2} U_{\tilde{k}_1, -k'}^* (-1)^{\tilde{k}_1+\tilde{k}_2} \begin{bmatrix} \tilde{x} \\ \tilde{k}_2 - \tilde{k}_1 \end{bmatrix} \alpha^{-k'x'} V_{\tilde{k}_2, k'} \quad (18)$$

$$\sum_{y=0}^{N-1} u_y^* v_{x-y} = \frac{1}{n} \sum_{k'=0}^{n-1} \sum_{\tilde{k}_2=0}^{p^\nu-1} \sum_{\tilde{k}_1=0}^{\tilde{k}_2} U_{\tilde{k}_1, k'}^* (-1)^{\tilde{k}_2} \begin{bmatrix} \tilde{x} - \tilde{k}_1 \\ \tilde{k}_2 - \tilde{k}_1 \end{bmatrix} \alpha^{-k'x'} V_{\tilde{k}_2, k'} \quad (19)$$

The proof follows directly from the convolution lemma (Lemma 5).

The first part of Theorem 7 (Equation (18)) implies

**Corollary 8** (Parseval theorem) *Let  $\underline{u}, \underline{U}, \underline{v}$  and  $\underline{V}$  be as in Theorem 7, then*

$$\frac{1}{n} \sum_{k=0}^{N-1} U_{\tilde{k}, -k'}^* V_k = \sum_{x=0}^{N-1} u_x^* v_x.$$

The dual of generalized Reed-Muller codes is immediately obtained from this result (see Section V). The more important consequence of Theorem 7 is, however, its implication on the solution of linear difference equations, which is discussed in the next section. For that discussion, a polynomial representation of the finite field Fourier transforms is useful. In the remaining part of the present section, we shall develop that representation.

Let  $\underline{v}$  denote a vector of length  $N$ , and define

$$v(z) := \sum_{x=0}^{N-1} v_{\tilde{x}n, x}^* z^x. \quad (20)$$

In the case  $\gcd(N, p) = 1$ , this polynomial was used by Mattson and Solomon [7] to express the traditional finite field Fourier transform in the following form

$$V_k = v(z) \Big|_{z=\alpha^k}.$$

A similar expression can also be found in the case of the generalized discrete Fourier transform. It involves the Hasse derivative, which was brought to our attention by Massey. The  $\tilde{k}$ -th Hasse derivative of a polynomial  $u(z)$  of degree  $N - 1$  is defined by [8]

$$D_z^{(\tilde{k})} u(z) := \sum_{x=\tilde{k}}^{N-1} u_x \begin{bmatrix} x \\ \tilde{k} \end{bmatrix} z^{x-\tilde{k}},$$

and can be computed by differential calculus, as long as the reduction modulo  $p$  is made properly:

$$D_z^{(\tilde{k})} u(z) = \frac{1}{\tilde{k}!} \left( \frac{d}{dz} \right)^{\tilde{k}} u(z) \pmod{p}. \quad (21)$$

With this definition, the generalized discrete Fourier transform reads

$$V_k^* = (-z)^{\tilde{k}} D_z^{(\tilde{k})} v(z) \Big|_{z=\alpha^k}. \quad (22)$$

Equation (22) was used by Castagnoli, Massey, Schoeller, and Seemann in their study of repeated root cyclic codes [9]. The new aspect is that the relationship is given by a finite field Fourier transform. The inverse transform can also be indicated in terms of the polynomial  $V(z) = \sum_{k=0}^{N-1} V_{\tilde{k}n, k}^* z^k$ :

$$v_x^* = \frac{1}{n} (-z)^{\tilde{x}} D_z^{(\tilde{x})} V(z) \Big|_{z=\alpha^{-x}}. \quad (23)$$

## IV Solutions of Linear Difference Equations with Constant Coefficients

Linear difference equations with constant coefficients play an important role in the design of stream ciphers. An element  $v_x \in \text{GF}(q)$  from the solution of such an equation is obtained from its predecessors by linear combination:

$$v_x = \sum_{i=1}^d c_i^* v_{x-i}. \quad (24)$$

The coefficients  $c_i^*$  are from  $\text{GF}(q)$ . The star was introduced for compatibility with the definitions from the previous section. With the convention  $c_0^* = 1$ , Equation (24) can also be written in the form

$$\sum_{i=0}^d c_i^* v_{x-i} = \sum_{i=0}^d c_i^* D^i v_x = 0, \quad (25)$$

where  $D$  denotes the delay operator. The polynomial  $c(z) := \sum_{i=0}^d c_i^* z^i$  is called the characteristic polynomial of the recursion. The most general form of such a polynomial is

$$c(z) = \prod_{k' \in \mathcal{K}} (z - \alpha^{k'})^{m_{k'}}. \quad (26)$$

Note that  $\mathcal{K}$  needs to be a union of cyclotomic sets if  $c_i^*$  is to be in  $\text{GF}(q)$ , more precisely: if  $k' \in \mathcal{K}$ , then  $\{k'q^i \pmod{q^\mu - 1}, i \in \{0, 1, 2, \dots, \mu - 1\}\} \subset \mathcal{K}$ . Furthermore, in order to simplify later expressions, we define  $m_{k'} = 0$ ,  $\forall k' \notin \mathcal{K}$ .

The Fourier transform  $\underline{C}^*$  of  $c(z)$  can be expressed in terms of its Hasse derivative

$$C_k^* = (-z)^{\tilde{k}} D_z^{(\tilde{k})} c(z) \Big|_{z=\alpha^{k'}}, \quad k \in \{0, 1, \dots, N-1\}. \quad (27)$$

By Equations (21) and (26), the coefficients have the property

$$C_{\tilde{k}, k'}^* = 0, \quad \forall \tilde{k} < m_{k'}, \quad k' \in \mathcal{K}, \quad (28)$$

and may be nonzero otherwise.

With these preparations, the general solution of the recursion associated with the polynomial  $c(z)$ , as described by Milne-Thomson [10], is easily obtained.

**Theorem 9** (Milne-Thomson [10]) Let  $N$  be the period of the polynomial  $c(z)$ , i.e., let  $N$  be the smallest integer such that  $c(z)|z^N - 1$ , then all sequences  $\underline{v} \in (\text{GF}(q))^N$  generated by  $c(z)$  can be represented in the form

$$v_x = \frac{1}{n} \sum_{k' \in \mathcal{K}} \sum_{\tilde{k}=0}^{m_{k'}-1} V_k (-1)^{\tilde{k}} \begin{bmatrix} \tilde{x} \\ \tilde{k} \end{bmatrix} \alpha^{-k' x'}, \quad (29)$$

with  $V_k \in \text{GF}(q^\mu)$  satisfying the conjugacy constraints (14).

Remember that  $\mu$  was the smallest integer such that  $N|q^\mu - 1$ .

The finite field Fourier transform based proof of Theorem 9 runs as follows: Choose  $\underline{u} = \underline{c}$  in the second statement of Theorem 7, then the linear recursion, i.e., Equation (25), is satisfied whenever  $V_{k',\tilde{k}_2} = 0$ ,  $\forall \tilde{k}_2 \geq \tilde{k}_1$  such that  $C_{k',\tilde{k}_1}^* \neq 0$ , or with Equation (28), whenever  $\tilde{k}_2 \geq m_{k'}$ . This proves that all sequences of the form (29) satisfy the linear recursion. The restriction of  $k$  in (29) allows for  $\sum_{k'} m_{k'}$  different values, which are associated with as many linearly independent functions. Let us assume that each coefficient is arbitrary from  $\text{GF}(q^\mu)$ , then the number of solution matches the number of initial conditions for the recursion. In particular, that set includes the  $\text{GF}(q)$ -valued solutions, we are interested in. According to Equation (13), they are obtained whenever (14) is fulfilled, which completes the proof.

The theorem can be interpreted to state that the sequences generated by the polynomial  $c(z)$  are the sequences with vanishing Fourier coefficients  $V_k$  where the polynomial has non-vanishing coefficients  $C_k^*$ .

Theorem 9 also has an interesting implication for repeated root cyclic codes. Consider the polynomial  $c(z)$  from Equation (26) as being the generating polynomial of a cyclic code, then Theorem 9 implies that the vectors described by Equation (29) are the parity check vectors of that code. Since the theorem also holds when the primes and tilde are removed from  $x$  (reordering of the components of  $\underline{v}$  and  $\underline{V}$ ), the following vectors, with  $0 \leq \tilde{k} \leq m_{k'} - 1$ ,  $m_{k'} \in \mathcal{K}$ , are parity checks for the code

$$\left( \begin{bmatrix} 0 \\ \tilde{k} \end{bmatrix}, \begin{bmatrix} 1 \\ \tilde{k} \end{bmatrix}_{\alpha^{-k'}}, \dots, \begin{bmatrix} N-1 \\ \tilde{k} \end{bmatrix}_{\alpha^{-k'(N-1)}} \right).$$

These vectors are the parity checks originally found by Castagnoli, Massey, Schoeller, and Seemann [9].

The dual of Theorem 9 is obtained in a similar way

**Theorem 10** *If  $\underline{v} \in (\text{GF}(q))^n$  is a sequence of the form*

$$v_x = \frac{1}{n} \sum_{k=0}^{N-1} V_k (-1)^{\tilde{k}} \begin{bmatrix} \tilde{x} \\ \tilde{k} \end{bmatrix}_{\alpha^{-k'x'}},$$

*then it can be generated by the linear recursion associated with the polynomial*

$$c(z) = \prod_{k'=0}^{n-1} (z - \alpha^{k'})^{m_{k'}},$$

*with*

$$m_{k'} = \begin{cases} 0 & \text{if } V_{\tilde{k},k'} = 0 \quad \forall \tilde{k} \\ \{\max \tilde{k} : V_{\tilde{k},k'} \neq 0\} + 1 & \text{otherwise.} \end{cases}$$

*If the sequence is repeated periodically,  $c(z)$  is minimal.*

This result indicates that the characteristic polynomial of a sequence is uniquely and directly determined by its Fourier transform. We see that linear recursions in finite fields lead to a duality between sequences and their generating recursions. This fact by itself is not new, the

understanding that the relationship is through a generalized finite field Fourier transform seems not to have been known, however.

Theorem 10 has important implications like:

**Theorem 11** (Key [11]) *The length  $\mathcal{L}(\underline{v})$  of the shortest linear recursion that can generate the periodic sequence  $\underline{v}$  is given by*

$$\mathcal{L}(\underline{v}) = \sum_{k'} m_{k'}.$$

In the case  $p \nmid N$ , we have  $m_{k'} \in \{0, 1\}$ , and Key's theorem becomes

**Corollary 12** (Blahut's theorem [2]) *In the case  $p \nmid N$ , we have*

$$\mathcal{L}(\underline{v}) = \omega_H(\underline{V}),$$

where  $\omega_H(\underline{V})$  denotes the Hamming weight of  $\underline{V}$ .

Key's theorem has played an important role in the design of stream ciphers. Various examples can be found in Rueppel [12]. Blahut's use of Corollary 12 on the other side was crucial to achieve a deeper understanding of BCH and RS decoding [2].

## V Decoding of Reed-Muller Codes

BCH codes and RS codes have a spectral representation. Binary Reed-Muller codes are also known to have this property if they are obtained from their punctured associated cyclic code, (see e.g. [13]). With the above generalization of the finite field Fourier transform, a more direct spectral representation can be given, however. In particular, this applies to a slight generalization of Reed-Muller codes. Their spectral definition is given by

$$c_x = \sum_{\substack{k=0 \\ \omega_H(k) \leq \nu - \delta}}^{p^\nu - 1} (-1)^k C_k \begin{bmatrix} x \\ k \end{bmatrix}, \quad (30)$$

with  $C_k \in \text{GF}(p)$ . In this expression  $\omega_H(k)$  denotes the number of nonvanishing coefficients in the  $p$ -ary representation of  $k$ . The condition  $\omega_H(k) \leq \nu - \delta$  determines the allowed Fourier components. They are the information symbols. The complement of the condition, i.e.,  $\omega_H(k) > \nu - \delta$  determines the spectral nulls.

One of the most interesting property of the spectral representation of the codes is a new interpretation of Reed decoding. Let the transmitted codeword  $\underline{c}$  be modified by an error vector  $\underline{e} \in (\text{GF}(p))^\nu$  and let the received vector  $\underline{r}$  correspondingly read  $\underline{r} = \underline{c} + \underline{e}$ . We consider the cyclic shifts of that received vector

$$\underline{r}^{(\xi)} = (r_\xi, r_{1+\xi}, \dots, r_{N-1+\xi}),$$

and their Fourier coefficients  $R_k^{(\xi)}$

$$R_k^{(\xi)} = \sum_{x=0}^{p^\nu - 1} r_{x+\xi} (-1)^x \begin{bmatrix} k \\ x \end{bmatrix} \quad (31)$$

for particular values of  $k$  and  $\xi$ . Take  $k$  with  $\omega_H(k) = \nu - \delta$ , then there is a set of  $\delta$  indices  $i_j$  such that  $k_{i_0} = k_{i_1} = \dots = k_{i_{\delta-1}} = 0$ . This induces a decomposition of any number  $\eta \in \{0, 1, \dots, p^\nu - 1\}$  into the following components:

$$\hat{\eta} = \sum_{j=0}^{\delta-1} \eta_{i_j} p^{i_j}, \quad \text{and} \quad \check{\eta} = \eta - \hat{\eta}.$$

For  $k$ , the decomposition reads  $\hat{k} = 0$  and  $\check{k} = k$ . Now consider shifts  $\xi$  such that  $\check{\xi} = 0$ , then Lucas' lemma (Lemma 1), the convolution lemma (Lemma 5), and  $[i]_k^0 = \delta_{i,k}$  imply

$$R_k^{(\xi)} = C_k + \sum_{\check{x}} e_{\check{x}+\hat{\xi}} (-1)^{\check{x}} \begin{bmatrix} k \\ \check{x} \end{bmatrix}. \quad (32)$$

These are  $p^\delta$  estimates of  $C_k$ , each one corrupted by a linear combination of error bits. Since each combination involves different error bits, a majority decision amongst the estimates corrects up to  $\lfloor (p^\delta - 1)/2 \rfloor$  errors.

After completion of the above steps for all  $k$  with  $\omega_H(k) = \nu - \delta$ , the corresponding components can be removed from the received word. The remaining components are associated with  $k$  such that  $\omega_H(k) \leq \nu - \delta - 1$ , and the decoding procedure can be repeated for the new code. The number of errors that can be corrected becomes  $\lfloor (p^{\delta+1} - 1)/2 \rfloor$ . The further iteration is straight forward. It is easy to see that the overall decoding algorithm can decode up to  $\lfloor (p^\delta - 1)/2 \rfloor$  errors. In the case  $p > 2$ , this implies that the minimum distance is at least  $p^\delta$ . Since the codeword with  $C_k = 1$  when  $k = p^{\nu-\delta} - 1$  and 0 otherwise has Hamming weight  $p^\delta$ , we conclude that the minimum distance is exactly  $p^\delta$ .

The decoding rule described above gives a simple interpretation to the construction of the decision variables in the Reed decoding of Reed-Muller codes in terms of an *inverse finite field Fourier transform*.

A similar approach is conceivable for the more general  $p$ -ary Reed-Muller codes introduced by Massey, Costello, and Justesen [14]. The polynomial form in which these codes were defined is equivalent to the following spectral description. Let  $d$  be any number of the form  $\prod_{j=0}^{\nu-1} d^{(i)}$  with  $d^{(i)} \in \{1, 2, \dots, p\}$ , then the  $p$ -ary Reed-Muller codes with distance  $d$  are given by

$$c_x = \sum_{\substack{k=0 \\ \omega_{MCJ}(k) \geq d}}^{p^\nu-1} (-1)^k C_k \begin{bmatrix} x \\ k \end{bmatrix}, \quad (33)$$

with  $C_k \in \text{GF}(p)$ . The relationship of these codes with the previous ones is described by

$$\omega_{MCJ}(k) \begin{cases} = 2^{\nu - \omega_H(k)} & \text{if } p = 2, \\ \geq p^{\nu - \omega_H(k)} & \text{otherwise,} \end{cases}$$

i.e., the codes agree, in the case  $p = 2$ , and the former codes are subcodes of the latter ones in the case  $p > 2$  and  $d = p^\delta$ .

Again, finding  $d$  linearly independent syndromes which satisfy relations similar to Equation (32) is not difficult. Typically, a single error will, however, now corrupt several syndromes. This is the reason why a majority decision decoding can no more be used for the

$p$ -ary Reed-Muller codes. A procedure which decodes these codes up to their minimum distance remains a challenge.

Finally, we note that the  $p$ -ary Reed-Muller codes form a natural class of codes in the sense that the dual of such codes are again  $p$ -ary Reed-Muller codes. Theorem 8 states that

$$\sum_{x=0}^{p^\nu-1} h_x^* c_x = \sum_{k: \omega_{MCJ}(k) \geq d} H_k^* C_k.$$

Therefore, using Equation (16), the dual code is defined by  $H_k = 0$ , for all  $k : \omega_{MCJ}(p^\nu - 1 - k) < d$ , i.e., is a  $p$ -ary Reed-Muller code of minimum distance  $d^* = \prod_{i=0}^{\nu-1} (p + 1 - d^{(i)})$  ( $d = \prod_{i=0}^{\nu-1} (d^{(i)})$ ).

## VI Conclusion

The finite field Fourier transform has been very successful in coding and cryptography. So far it was restricted to vectors of a length  $N$ , with  $N$  relatively prime to the characteristic of the field  $p$ . In the present paper, we have exposed one possible way of generalizing this transform to vectors of arbitrary length. Many known results have a simpler interpretation in terms of this generalized finite field Fourier transform. Two of them have been considered: the solution of linear recursions and the decoding of Reed-Muller codes. Many other results can also be obtained in this framework, even unexpected ones, like the algebraic normal form of Boolean functions. Finally, we note that the generalization is not restricted to cyclic vectors ( $v_{x+N} = v_x$ ) but could also be developed for classes of constacyclic codes ( $v_{x+N} = \xi^n v_x$ ,  $\xi \in \text{GF}(q)$ ). It is our hope, that the concept will find some interest and will be further developed.

## Acknowledgement

I would like to thank Dr. Walter Schneider from ABB Corporate Research, Baden, Switzerland, for an alternative proof of Theorem 6 which was used in earlier versions and presented in Bellagio in 1987, and I would like to thank Dr. Richard Blahut from IBM Corporation, Owego, USA, for his interest and for helpful comments and suggestions during the preparation of the paper.

## A Proof of Corollary 2

Let  $\nu = 1$ , assume that  $\xi \geq \eta$  (otherwise both sides are zero), and assume that  $\xi \geq p - \eta$ , then

$$\begin{aligned} \begin{bmatrix} \xi \\ \eta \end{bmatrix} &= \frac{\xi(\xi-1)\dots(p-\eta)}{\eta(\eta-1)\dots(p-\xi)} \begin{bmatrix} p-1-\eta \\ p-1-\xi \end{bmatrix} \\ &= \frac{\xi}{p-\xi} \frac{\xi-1}{p-\xi+1} \dots \frac{p-\eta+1}{\eta-1} \frac{p-\eta}{\eta} \begin{bmatrix} p-1-\eta \\ p-1-\xi \end{bmatrix} \\ &= (-1)^{\xi+\eta-p} \begin{bmatrix} p-1-\eta \\ p-1-\xi \end{bmatrix} \end{aligned}$$

$$= (-1)^{\xi+\eta} \begin{bmatrix} p-1-\eta \\ p-1-\xi \end{bmatrix} \pmod{p}.$$

In the case  $\xi \leq p - \eta$ , the proof can be started from the other side. Next, let  $\nu$  be arbitrary, then we use

$$\begin{aligned} \xi &= \sum_{i=0}^{\nu-1} \xi_i p^i \\ p^\nu - 1 &= \sum_{i=0}^{\nu-1} (p-1) p^i \\ p^\nu - 1 - \xi &= \sum_{i=0}^{\nu-1} (p-1 - \xi_i) p^i, \end{aligned}$$

as well as corresponding expressions for  $\eta$  and Lucas' lemma (lemma 1) to obtain:

$$\begin{bmatrix} p^\nu - 1 - \eta \\ p^\nu - 1 - \xi \end{bmatrix} = \prod_{i=0}^{\nu-1} (-1)^{\xi_i + \eta_i} \begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix} \pmod{p}.$$

Again with Lucas' lemma and with  $(-1)^{p^i} = -1$ , this implies the assertion.

## B Proof of Corollary 3

Define  $\mu = \lceil \log_p \eta \rceil$ , then  $\eta = \eta_{\mu-1} p^{\mu-1} + \sum_{i=0}^{\mu-2} \eta_i p^i$ , with  $\eta_{\mu-1} \neq 0$ . Using Equation (2) and Lemma 1, this implies

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi_{\mu-1} \\ \eta_{\mu-1} \end{bmatrix} \prod_{i=0}^{\mu-2} \begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix}.$$

Therefore,  $p^\mu$  is a period but  $p^{\mu-1}$  is no more a period, which proves the lemma.

## C Proof of Lemma 5

Define the delay operators  $D : v_x \rightarrow v_{x-1}$  and  $E : v_x \rightarrow v_{x+1}$ , with  $x \pm 1$  being taken modulo  $p^\nu$  (on the boundary). Note that this condition is automatically met for the binomial coefficients in  $\text{GF}(p)$  (see Lemma 3). Now consider the first expression:

$$\begin{aligned} \sum_{y=0}^{p^\nu-1} (-1)^y \begin{bmatrix} \kappa \\ \eta \end{bmatrix} \begin{bmatrix} \xi + \eta \\ \lambda \end{bmatrix} &= \sum_{\eta=0}^{\kappa} \binom{\kappa}{\eta} (-E)^\eta \begin{bmatrix} \xi \\ \lambda \end{bmatrix} \pmod{p} \\ &= (1 - E)^\kappa \begin{bmatrix} \xi \\ \lambda \end{bmatrix} \pmod{p}. \end{aligned}$$

Since Equation (1) and (2) imply

$$(1 - E) \begin{bmatrix} \xi \\ \lambda \end{bmatrix} = \begin{cases} -[\lambda]_{\kappa-1}^{\xi} & \text{if } \lambda \geq 1 \\ 0 & \text{if } \lambda = 0, \end{cases}$$

this completes the proof of the first statement. The second statement is obtained similarly, but using

$$(1 - D) \begin{bmatrix} \xi \\ \lambda \end{bmatrix} = \begin{cases} \begin{bmatrix} \xi^{-1} \\ \lambda^{-1} \end{bmatrix} & \text{if } \lambda \geq 1 \\ 0 & \text{if } \lambda = 0. \end{cases}$$

## References

- [1] R.E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley Publishing Company, Inc., 1983.
- [2] R.E. Blahut, "Transform techniques for error control," *IBM J. Res. and Dev.*, Vol. 23, pp. 299-315, May 1979.
- [3] P. Mathys, "A generalization of the discrete Fourier transform in finite fields," *Proc. IEEE Symp. Inform. Theory*, San Diego (CA), Jan. 14-19, 1990.
- [4] S.R. Blackburn, "A Generalization of the Discrete Fourier Transform: Determining the Minimal Polynomial of a Periodic Sequence," *IEEE Trans. Inform. Theory*, (to appear).
- [5] C.G. Günther, "Fourier transform in cryptography and coding," *IEEE Workshop on Inform. Theory*, Bellagio, Italy, June 1987.
- [6] E. Lucas, "Théorie des fonctions numériques simplement périodiques", *American Journal of Mathematics*, vol. 1, pp. 184-321, 1878
- [7] H.F. Mattson and G. Solomon, "A new treatment of Bose Chaudhuri codes," *J. Soc. Indus. Appl. Math.*, Vol. 9, pp. 654-659, 1961.
- [8] H. Hasse, "Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenem Konstantenkörper bei beliebiger Charakteristik," *J. reine angew. Math.*, Bd. 175, S. 50-54, 1936.
- [9] G. Castagnoli, J.L. Massey, P.A. Schoeller, and N. Seemann, "On repeated-root cyclic codes," *IEEE Trans. on Inform. Theory*, vol. IT-37, pp. 337-342, 1991.
- [10] L.M. Milne-Thomson, *The Calculus of Finite Differences*, MacMillan and Co., London, 1951.
- [11] E.L. Key, "An analysis of the structure and complexity of nonlinear sequence generators," *IEEE Trans. on Inform. Theory*, vol. IT-22, pp. 732-736, 1976.
- [12] R.A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag, Berlin, Heidelberg, 1986.
- [13] W.W. Peterson and E.J. Weldon, Jr., *Error Correcting Codes*, MIT Press, Cambridge, MA, and London, 1972.
- [14] J.L. Massey, D.J. Costello Jr. and J. Justesen, "Polynomial weights and code constructions," *IEEE Trans. on Inform. Theory*, vol. IT-19, pp. 101-110, 1973.

# Soft is Better Than Hard

Joachim Hagenauer  
Technical University of Munich  
D-80290 Muenchen

## **Abstract**

The well-known argument that soft decisions should be used as input values of a decoder is extended to “soft outputs”. For binary variables “soft” values are defined as log-likelihood ratios. The “soft-output”–Viterbi algorithm (SOVA) is described in a compact way, as well as a general rule for ”soft-in/soft-out” decoding of binary block codes. An example with a Reed–Muller code in a concatenated scheme shows a coding gain of 3.5 dB at a BER of  $10^{-5}$ , where only trivial repetition codes and single parity check codes are used as building blocks.

## I Introduction

It is well known from information theory arguments that soft decisions on an AWGN channel give a better performance of channel coding systems. For a binary input channel and with a code rate of 1/2 the absolute Shannon limit is at an  $E_b/N_0$  of 0.1 dB for soft decisions and at 1.9 dB for hard decisions. With the more realistic so-called  $R_0$  criterion promoted by Massey in [2] the respective numbers are 2.4 dB and 4.5 dB. This makes it quite clear that at least at the AWGN – and the same is true for many other channels like the fading channel – soft decisions should be used. In his 1984 Zurich Seminar paper [3] Massey has emphasized that the demodulator should make no decisions at all, rather deliver relative likelihoods about the received variables. Unfortunately contrary to convolutional codes where soft decoding became very popular through the use of the Viterbi algorithm, soft decision decoding for block codes is in general not a simple task and for powerful block codes like Reed–Solomon codes, soft decision algorithms are not readily available. Massey considered this problem already in 1962 in his doctoral thesis published in [1] where he showed that for the class of L-orthogonal block codes maximum *a posteriori* “soft” decoding is easily possible.

Using these facts and ideas we will argue here that the decoder should not only accept soft inputs from the channel but also deliver soft outputs, i.e. likelihood ratios. These soft, analog values can be utilized by an outer decoder of a concatenated scheme or by the source decoder. Furthermore, very powerful codes can be constructed from simple codes by interleaving, cascading, concatenation or by forming products. Now, with soft-in/soft-out decoding algorithms being available, these codes can be decoded step by step in a multistage fashion and achieve almost optimum performance with low decoder complexity.

In this case we benefit from the decoding techniques used for decoding multilevel coded modulation.

The general philosophy in this paper is: Never make a hard decision, only deliver probabilities or log-likelihood values of possible decisions to the next stage (level, decoder, sink, or authority). We restrict ourselves here to binary decisions, represented by the sign of a real random variable, called the soft value. The magnitude of this real number represents the reliability of the decision. Viewed this way, a decoder is like a filter which accepts analog values and delivers analog values hopefully with a smaller variance.

## II Soft Values of Source, Channel Bits and Decoded Bits

### Likelihood Algebra of a Binary Random Variable

We restrict ourselves to binary data and formulate the concept of the log-likelihood ratio  $L(u)$  of a binary random variable  $u$ . Let  $u$  be in GF(2) with the elements  $\{+1, -1\}$  with  $+1$  as the “null” element under the addition  $\oplus$  and define  $L(u)$  as the real number

$$L(u) = \log \frac{P(u = +1)}{P(u = -1)}. \quad (1)$$

Unless otherwise stated the logarithm is the natural logarithm. We will subsequently call  $L(u)$  the “soft”-value or L-value of a binary random variable. The sign of  $L(u)$  corresponds to the hard decision, and the magnitude  $|L(u)|$  is the reliability of this decision. Note that given  $L(u)$  the probability of making a correct hard decision is

$$P(\text{correct}|L(u)) = \frac{e^{|L(u)|}}{1 + e^{|L(u)|}}. \quad (2)$$

Sometimes the probability is conditioned on some observations  $z_i$ . Then we have the conditioned log-likelihood ratio

$$L(u|z_i) = \log \frac{P(u = +1|z_i)}{P(u = -1|z_i)}. \quad (3)$$

It is easy to show that for statistically independent  $u$ :

$$\begin{aligned} L(u_1 \oplus u_2) &= \log \frac{1 + e^{L(u_1)} e^{L(u_2)}}{e^{L(u_1)} + e^{L(u_2)}} \\ &\approx \text{sign}(L(u_1)) \text{sign}(L(u_2)) \min(|L(u_1)|, |L(u_2)|). \end{aligned} \quad (4)$$

For the log-likelihood or algebraic values  $L(u)$  – the “soft values” – we can define a special algebra where  $\boxplus$  denotes the addition for this set

$$\begin{aligned} L(u_1) \boxplus L(u_2) &= L(u_1 \oplus u_2), \\ L(u) \boxplus \infty &= L(u) \\ \text{and } L(u) \boxplus 0 &= 0. \end{aligned} \quad (5)$$

Note that the addition of the soft values has the “null” element  $\infty$ , and the commutative and associative law are valid. However, in general no inverse (minus) element exists because two unreliable elements cannot add up to the reliable element  $\infty$ . By complete induction one can further prove that

$$\begin{aligned} L\left(\sum_{\substack{j=1 \\ \oplus}}^J u_j\right) &= \sum_{j=1}^J L(u_j) = \log \frac{\prod_{j=1}^J (e^{L(u_j)} + 1) + \prod_{j=1}^J (e^{L(u_j)} - 1)}{\prod_{j=1}^J (e^{L(u_j)} + 1) - \prod_{j=1}^J (e^{L(u_j)} - 1)} \\ &= \log \frac{1 + \prod_{j=1}^J \tanh(L(u_j)/2)}{1 - \prod_{j=1}^J \tanh(L(u_j)/2)} \end{aligned} \quad (6)$$

and finally approximate it as in Equation (4) by

$$L\left(\sum_{\substack{j=1 \\ \oplus}}^J u_j\right) = \sum_{j=1}^J L(u_j) \approx \prod_{j=1}^J \text{sign}(L(u_j)) \cdot \min_{j=1, \dots, J} |L(u_j)|. \quad (7)$$

The reliability of the sum therefore is determined by the smallest reliability of the terms. The  $\boxplus$  algebra can be realized as shown in Figure 1.

## Soft Channel Outputs

In this section we deal with the channel, including the demodulator and the matched filter. We will define more clearly what is meant by the “soft output” of the channel: If we encode the binary values  $u$  having a soft values  $L(u)$  then we obtain coded bits  $x$  with soft values  $L(x)$ . After transmission over a binary symmetric channel (BSC) or a Gaussian/Fading channel we can calculate the log-likelihood ratio (soft value) of  $x$  conditioned on the matched filter output  $y$  and – if applicable – the fading value  $a$ :

$$\begin{aligned} L(x|y) &= \log \frac{P(x = +1|y)}{P(x = -1|y)} = \log \left( \frac{p(y|x = +1)P(x = +1)}{p(y|x = -1)P(x = -1)} \right) \\ &= L(y|x) + L(x) = L_c y + L(x). \end{aligned} \quad (8)$$

The channel reliability  $L_c$  is

$$L_c = L(e) = \log \frac{1 - P_0}{P_0} \quad (9)$$

for the binary symmetric channel where errors  $e$  occur with crossover probability  $P_0$  and where  $y$  is binary. For the fully interleaved fading channel, where the fading amplitude  $a$  is used as channel state information we obtain

$$L_c = 4a \frac{E_s}{N_0}. \quad (10)$$

For the Gaussian channel we set  $a$  to 1. The quantity

$$L(y|x) = \log \frac{p(y|x = +1)}{p(y|x = -1)} = L_c y = L(x|y) - L(x) \quad (11)$$

is what has been transmitted over the channel, namely the difference between the *a posteriori* and the *a priori* soft value of  $x$ . We call it the “*soft output*” of the channel and use it as “*soft input*” for the first decoder stage. It is not difficult to show that in  $L$  different, statistically independent transmissions as in a diversity system

$$L(x|\underline{y}) - L(x) = \sum_{i=1}^L L(x_i|y_i) - L(x). \quad (12)$$

### III Soft-In/Soft-Out Decoding for Binary Trellis Codes

If we wish to obtain reliability values for decoded bits of a binary trellis code we could use the maximum symbol-by-symbol decoding method derived by Bahl et al. [4]. They use the maximum *a posteriori* probability to decide on a symbol. Having calculated this probability we can use it as well in Equation (1) to obtain the soft output. However the method is quite complex.

Instead we wish to modify the Viterbi algorithm to accept in addition to soft inputs *a priori* information about the source bits and to deliver soft outputs. Forney [9] considered already to use also *a priori* information about the source bits  $u_k$  in the metric of the Viterbi algorithm. With the notation introduced above we can show [16] that the metric is then

$$M_k^{(m)} = M_{k-1}^{(m)} + \sum_{n=1}^N x_{k,n}^{(m)} L_{c_{k,n}} y_{k,n} + u_k^{(m)} L(u_k). \quad (13)$$

$x_{k,n}^{(m)}$  and  $u_k^{(m)}$  are the code bits, respectively the information bits on the  $m$ th path at time  $k$ . It is further shown in [16] that the probability of the path  $m$  at time  $k$  and the metric are related by

$$P(\text{path } m) = e^{M_k^{(m)}/2}. \quad (14)$$

This slight modification of the metric of the VA in Equation (13) incorporates the *a priori* or *a posteriori* information about the probability of the source bits. In Figure 2 it is depicted how the VA will use the soft value  $L(u_k)$ . If the channel is very good,  $L_{c_{k,n}}$  will be larger than  $|L(u_k)|$  and decoding relies on the received channel values. If the channel is bad, as during a deep fade, decoding relies on the *a priori* information  $L(u)$ . Both soft inputs are used in a balanced way. This is similar as in a Kalman filter where a prediction value is updated by the measurements.

The so-called *Soft-Output-VA* (SOVA) was introduced in [10] and it will now be described in a different way using the  $L(u)$  algebra. Other related proposals have been made in [4] and [8].

The VA proceeds in the usual way by calculating the metrics for the  $m$ th path using Equation (13) with or without  $L(u)$ . For each state it selects the path with the larger metric  $M_j^{(m)}$ . Figure 3 shows an example trellis where the VA has selected at time  $j$  the

ML-path (dotted line) with index  $m_0$  and has discarded the other path  $m'_0$  ending at this state. Let us define the difference

$$\Delta_j^0 = \frac{1}{2} (M_j^{(m_0)} - M_j^{(m'_0)}). \quad (15)$$

The probability that the path decision was correct at this point is from Equation (14)

$$P(\text{correct}) = \frac{P(\text{path } m_0)}{P(\text{path } m_0) + P(\text{path } m'_0)} = \frac{e^{M_j^{(m_0)}/2}}{e^{M_j^{(m_0)}/2} + e^{M_j^{(m'_0)}/2}} = \frac{e^{\Delta_j^0}}{1 + e^{\Delta_j^0}}. \quad (16)$$

Therefore with Equation (2) the likelihood ratio or “soft” value of this binary path decision is the positive quantity  $\Delta_j^0$ . Along the ML-path  $m_0$   $\delta + 1$  nonsurviving paths with indices  $l = 0, \dots, \delta$  have been discarded. The difference of their metrics is  $\Delta_j^l \geq 0$ . If the bit  $u_{j-\delta}^l$  on the discarded path equals the decided bit  $\hat{u}_{j-\delta}$ , we certainly would have made no bit error if we would have selected the discarded path. Thus the reliability of this bit decision is  $\infty$ . Otherwise, if the bits differ the log-likelihood value of a bit error  $e_{j-\delta}^l$  equals  $\Delta_j^l$ . Consequently we have

$$L(e_{j-\delta}^l) = \log \frac{P(e_{j-\delta}^l = +1)}{P(e_{j-\delta}^l = -1)} = \begin{cases} \infty & u_{j-\delta}^l = \hat{u}_{j-\delta} \\ \Delta_j^l & u_{j-\delta}^l \neq \hat{u}_{j-\delta} \end{cases}. \quad (17)$$

The total error resulting from all possible discarded paths for bit  $\hat{u}_{j-\delta}$  is

$$e_{j-\delta} = \sum_{\substack{l=0 \\ \oplus}}^{\delta} e_{j-\delta}^l. \quad (18)$$

If the  $\Delta_j^l$  and the  $e_j^l$  are statistically independent with respect to the indices  $l$  — which is approximately true for reasonable codes — then the log-likelihood ratio of the decisions, the *Soft-Output of the VA* (SOVA), is the decision  $\hat{u}_{j-\delta}$  times the L-value of the errors:

$$L(\hat{u}_{j-\delta}) = \hat{u}_{j-\delta} \sum_{\substack{l=0 \\ \oplus}}^{\delta} L(e_{j-\delta}^l) = \hat{u}_{j-\delta} \sum_{\substack{l=0 \\ \oplus}}^{\delta} \Delta_j^l \approx \hat{u}_{j-\delta} \cdot \min_{l=0, \dots, \delta} \Delta_j^l. \quad (19)$$

It is clear from the relation (4) that the sum and the minimum in Equation (19) has to be taken only over those indices  $l$  where the bits differ. We therefore have the same hard decisions as the classical VA and the reliability of these decisions is obtained by taking the minimum of the metric differences along the ML-path, whenever the update sequences  $e_{j-\delta}^l$  indicate this. Typically only two to three updates are necessary. If we are interested in the reliability of the encoded bits  $L(\hat{x})$  instead of  $L(\hat{u})$  the elements of the update sequence are vectors of length  $N$  and at the decision the bit elements of the vector  $\hat{x}_{j-\delta}$  have the reliability  $\min \Delta_{j-\delta}^l$  where the minimum is taken according to the relevant update sequence. In the trace-back implementation the SOVA proceeds as the classical VA. For each state it stores also metric differences and renews the update sequences  $e_{j-\delta}^l$

by modulo two addition of two words. For the decision and its reliability a trace-back is performed as described above.

Simulations have shown that the SOVA performs almost as well as the optimum MAP algorithm and is much less complex.

With this soft-in/soft-out feature the decoder acts like a digital filter: The noisy input samples are transformed into — hopefully less noisy — output samples. If we decode to  $L(\hat{u})$  instead to  $L(\hat{x})$  our filter decimates by a factor which is the inverse of the code rate. The filter is nonlinear since the VA algorithm performs nonlinear operations. Therefore the output of our filter after a Gaussian channel is non gaussian, albeit only slightly. As shown in Figure 4 we can measure the output SNR of the filter

$$\text{SNR} = \frac{\langle u \cdot L(\hat{u}) \rangle^2}{\langle (u \cdot L(\hat{u}) - \langle u \cdot L(\hat{u}) \rangle)^2 \rangle}. \quad (20)$$

We have now a modified Viterbi algorithm available which accepts log-likelihood (soft) L-values from the source and the channel and delivers such values for the output bits. These L-values of the bits are directly related to their probabilities through the inverse of Equation (1) and are thus well-defined quantities. It should be noted that, contrary to the metrics in the classical VA, the L-values should not be multiplied by an arbitrary constant because by Equation (1) this would transform the probabilities into powers of probabilities.

## IV Soft-In/Soft-Out Decoding for Linear Binary Block Codes

Maximum likelihood — or MAP — decoding of block codes with a simple algorithm as simple as the Viterbi algorithm (VA) is still an open problem. Of course, with a block code allowing a trellis representation with a reasonable number of states, the VA or the MAP i.e. Bahl [4] algorithm could be used. Soft outputs are then obtained via the MAP algorithm, respectively the SOVA if the trellis is binary. Soft decoding of L-step orthogonizable codes has already been performed by Massey [1] as an extension of his threshold decoding algorithm in 1963. The optimum symbol-by-symbol decoding rule has been described for linear codes by Hartmann and Rudolph [5] and by Battail [7]. We describe the soft output of these algorithms with our notation although they are quite complex in the general case.

Let  $\underline{x}$  be the codeword of a systematic binary linear code  $C$  over the binary alphabet  $\{+1, -1\}$  with parameters  $(n, k, d)$  and  $C'$  its dual code. The information bits are equally likely. The bit  $x_j$  is transmitted and received as soft decision  $y_j$ , whereas  $y_{Hj} = \text{sign}(y_j)$  is the hard decision. The soft input  $L(y_j|x_j)$  to be used by the decoder is  $L(y_{Hj}) = L_{cj}y_j$ . The decision rule in [5] is similar to Massey's threshold decoding rule for soft inputs: Calculate

$$f_d = \sum_{i=1}^{2^n - k} \prod_{j \in \{I_i\}} \frac{1 - e^{-L(y_{Hj})}}{1 + e^{-L(y_{Hj})}} \quad (21)$$

and decode  $x_m = -1$  if  $f_d < 0$ , otherwise decode to  $+1$ . The index set  $\{I_i\}$  contains those  $j$  for which  $c'_{ij} \cdot \delta_{jm} = -1$ , i.e. is different from the “null” element  $+1$ .  $\delta_{jm}$  is  $-1$  if  $j = m$ ,

otherwise it is +1. As an example we show  $c'_{ij} \cdot \delta_{jm}$  for the first bit ( $m = 1$ ) of a single parity check code (5,4,2) with the (5,1,5) repetition code as its dual code

$$\begin{array}{ccccc} -1 & +1 & +1 & +1 & +1 \\ +1 & -1 & -1 & -1 & -1 \end{array}. \quad (22)$$

Another example is the (3,1,3) repetition code which has the associated set

$$\begin{array}{ccc} -1 & +1 & +1 \\ +1 & -1 & +1 \\ +1 & +1 & -1 \\ -1 & -1 & -1 \end{array}. \quad (23)$$

Clark and Cain have modified this approach and show [6, p. 157] using an equation by Massey [1] that  $f_d$  is equivalent to

$$f_d = \sum_{i=1}^{2^n-k} B_i (1 - 2 \cdot P(B_i \text{ in error})) \quad (24)$$

where

$$B_i = \sum_{\substack{j \in \oplus \\ j \in \{I_i\}}} y_{Hj}. \quad (25)$$

Since the  $y_{Hj}$  are statistically independent we obtain for the L-values of (25)

$$L(B_i | \underline{y}) = \log \frac{P(B_i = +1 | \underline{y})}{P(B_i = -1 | \underline{y})} = \sum_{\substack{\boxplus \\ j \in \{I_i\}}} L(y_{Hj}) \quad (26)$$

and finally

$$f_d = \sum_{i=1}^{2^n-k} L(B_i). \quad (27)$$

This describes the hard output rule for decision. The same rule has been derived by Kolesnik [14] for threshold decodable codes. Unfortunately the  $P(f_d < 0)$  and thus the soft output  $L(x_m)$  cannot be easily obtained from this simplified rule. Kolesnik gives some approximations for the error probability for this decision and thus indirectly for the soft output.

For the exact solution we would have to start again from the beginning of the derivation in [5] where we obtain

$$L(\hat{x}_m) = L(x_m | \underline{y}) = \log \frac{1 + \beta}{1 - \beta} \quad (28)$$

with

$$\beta = \frac{\sum_{i=1}^{2^n-k} \prod_{j=1}^n (1 + c'_{ij} \cdot \delta_{mj} e^{-L(y_{Hj})})}{\sum_{i=1}^{2^n-k} \prod_{j=1}^n (1 + c'_{ij} \cdot e^{-L(y_{Hj})})}. \quad (29)$$

Similar formulas are given by Battail in [7]. Simplifications are obtained for single parity-check codes  $(n, n - 1, 2)$

$$L(\hat{x}_m) = L(y_{Hm}) + \sum_{\substack{\boxplus \\ j=1, j \neq m}}^n L(y_{Hj}) \quad (30)$$

and for the  $(n, 1, n)$  repetition codes as to be expected

$$L(\hat{x}_m) = \sum_{j=1}^n L(y_{Hj}). \quad (31)$$

Suboptimal solutions are possible by using a limited number of quasi orthogonal codewords in  $C'$  and applying iterative decoding.

Elke Offer [13] has rederived the results of Hartmann/Rudolph and Battail for the case of information bits with  $L(u) \neq 0$ , i.e. unequally likely codewords. In this case  $L(u)$  has to be added to  $L(y_{Hj})$  for  $j \leq k$  in Equations (28) to (31).

Equations (28) to (31) provide the soft-in/soft-out transformation of the received codeword. Therefore this decoder is a (in general nonlinear) filter which improves the SNR of the input values  $y_j$  or  $L(y_{Hj}) = L_c y_j$  to the SNR of the output values  $L(\hat{x}_m)$ . So far we have only found that the trivial codes  $(n, 1, n)$  give a linear transformation.

## V Applications

### Decoding of Reed–Muller (RM) Codes in Concatenated Systems

It is well known that RM codes can be constructed by concatenation of smaller codes

$$\text{RM}(r, m) = \{|u|w = u \oplus v| : u \in \text{RM}(r, m - 1), v \in \text{RM}(r - 1, m - 1)\}. \quad (32)$$

We follow the decoding procedure given by Schnabl and Bossert [12], who break down decoding of RM codes to the decoding of single parity check codes (SPC) and repetition codes. However we use the soft output formulas (30) and (31) together with interleaving in a concatenated system. We show the procedure in an example as illustrated in Figure 5 where an outer  $(8,7,2)$  SPC code and an inner  $(8,4,4)$  RM(1,3) code are used. After encoding the outer SPC code the  $w_{ij}$  are formed by combining bits of different codewords  $w_{ij} = u_{ij} \oplus v_{ij}$  and transmitted together with the  $u_{ij}$ . The decoder obtains first  $L(v_{ij'}) = L(u_{ij} \oplus w_{ij}) = L(u_{ij}) \boxplus L(w_{ij})$  and – after rearranging the  $v$  soft values – decodes the repetition code via Equation (31). The repeated results  $L(\hat{v}_{ij'})$  are rearranged back and these statistically independent soft values are added to the soft received values  $L(w_{ij})$  via Equation (5). They constitute together with the received soft  $u$ -values another almost independent estimate for the  $u$  bits. Therefore the expression

$$L(u_{ij}) + (L(w_{ij}) \boxplus L(\hat{v}_{ij})) \quad (33)$$

gives the soft values for decoding the vertical SPC with Equation (30) as part of the RM code. Finally the outer SPC code is decoded with the same Equation (30) to give a soft

output which might be utilized by the source decoder or another outer code. Of course, we can use in all these operations the simple but good approximation of the  $\boxplus$  algebra given in Equation (4). Figure 6 shows the simulation result for this code. Thus, using only trivial codes, modulo 2 additions, additions of real numbers and minimum operations we can achieve a coding gain of 4 dB at a BER of  $10^{-6}$ .

## Decoding of Product Codes

Using the above discussed principles of soft-in/soft-out decoding Lodge et al. [15] have decoded iteratively multidimensional product codes built up from simple codes. As an example a three-dimensional product code built from (16,11) extended Hamming code achieves a BER of  $10^{-4}$  at an  $E_b/N_0$  of 1.3 dB after six cycles of iterative decoding. With a soft output VA we can also perform cascaded Viterbi [11] decoding which was not very effective as long as the outer VA received only hard decisions from the inner decoder.

## Joint Source and Channel Decoding

Source and channel coding have been treated separately in most cases. It can be observed that most source coding algorithms for voice, audio, and images still have correlation in certain bits. Transmission errors in these bits usually accounts for the significant errors in the reconstructed source signal. Therefore concealing algorithms are very popular in many applications. They can use soft output information from the channel decoder for interpolation of samples to conceal the errors left by the channel decoder in the reconstructed analog source signal. It is much better to avoid these errors by supplying the channel decoder with soft information about the correlated source bits it is about to decode. In a framed transmission system such an *a priori* soft information can be easily obtained from the soft output of the previous frame and a soft bit correlation estimator which uses a simple first-order Markov model. The use of the soft output avoids error propagation. In such a source controlled channel decoder [16] soft decisions, soft channel state information and soft source information is combined for better performance. In [16] applications are given for PCM transmission and the full rate GSM speech codec. For a PCM coded oversampled bandlimited Gaussian source transmitted over Gaussian and Rayleigh channels with convolutional codes the decoding errors are reduced by a factor of 4 to 5 when this method is used instead of the VA with hard output. This results in much less signal distortion. The GSM channel decoder can be modified for those few significant bits only which still have correlation between consecutive 20 msec speech frames. With these receiver-only modifications the channel SNR in a bad mobile environment can be lowered by 2 to 3 dB while still maintaining the same voice quality.

## VI Conclusions

We have shown that decoding of binary trellis and block codes should be performed with soft-in/soft-out algorithms. In a receiver chain, in concatenated and in iterated decoding schemes only soft values should be passed between the different stages. Jim Massey

throughout his professional life has always advocated the use of soft decisions. Guided by his arguments and supported by some examples we propose that soft decisions should not only be used as inputs, but demodulators, equalizers, decoders should also produce soft decisions and pass them on to the next stage.

## VII Acknowledgement

The author would like to thank Dipl.-Ings. Elke Offer and Thomas Woerz for stimulating discussions on Section 4.

## References

- [1] J.L. Massey, *Threshold Decoding*, Cambridge, Ma, M.I.T. Press, 1963.
- [2] J.L. Massey, “Coding and modulation in digital communications”, Proc. of the 1980, Zurich Seminar on Digital Communications, IEEE Cat. CH 1980, pp. E.2.1 - E.2.3, 1980.
- [3] J.L. Massey, “The how and why of channel coding”, Proc. of the 1984 Zurich Seminar on Digital Communications, IEEE Cat. No. 84 CH 1998-4, pp. 67-73, 1984.
- [4] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate”, *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, 1974.
- [5] C.R.P. Hartmann and L. D. Rudolph, “An optimum symbol-by-symbol decoding rule for linear codes”, *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 514-517, 1976.
- [6] G.C.Clark, Jr. and J.B. Cain, *Error-Correction Coding for Digital Communications*, Plenum Press, 1982.
- [7] G. Battail, M. C. Decouvelaere, and P. Godlewski, “Replication decoding”, *IEEE Trans. on Inform. Theory*, vol. IT-25, No. 3, pp. 332-345, 1979.
- [8] G. Battail, “Building long codes by combination of simple ones, thanks to weighted-output decoding”, in *Proc. URSI ISSSE*, Erlangen, Germany, pp. 634-637, 1989.
- [9] G.D. Forney, “The Viterbi algorithm”, *Proc. IEEE*, vol. 61, pp. 268-278, 1973.
- [10] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications”, *Proc. GLOBECOM '89*, Dallas, Texas, pp. 47.1.1-47.1.7, 1989.
- [11] J. Hagenauer and P. Hoeher, “Concatenated Viterbi-decoding”, *Proceedings of the 4. Joint Swedish-Soviet Int. Workshop on Information Theory*, Gotland, Sweden, Studentliteratur Lund, 27. Aug. - 1. Sept. 1989.

- [12] G. Schnabl and M. Bossert, “Soft decision decoding of Reed–Muller codes as generalized multiple concatenated codes”, submitted to *IEEE Trans. on Inform. Theory*.
- [13] E. Offer, “Verbesserungen verketteter Codiersysteme durch Verwendung von Qualitatsinformation bei der Decodierung”, Ph.D. thesis in submission to Techn. Univ. Muenchen.
- [14] V. Kolesnik, “Probabilistic decoding of majority codes”, *Problemy Peredachi Informatsii*, Vol. 7, No. 3, pp.3–12, July–Sept., 1971.
- [15] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, “Separable MAP filters for the decoding of product and concatenated coding”, *Proc. of the IEEE Int. Conf. on Comm.*, Geneva, pp. 1740–1745, 1993.
- [16] J. Hagenauer, “Source controlled channel decoding”, accepted for *IEEE Transactions on Communications*.

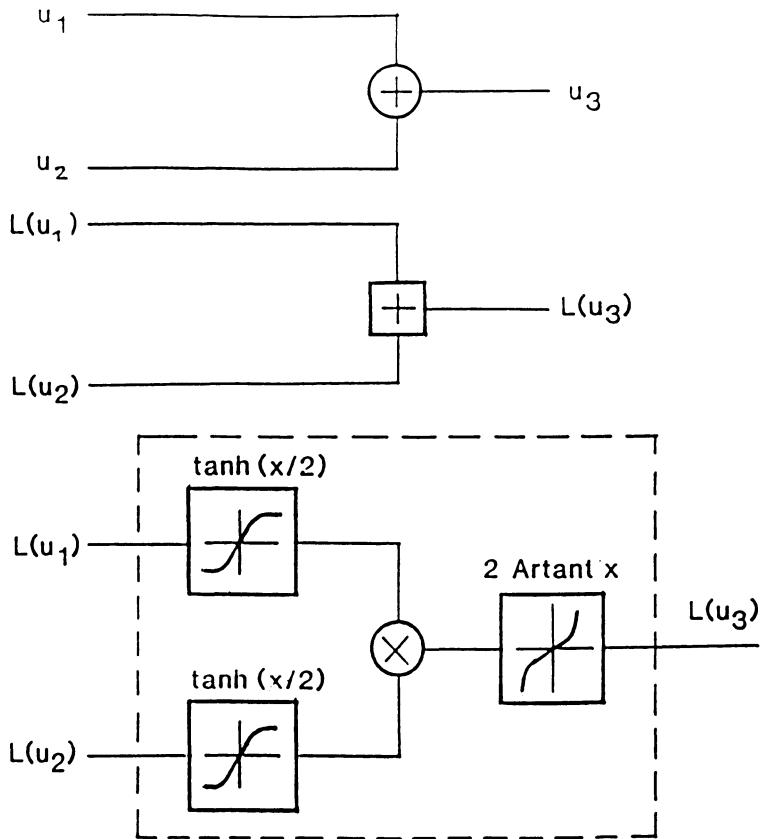


Fig. 1: Nonlinear memoryless circuit for the realization of the addition of log-likelihood ratios ("soft"-values) of two binary random variable  $u_1$  and  $u_2$

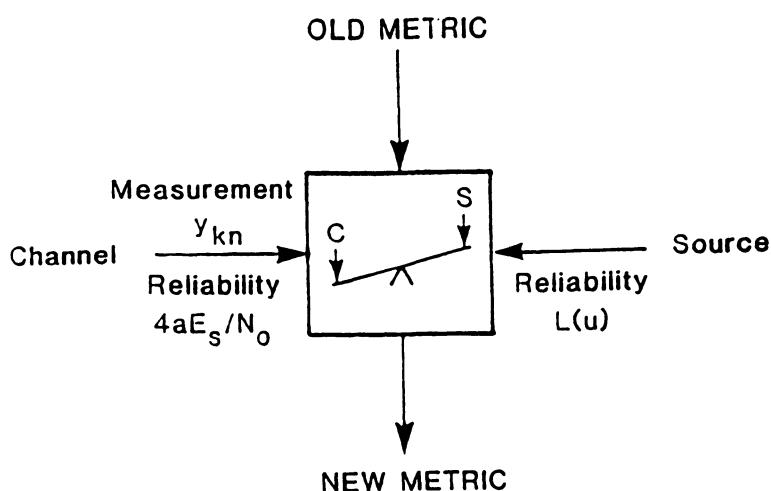
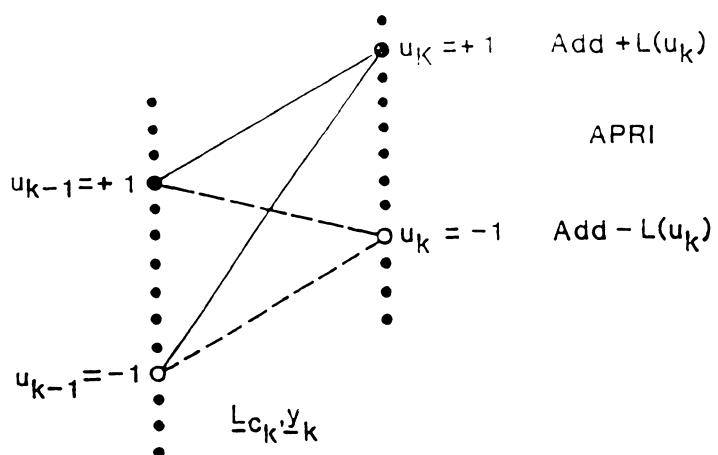
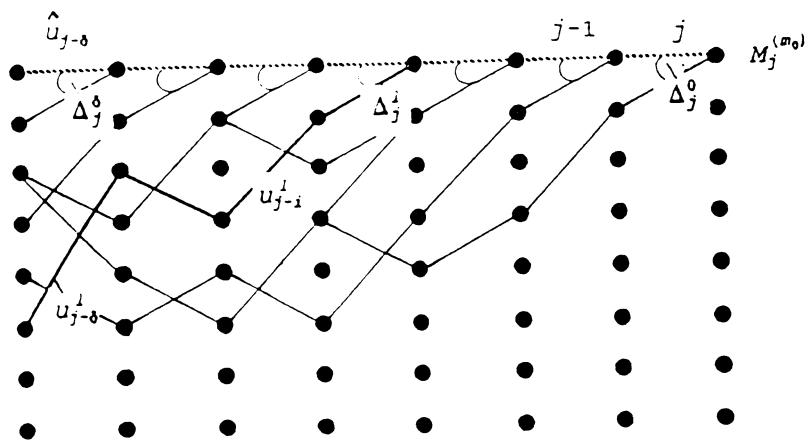


Fig. 2: Trellis and weighting property of the Viterbi algorithm with a-priori information (APRI-VA)



**Fig. 3:** Example trellis with metric differences for the derivation of the traceback SOVA.

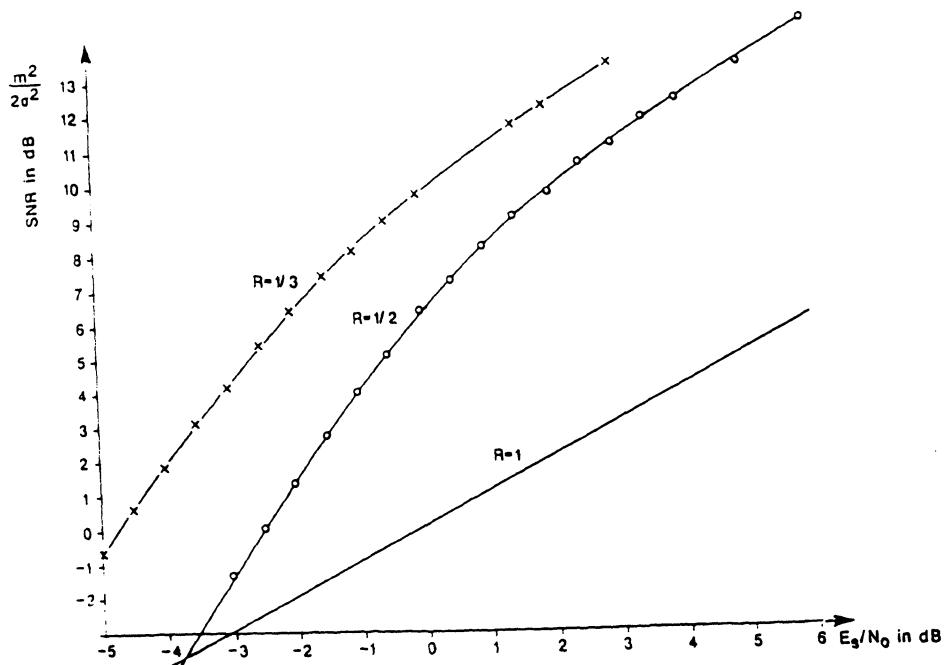
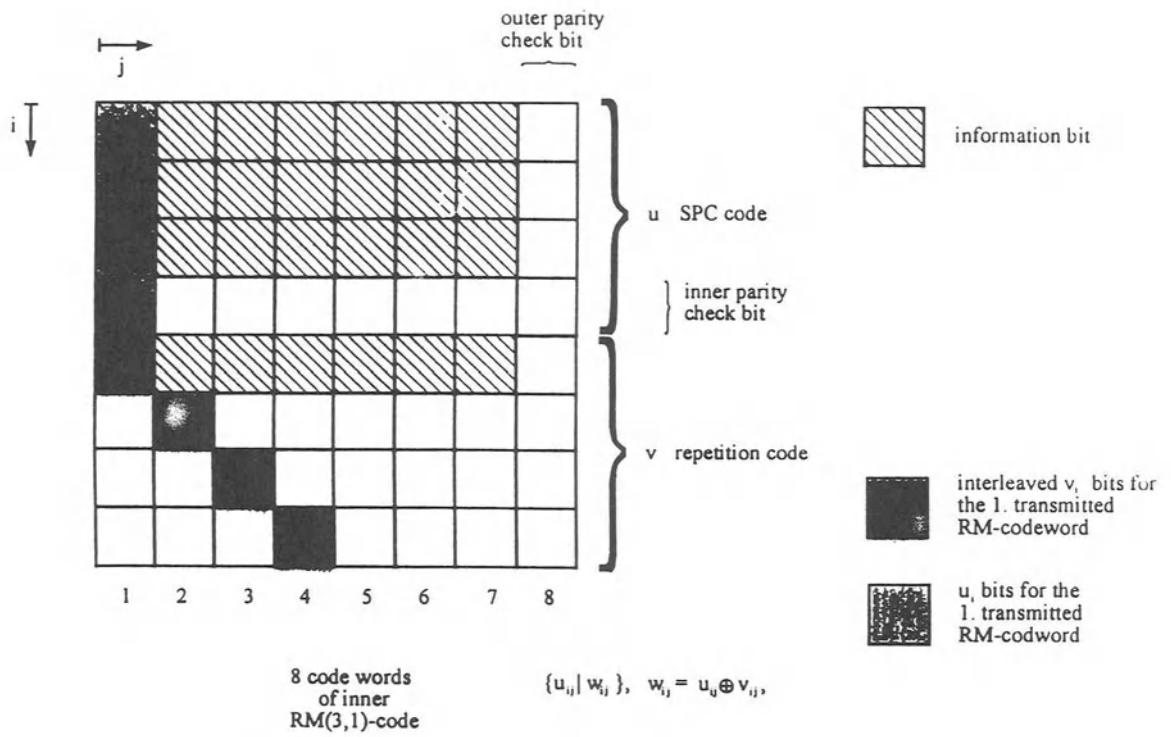


Fig. 4: Improvement of signal to noise ratio for soft-in/soft-out decoding of convolutional codes.



**Fig. 5:** Interleaving scheme for decoding the RM(3,1) code with parameters (8, 4, 4) as inner code and the SPC code ( $n, n-1, 2$ ) as outer code, shown for  $n = 8$ .

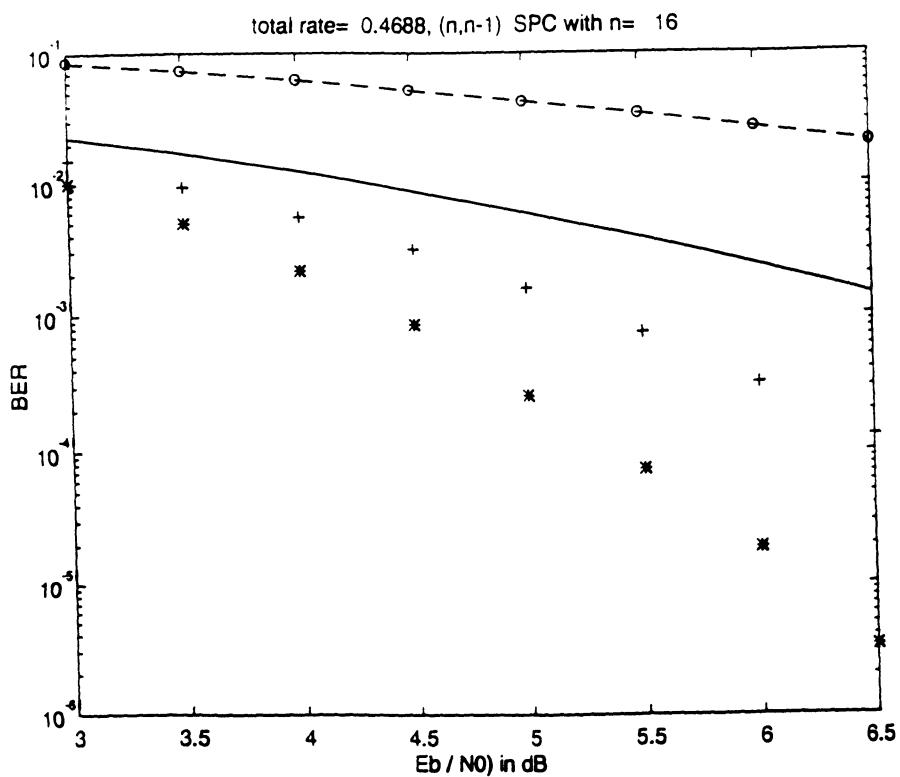


Fig. 6: Simulation result of the coding scheme of Fig.5 with  $n=16$ .  
+ RM code alone, \* Soft output RM with outer parity check code

# Charge Constrained Convolutional Codes

Mark A. Herro

Department of Electrical Engineering  
University of Notre Dame  
Notre Dame, IN 46556, USA.

Robert H. Deng

Department of Electrical Engineering  
National University of Singapore, 0511

Yuan Xing Li

Magnetic Technology Centre  
National University of Singapore, 0511

Dedicated to James L. Massey on his 60th birthday.

## Abstract

A new class of charge constrained (dc-free) convolutional codes is presented. They are constructed based on error-correcting convolutional codes and can be encoded and decoded using convolutional encoders and decoders with slight modifications. The dc-free error-correcting codes are especially well suited for applications in high-speed noisy channels and optical fiber communication systems.

## I Introduction

High speed transmission of digital data over optical fiber or metallic cable and the recording of data on magnetic or optical media often require transmission codes that have a spectral null at dc and suppressed low-frequency components [1]. In some applications it is also desirable that these codes provide some error-correcting capability. The construction of dc-free error-correcting line (transmission) codes has received much attention in the recent literature (see, for example [1]–[7]). Most of the dc-free error-correcting codes, however, are block codes. The objective of this paper is to present a class of convolutional codes that combine suppression of low-frequency components with significant error-correcting properties, and which can be encoded and decoded using convolutional encoders and decoders with slight modifications.

The motivation for constructing dc-free codes stems from applications in high-speed digital transmission and storage. In fiber optic digital transmission systems, (as also in wire-pair transmission systems), consecutive stages are often AC-coupled. As a result, an input pulse produces at the output a pulse with a tail of the opposite polarity. A succession of pulses with similar polarity at the input causes the “baseline wander” effect [8, 9] and the ability of the receiver

circuitry to distinguish between pulses present and absent can be hampered. In digital magnetic recording systems, a voltage is induced at the output of the read head by a change in polarity of the magnetization of the medium. Encoding schemes such as the NRZI system encode a *one* by a change in polarity, and a *zero* by no change in polarity. Self-clocking reading methods derive the clock signal from the data itself. This is facilitated if an output voltage is obtained frequently, and this in turn requires that there be a frequent change in the polarity of magnetization. Long strings of data symbols of like polarity are therefore apt to result in a loss of synchronization. This is also a problem for self-clocking receivers in fiber optic communication systems. For the reasons mentioned above, it is desirable to limit the number of consecutive symbols in the transmitted data, i.e., to limit the *runlength*.

The number of consecutive *zeros* that occur between two neighboring *ones* in a sequence is called the *runlength* of the sequence. The maximum runlength is a function of the *instantaneous accumulated charge* (or *disparity*) in the sequence. The disparity is defined as the difference between the number of *ones* and *zeros* in the encoded sequence. When the encoded *ones* and *zeros* are assigned values of +1 and -1 respectively, the disparity is equivalent to the *running digital sum* (RDS) of the transmitted symbols. The running digital sum (or disparity or instantaneous accumulated charge) is a measure of the energy of the transmitted data stream at dc. We let  $S(t)$  denote the RDS of a sequence at time  $t$ . A code is said to be *charge constrained* or have a bounded RDS if all possible information sequences can be encoded into sequences such that  $|S(t)| \leq D$  from some given positive integer  $D$ . For finite state encoders, a necessary and sufficient condition for a dc null in the power spectrum is that the RDS of the transmission code symbols takes values only in a finite range [10, 11], i.e., the code is charge constrained. In [4], we show that if a code is charge constrained, it is also runlength limited.

## II Description of the Codes

Consider an  $(n, k, m)$  convolutional code  $C_e$  with code rate  $R = k/n$  and memory order  $m$ . Let  $x(t) = (x_1(t)x_2(t)\dots x_k(t))$  be the input  $k$ -tuple to the encoder of  $C_e$  at some discrete time  $t$ . The output  $n$ -tuple  $y(t) = (y_1(t)y_2(t)\dots y_n(t))$  of the encoder is then given by [8]

$$y(t) = \sum_{\tau=0}^m x(t-\tau)G(\tau), \quad t = 0, 1, 2, \dots \quad (1)$$

where by way of convention  $x(t) = 0$ , for  $t < 0$ , and where

$$G(\tau) = \begin{bmatrix} g_{11}(\tau) & g_{12}(\tau) & \cdots & g_{1n}(\tau) \\ g_{21}(\tau) & g_{22}(\tau) & \cdots & g_{2n}(\tau) \\ \vdots & \vdots & & \vdots \\ g_{k1}(\tau) & g_{k2}(\tau) & \cdots & g_{kn}(\tau) \end{bmatrix}, \quad \tau = 0, 1, 2, \dots, m, \quad (2)$$

are the  $k \times n$  generator matrices of the code  $C_e$ .

Convolutional codes are not dc-free. This is evident since the all zero sequence is a valid code sequence of any convolutional code. The  $(n, k-1, m)$  dc-free error-correcting convolutional code, denoted by  $C_d$ , is constructed based on an  $(n, k, m)$  convolutional code  $C_e$ . Let  $d(t)$  denote the disparity of the encoder output  $n$ -tuple  $y(t)$ , and let  $S(t)$  denote the disparity up to the end of  $y(t)$  in the encoder output sequence,  $y(0), y(1), \dots, y(t)$ . The extra input bit in  $C_e$  is used to control the polarity of  $d(t)$  so that the output code sequence has bounded disparity (or, equivalently, a limited RDS). To this purpose, we require that at least one row, say the  $j$ th row, of the generator matrix  $G(0)$  (see Equation (2)) of  $C_e$  be the all-one vector  $\mathbf{1}_n$ . That is,

$$(g_{j1}(0)g_{j2}(0)\cdots g_{jn}(0)) = (1\ 1\ \cdots\ 1) = \mathbf{1}_n. \quad (3)$$

Consider an input  $k$ -tuple  $x(t) = (x_1(t), \dots, x_{j-1}(t), x_{j+1}(t), \dots, x_k(t))$  with  $x_j(t) = 0$ . Let  $y(t)$  be the corresponding output  $n$ -tuple. Now, if we set  $x_j(t) = 1$  and leave the other  $k-1$  input bits unchanged, it follows from Equations (1) and (3) that the output  $n$ -tuple becomes the complement of  $y(t)$ . Obviously, the disparities of  $y(t)$  and its complement have the same absolute value but opposite polarity. In other words, the input bit  $x_j(t)$  controls the polarity of the disparity of  $y(t)$ . Then, by alternating the disparity polarity of consecutive  $n$ -tuples in the code sequence, the sequence will have a limited RDS.

Encoding of  $C_d$ : Suppose  $(k-1)$  information bits  $(x_1(t), \dots, x_{j-1}(t), x_{j+1}(t), \dots, x_k(t))$  are to be encoded at time  $t$ . Form the  $k$ -tuple  $x(t) = (x_1(t), \dots, x_{j-1}(t), 0, x_{j+1}(t), \dots, x_k(t))$ . Use the notations  $S(t)$  and  $d(t)$  introduced earlier, and assume  $S(-1) = 0$ .

- 1) Encode  $x(t)$  by the encoder of  $C_e$  to get  $y(t)$ ; compute  $S(t-1)$  and  $d(t)$ . If  $S(t-1)d(t) \leq 0$ , let  $S(t) = S(t-1) + d(t)$ , output  $y(t)$ , go to step (3); else go to step 2).
- 2) Output  $y(t) \oplus \mathbf{1}_n$ , let  $S(t) = S(t-1) - d(t)$ , go to step 3).
- 3) Encode the next  $k-1$  information bits.

Decoding of  $C_d$ : Let  $y'(0), y'(1), \dots, y'(t), \dots$  be the received sequence corresponding to the transmitted sequence  $y(0), y(1), \dots, y(t), \dots$ . The received sequence may be decoded by any algorithm suitable for decoding convolutional codes. Let  $x'(t) = (x'_1(t), \dots, x'_{j-1}(t), x'_j(t), x'_{j+1}(t), \dots, x'_k(t))$  be the decoder output at time  $t$ . Then the estimate of the corresponding information  $(k-1)$ -tuple  $(x_1(t), \dots, x_{j-1}(t), x_{j+1}(t), \dots, x_k(t))$  is obtained from  $x'(t)$  with the  $x'_j(t)$  bit removed.

### III Properties of the Codes and Examples

The code  $C_d$  is a subset of the code  $C_e$ . Therefore,  $C_d$  is at least as powerful as  $C_e$  in error-correcting capability. In particular, we have  $d_d \geq d_e$ , where  $d_d$  and  $d_e$  are the free distances of  $C_d$  and  $C_e$ , respectively. If  $D$  denotes the maximum RDS of  $C_d$ , it can be shown [4] that

$$D \leq n + \lfloor n/2 \rfloor. \quad (4)$$

The actual value of  $D$  may be considerably smaller than the above bound.

The construction of the dc-free error-correcting code  $C_d$  requires the generator matrix  $G(0)$  of  $C_e$  satisfy Equation (3). It should be noted that not all the best known convolutional codes have this property. However, if the generator matrix  $G(m)$  has an all-one row, then by reversing the order of the generator matrices, Equation (3) will be satisfied (reversing the order of the generator matrices of a code does not change the code performance). Several  $(n, k-1, m)$  dc-free error-correcting convolutional codes derived from the best known  $(n, k, m)$  convolutional codes are given below, where the generator matrices are expressed in octal form.

$$\text{Code 1: } n = 3, k = 2, m = 1, G(0) = \begin{bmatrix} 7 \\ 4 \end{bmatrix}, \quad G(1) = \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \\ d_d \geq 3, D \leq 4.$$

$$\text{Code 2: } n = 4, k = 3, m = 2, G(0) = \begin{bmatrix} 74 \\ 24 \\ 14 \end{bmatrix}, \quad G(1) = \begin{bmatrix} 00 \\ 30 \\ 20 \end{bmatrix}, \quad G(2) = \begin{bmatrix} 00 \\ 00 \\ 14 \end{bmatrix}, \\ d_d \geq 4, D \leq 6.$$

$$\text{Code 3: } n = 5, k = 4, m = 1, G(0) = \begin{bmatrix} 76 \\ 56 \\ 32 \\ 42 \end{bmatrix}, \quad G(1) = \begin{bmatrix} 46 \\ 30 \\ 12 \\ 06 \end{bmatrix}, \\ d_d \geq 4, D \leq 7.$$

$$\text{Code 4: } n = 8, k = 5, m = 1, G(0) = \begin{bmatrix} 776 \\ 514 \\ 142 \\ 532 \\ 230 \end{bmatrix}, \quad G(1) = \begin{bmatrix} 026 \\ 052 \\ 116 \\ 214 \\ 416 \end{bmatrix}, \\ d_d \geq 7, D \leq 12.$$

In general, convolutional codes satisfying Equation (3) can be obtained by computer search, and may be inferior to the best known codes in error-correcting performance.

## IV Conclusions

In this note, we have presented a new class of charge constrained (dc-free) error-correcting codes. These codes are derived from convolutional codes and therefore can be encoded and decoded by the usual convolutional encoders and decoders, with only slight modifications. They are attractive for applications in high-speed channels, such as in optical fiber channels and optical recording systems.

## References

- [1] K. A. Schouhamer Immink, *Coding techniques for digital recorders*, Prentice Hall, New York, 1991.
- [2] H. C. Ferreira, "Lower bounds on the minimum Hamming distance achievable with runlength constrained or dc-free block codes and the synthesis of a  $(16,8)$   $d_{min} = 4$  dc-free block code", *IEEE Trans. Magn.*, MAG-19, pp. 2691-2693, 1983.
- [3] M. A. Herro and R. H. Deng, "Error-correcting DC-free binary transmission codes for fiber optic digital communications", in *Proc. Conf. Inf. Sci. Syst.*, Baltimore, MD, pp. 559-564, 1987.
- [4] R. H. Deng and M. A. Herro, "DC-free coset codes", *IEEE Trans. Inform. Theory*, IT-34, pp. 786-792, 1988.
- [5] G. D. Cohen and S. Litsyn, "DC-constrained error-correcting codes with small running digital sum", *IEEE Trans. Inform. Theory*, IT-37, 949-954, 1991.
- [6] T. Etzion, "Constructions of error-correcting dc-free block codes", *IEEE Trans. Inform. Theory*, IT-36, pp. 899-905, 1990.
- [7] M. Blaum, S. Litsyn, V. Buskens, and H. van Tilborg, "Error-correcting codes with bounded running digital sum", *IEEE Trans. Inform. Theory*, IT-39, pp. 216-227, 1993.
- [8] R. M. Brooks and A. Jessop, "Line coding for optical fibre systems", *Int. J. Electron.*, vol. 55, no. 1, pp. 81-120, 1983.
- [9] A. E. Brouwer, "A few new constant weight codes", *IEEE Trans. Inform. Theory*, IT-26, p. 336, 1980.
- [10] J. Justesen, "Information rates and power spectra of digital codes," *IEEE Trans. Inform. Theory*, IT-28, pp. 457-472, 1982.
- [11] G. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inform. Theory*, IT-30, pp. 435-439, 1984.
- [12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, NJ, 1983.

# **Delay Estimation for Truly Random Binary Sequences or How to Measure the Length of Rip van Winkle's Sleep**

Ingemar Ingemarsson  
Linköping University  
Sweden

## **Abstract**

Given a sequence generated by a binary symmetric memoryless source and a delayed version of the same sequence, the problem is to determine the delay. As a measure of complexity we use the number of comparisons of two digits in the sequence. A straightforward exhaustive search would compare the sequences after having delayed one of them each of the  $N$  possible delay values. On the average, two bits are compared before a mismatch is discovered. Hence the exhaustive method requires on the order of  $2N$  binary comparisons before all but one of the possible delay values are eliminated.

This paper constructs an algorithm that requires on the order of  $\sqrt{N} \log_2 N$  comparisons to determine the delay. It was previously known that at least  $\sqrt{N}$  comparisons are needed on the average before the delay is determined.

## **Foreword**

The following paper grew out of a very fruitful cooperation with Jim Massey which resulted in the conference paper [1]. The result of the research is an entirely impractical cipher since the recipient has to wait an exorbitant long time to decipher the message. This fact inspired Jim to the name Rip van Winkle cipher after the famous fairytale. The important theoretical result is that the cipher was provably secure, which later motivated the important research on provably secure randomized ciphers by Ueli Maurer [2].

The full paper on the Rip van Winkle cipher was never published except as a conference paper. The present contribution is a constructive algorithm to measure the delay of a binary sequence.

## **I Introduction**

A sequence  $X$  generated by a memoryless binary symmetric source and a delayed version of  $X$  are observed. The problem is to determine the delay. This problem arises for example in

direct sequence spread spectrum communication systems where the receiver has to synchronize his added sequence to the sequence used by the transmitter, i.e. he has to determine the delay. The same problem also appears when trying to cryptanalyze the Rip van Winkle cipher [1].

A straightforward exhaustive search would compare the sequences after having delayed one of them each of the  $N$  possible delay values. On the average, two bits are compared before a mismatch is discovered, so this method requires on the average  $2N$  binary comparisons before the delay is determined. Our goal is to minimize the average number of comparisons of binary digits needed to determine the delay. In [1] we have proven that at least  $\sqrt{N}$  comparisons are needed. Here we present an algorithm that requires on the average  $\frac{\sqrt{N}}{2} \log_2 N$  comparisons before the delay is determined.

## II Notation and Preliminaries

$X = \{x_i\}$  is a sequence generated by a memoryless binary symmetric source.  $Y = \{y_1\}$  is a delayed version of  $X$ , i.e.

$$y_i = x_{i-d}$$

for

$$0 \leq d < N$$

Here  $N$  is the maximum delay.  $N$  is assumed to be a perfect square. The actual delay  $d$  is unknown to the observer of  $X$  and  $Y$ . His problem is to determine  $d$ . The sequences  $X$  and  $Y$  are sampled in different ways. Consecutive samples are taken from  $X$ . At integer time instant  $t$  (starting at  $t = 1$ ) we obtain  $x_t$ .  $Y$  is sparsely sampled. Initially we need the sequence  $y_{\sqrt{N}}, y_{2\sqrt{N}}, \dots, y_N$ . Later we will need samples of the type  $y_{j+i\sqrt{N}}$  for  $j = 1, 2, \dots$

## III The Algorithm

The algorithm can be described as the growing of a binary tree. The nodes are numbered in the regular binary way. Two items are associated with each node. The first is either a null or an integer. The integer is the first position in the sequence  $x_1, x_2, \dots$  from which the node number can be read. Thus the node number is  $x_i, x_{i+1}, x_{i+2}, \dots$  where  $i$  is the associated integer. If no such position exists, the item is a null.

The second item associated with each node is the set of integers  $\{i\}$  in the range  $[1, \sqrt{N}]$ , such that the node number is  $y_{i\sqrt{N}}, y_{1+i\sqrt{N}}, y_{2+i\sqrt{N}}, \dots$ . To each integer  $i$  there is also associated a counting integer variable,  $c_i$ . Initially this variable is set to zero.  $c_i$  is incremented by one each time the integer  $i$  is associated with a node for which the first item is a null. The variables  $c_i$  are used in terminating the algorithm. It is shown below that a branch can be determined when  $c_i = \sqrt{N}$ .

The first item is generated in the following way. At the root the first item is initially 1 and is incremented by one each time instant. At time  $t$  the first item is moved to the left or right in the tree according to whether  $x_t$  is zero or one respectively. The process is illustrated in Figure 1. At time  $t$  the node labeled 1 is the binary  $t$ -bit word  $x_1, x_2, \dots, x_t$ , the node labeled 2 is

Figure 1: Comparisons made between  $y$  and  $x$  and the resulting exclusion of delay values.

|          |       | → Time,t |   |   |   |   |   |   |   |   |    |    | → Time,t              |   |   |   |   |   |    |    |    |    |    |    |   |
|----------|-------|----------|---|---|---|---|---|---|---|---|----|----|-----------------------|---|---|---|---|---|----|----|----|----|----|----|---|
|          |       | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 1                     | 2 | 3 | 4 | 5 | 6 | 7  | 8  | 9  | 10 | 11 |    |   |
|          | $x_t$ | 0        | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0  | 1  | Excluded delay values |   |   |   |   |   |    |    |    |    |    |    |   |
| $y_5$    | 0     | 0        |   |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 0     |          | 0 |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 0     |          |   | 1 |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    | 4  | 3  | 2  | 1  | 0 |
|          | 0     |          |   |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
| $y_{10}$ | 1     | 1        |   |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   | 9  | 8  |    |    |    |    |   |
|          | 1     |          |   | 0 |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 0     |          |   |   | 0 |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 1     |          |   |   |   | 0 |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 1     |          |   |   |   |   | 0 |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
| $y_{15}$ | 1     | 1        |   |   |   |   |   | 0 |   |   |    |    |                       |   |   |   |   |   | 14 | 13 |    |    |    |    |   |
|          | 0     |          |   | 1 |   |   |   |   | 0 |   |    |    |                       |   |   |   |   |   |    |    | 12 |    |    |    |   |
|          | 0     |          |   |   | 1 |   |   |   |   | 0 |    |    |                       |   |   |   |   |   |    |    | 11 | 10 |    |    |   |
|          | 1     |          |   |   |   |   |   |   |   | 0 |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
| $y_{20}$ | 0     | 0        |   |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 1     |          | 1 |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    | 19 |    |    |    |   |
|          | 1     |          |   | 0 |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 1     |          |   |   | 1 |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    | 18 | 17 | 16 |    |   |
|          | 1     |          |   |   |   |   |   | 1 |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    | 15 |   |
| $y_{25}$ | 1     | 1        |   |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |
|          | 0     |          | 1 |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    | 24 | 23 |    |    |   |
|          | 1     |          |   |   | 0 |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    | 22 |    |    |    |   |
|          | 0     |          |   |   |   | 1 |   |   |   |   |    |    |                       |   |   |   |   |   |    |    | 21 | 20 |    |    |   |
|          | 0     |          |   |   |   |   |   |   |   |   |    |    |                       |   |   |   |   |   |    |    |    |    |    |    |   |

the binary  $(t - 1)$ -bit word  $x_2, x_3, \dots, x_t$ . Unlabeled nodes cannot be found as consecutive bits  $x_x, \dots, x_t$  in  $x_1, x_2, \dots, x_t$ .

The second item starts as the set of integers  $\{1, 2, \dots, \sqrt{N}\}$  associated with the root of the tree. At  $t = 1$  the set is split so that the subset  $\{i\}$  for which  $y_{i\sqrt{N}}$  is 0 is associated with node 0 and the subset for which  $y_{i\sqrt{N}}$  is 1 is associated with node 1. At  $t = 2$  the set of integers associated with each node is split so that the number is  $y_{i\sqrt{N}}, y_{1+i\sqrt{N}}$ . This is done sequentially. At depth  $j$  if  $y_{j+i\sqrt{N}}$  is 0 that particular integer  $i$  is directed to the node number ending with 0 and if  $y_{j+i\sqrt{N}}$  is 1,  $i$  is diverted to the node number ending with a 1. If no such integer  $i$  exists the tree is pruned immediately to that node.

Since  $Y$  is a delayed version of  $X$  there will be at least one remaining node for which the first item is not a null. Suppose the first item is  $k$ . Then the following two binary words are equal.

$$\begin{aligned} &x_k, \quad x_{1+k}, \quad \dots \quad x_{j-1+k} \\ &y_{i\sqrt{N}}, \quad y_{1+i\sqrt{N}}, \quad \dots \quad y_{j-1+i\sqrt{N}} \end{aligned}$$

Here  $i$  is an integer in the set (item 2) associated with the node at depth  $j$ . Since the words are equal a candidate for the delay,  $d$ , is:

$$\hat{d} = i\sqrt{N} - k.$$

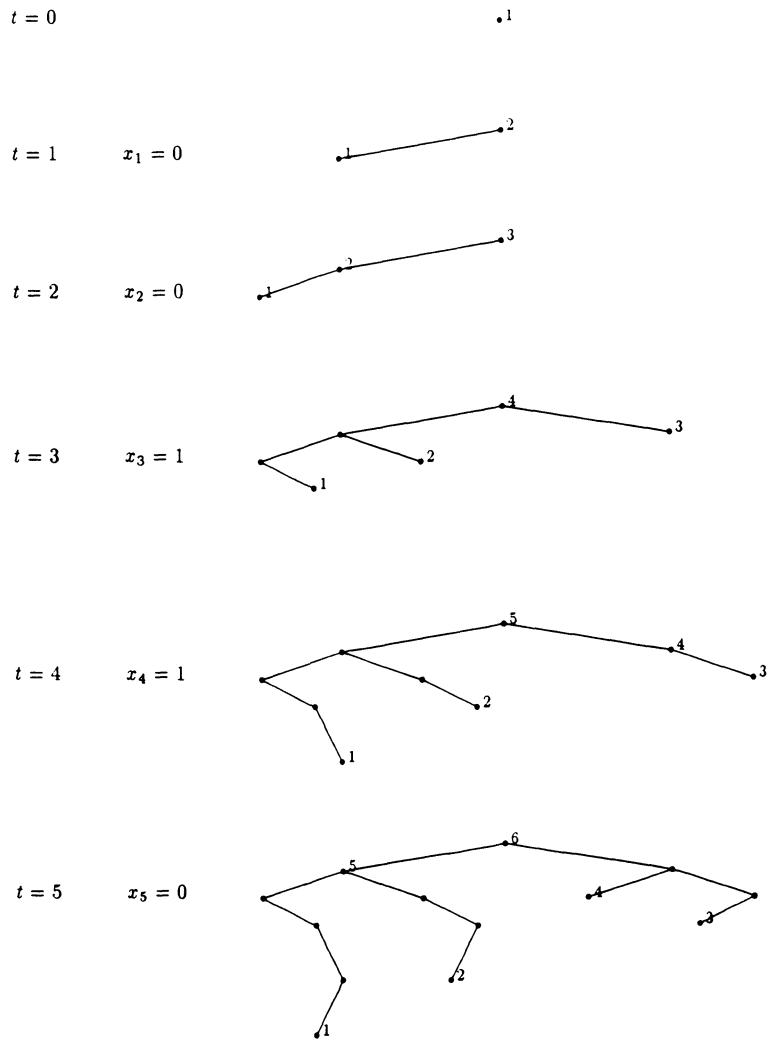


Figure 2: Association of the first item to the tree

## IV An Example

To illustrate the algorithm and to see how it terminates we give an example, which is an extension of Figure 1. As before we assume that  $N = 25$ . Figure 2 shows a table where the comparisons of  $y$  and  $x(x \oplus y)$  are marked. The corresponding growing tree is shown in Figure 3 with the nodes marked with the first item (single integers or nothing) and the second item (a set of integers and the variable  $c_i$  in a rectangular frame).

If we follow the table from the top left we see that  $x_1 = 0$  and since  $y_5 = y_{20} = x_1$  the set  $\{1, 4\}$  follows the root value 1 to node 0. The remaining set  $\{2, 3, 5\}$  for which  $y_{i\sqrt{N}} \neq x_1$  is associated with node 1. Hence the delay values 9, 14 and 24 can be excluded. The set  $\{2, 3, 5\}$  sits at node 1 until  $x_i = 1$ . Thus we can exclude, without making further comparisons, the delay values 8, 13 and 23, since  $x_2 = 0$ . This is shown in the right table in Figure 2.

The set  $\{1, 4\}$  at node 0 is split into  $\{1\}$  at node 00 (since  $y_6 = 0$ ) and  $\{4\}$  at node 01 (since  $y_{21} = 1$ ). Since  $x_2 = 0$  the first item (2) at node 0 is moved to node 00 and the delay value  $21 - 2 = 19$  is excluded. This last part required two comparisons ( $x_2 \oplus y_6$  and  $x_2 \oplus y_{21}$ ).

At time instant  $t = 3$  we can prune the tree at node 001. This is because  $y_{2+i\sqrt{N}} = 0$  for all  $i$  in the set  $\{1\}$  associated with node 00 at depth 2. We have only  $i = 1$  and  $y_7 = 0$ .

Item two,  $\{1\}$  sits at node 000 until another 0 in the  $x$ -sequence causes the first item at node 000 to be different from null. Actually this never happens because meanwhile the delay values 4, 3, 2, 1 and 0 are excluded (see Figure 2) and there is no meaning to continue since we have limited the delay to nonnegative values only.

The role of the variable  $c_i$  is now clear.  $c_i$  is increased by 1 each time the first item is a null. This means that another delay value is excluded. Since

$$\hat{d} = i\sqrt{N} - k$$

a particular integer  $i$  covers  $\sqrt{N}$  delay values,  $[i\sqrt{N} - \sqrt{N}, i\sqrt{N} - 1]$ . This means that there is no need to grow the tree beyond the point for which  $c_i = \sqrt{N}$ . These termination points are denoted by capital  $T$  in Figure 3.

The remaining node (at  $t = 94$ ) has number 1101110. This is the subsequence of  $X$  starting at  $x_3$ , as indicated by the node label 3. It is also a subsequence of  $Y$  starting at  $y_{10}$ . Here  $10 = 2\sqrt{25}$  where 2 is the remaining integer in the set forming item 2. Thus the only surviving candidate for the delay is

$$\hat{d} = 2\sqrt{25} - 3 = 7$$

Actually most other delay values have been excluded, as can be seen from the right table in Figure 2. Note, however, that because the node labeled 3 never terminated the hypothetical delays 5 and 6 are not excluded. In fact, in a general case, none of the delay values in the range from  $\hat{d}$  down to the multiple of  $\sqrt{N}$  closest below  $d$  will be tested by the algorithm. (Here  $\hat{d}$  is the single remaining delay candidate.) This is not a serious drawback, however, since we have discovered in the order of  $\sqrt{N}$  symbols for which  $y_i = x_{i-\hat{d}}$ . (This corresponds to the diagonal row of 8 zeroes in Figure 2.) Thus the probability of  $\hat{d}$  being false is only about  $2^{-\sqrt{N}}$ .

In the example there is a total of 24 comparisons made, which is pretty close to  $\sqrt{N} \log_2 N$ . This is typical for the examples we have run through. Next we will do an approximate performance analysis to estimate the number of comparisons needed to determine the delay.

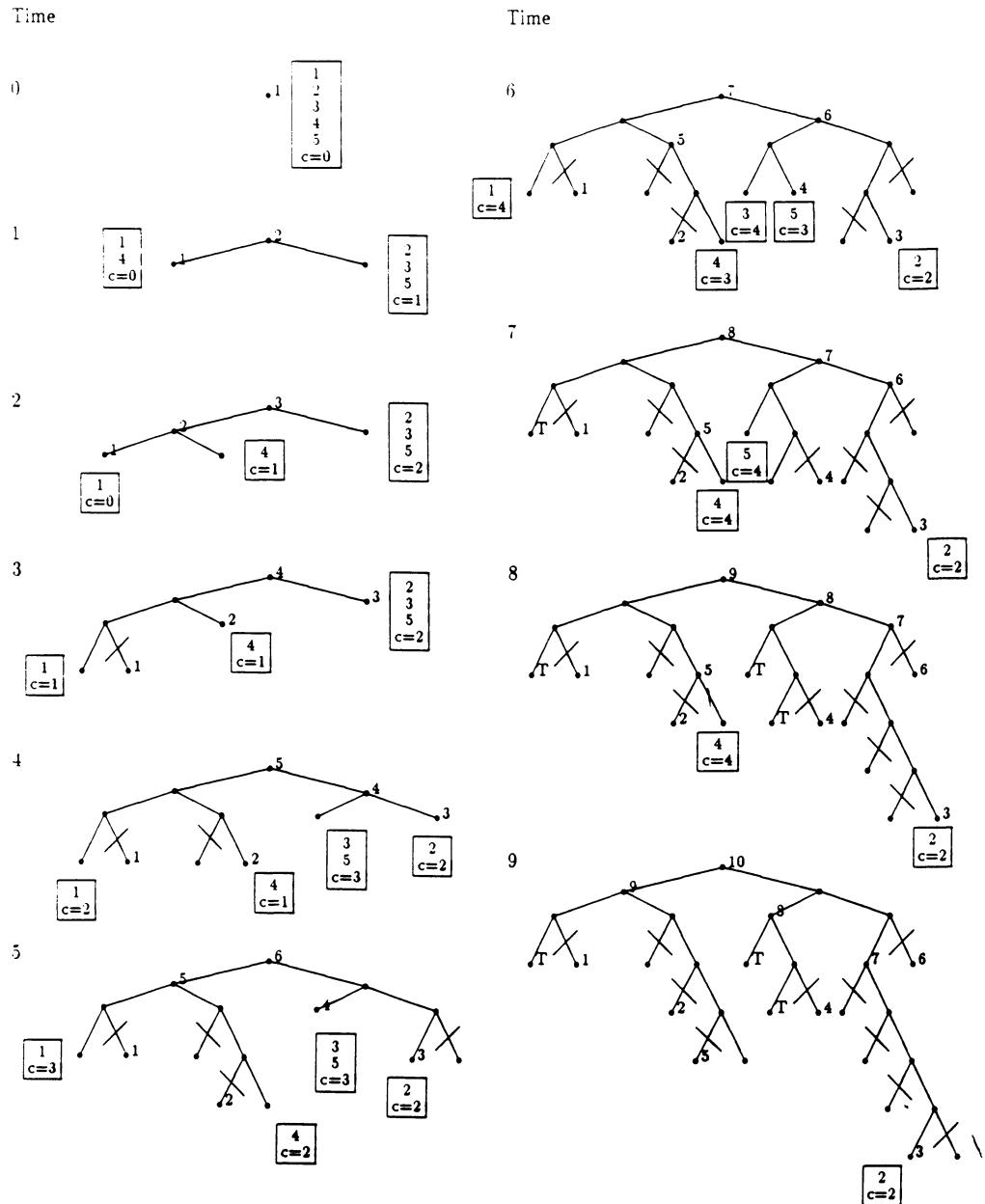


Figure 3: The growing tree corresponding to the example in Figure 2.

## V Approximate Performance Analysis

Figure 3 is a good illustration of a typical situation. A particular  $i$  in item 2 (in the rectangular frame) sits at a node at depth 1 until item 1 at that node is not null, i.e. until  $y_{i\sqrt{N}} = x_t$  for some  $t$ . This requires only one comparison (when  $i$  is leaving the root). Since  $X$  is a memoryless balanced binary sequence  $t = 2$  on the average when  $y_{i\sqrt{N}} = x_t$ . Thus on the average we have excluded one delay value ( $i\sqrt{N} - 1$ ).

The particular integer  $i$  we are regarding now moves to depth 2 in the tree and stays there until item 1 at that node is not null. This happens when the two-bit binary words  $y_{i\sqrt{N}}, y_{1+i\sqrt{N}}$  and  $x_t$  and  $x_{t+1}$  are equal for some  $t$ . The average time  $i$  stays at the node is:

$$\sum_{j=0}^{\infty} j \left(\frac{3}{4}\right)^{j-1} \frac{1}{4} = 4$$

Thus on the average three more delay values are excluded. In general at depth  $d$  in the tree the average waiting time until the  $d$ -bit words  $y_{i\sqrt{N}} \dots$  and  $x - t \dots$  match is:

$$\sum_{j=0}^{\infty} j (1 - 2^{-d})^{j-1} 2^{-d} = 2^d$$

This means that on the average  $2^d - 1$  delay values are excluded at a node at depth  $d$ . Note that for a particular  $i$  only one comparison per node is required.

If we do not hit the correct delay value the branch terminates when a total of  $\sqrt{N}$  delay values have been excluded for a particular  $i$ . Let  $d_T$  be the average depth when this happens. Then:

$$\sum_{j=1}^{d_T} (2^j - 1) = \sqrt{N}$$

$$2^{1+d_T} - 2 - d_T = \sqrt{N}$$

For large  $N$ :

$$d_T \approx \frac{1}{2} \log_2 N - 1$$

For each integer  $i$  in item 2 we have to make  $1 = d_T$  comparisons on the average. The average value of the total number of comparisons is thus:

$$\sqrt{N} \cdot (1 + d_T) = \frac{\sqrt{N}}{2} \log_2 N$$

Here we have disregarded the fact that the correct value of  $d$  is hit for one particular  $i$ . This will slightly decrease the above average.

## References

- [1] J. L. Massey and I. Ingemarsson, “The Rip van Winkle cipher: A simple and provably computationally secure cipher with a finite key”. *IEEE International Symposium on Information Theory*, Abstracts of Papers, p. 146, 1985.
- [2] U. Maurer, “Conditionally-perfect secrecy and a provably-secure randomized cipher”, *Journal of Cryptology*, Vol. 5, no. 1, pp 53-66, 1992.

# On Canonical Encoding Matrices and the Generalized Constraint Lengths of Convolutional Codes

Rolf Johannesson  
Dept. of Information Theory  
Lund University  
S-221 00 LUND, Sweden

Zhe-xian Wan  
Dept. of Information Theory  
Lund University  
S-221 00 LUND, Sweden  
and  
Institute of Systems Science  
Chinese Academy of Science  
Beijing 100080, China

## Abstract

This paper is devoted to rational convolutional encoding matrices. Canonical encoding matrices are introduced and it is shown that every canonical encoding matrix is minimal but that there exist minimal encoding matrices that are not canonical. Some equivalent conditions for an encoding matrix to be canonical are given. The generalized constraint lengths are defined. They are invariants of equivalent canonical encoding matrices.

## I Introduction

Jim Massey made early contributions of greatest importance to the structural theory of convolutional encoders. Together with Sain [1] he defined two convolutional encoding matrices to be *equivalent* if they encode the same code. They also proved that every convolutional code can be encoded by a *polynomial* encoding matrix. Later they studied conditions for a convolutional encoding matrix to have a polynomial right inverse [2]. Massey's work in this area was continued by his students [3], [4]. By exploiting the invariant-factor theorem and the realization theory of linear systems Forney extended these results [5]–[7]. Recently, there has been a renewed interest in these problems, see, e.g., [8]–[12].

In this paper we study binary rational convolutional encoding matrices. In Section II we introduce the generalized constraint lengths of a rational encoding matrix. Canonical encoding matrices are defined in Section III and we show that every canonical encoding matrix is minimal. Section IV is devoted to exponential valuations and the defect. In Section V we prove that the generalized constraint lengths are invariants of equivalent canonical encoding matrices. Finally, in Section VI we give some equivalent conditions for an encoding matrix to be canonical and show, by giving an example, the existence of minimal encoding matrices that are not canonical.

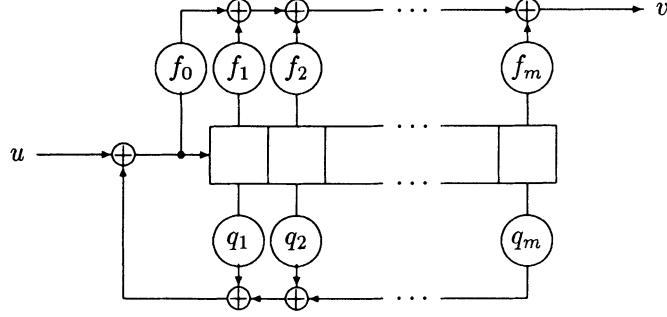


Figure 1: The controller canonical form of a rational transfer function.

## II Generalized Constraint Lengths

Consider the controller canonical form of a single-input single-output linear system as shown in Figure 1. The delay elements form a shift register, the output is a linear function of the input and the shift register contents, and the input to the shift register is a linear function of the input and the shift register contents. From Figure 1 it follows that

$$v(D) = u(D)f(D)/q(D), \quad (1)$$

where

$$f(D) = f_0 + f_1D + \dots + f_mD^m \quad (2)$$

and

$$q(D) = 1 + q_1D + \dots + q_mD^m. \quad (3)$$

Let  $g(D) = f(D)/q(D)$ , then  $v(D) = u(D)g(D)$  and we say that  $g(D)$  is a *rational transfer function* which transfers the input  $u(D)$  into the output  $v(D)$ . From (1), it follows that every rational function with a constant term 1 in the denominator polynomial  $q(D)$  (or, equivalently, with  $q(0) = 1$  or, again equivalently, with  $q(D)$  *delay-free*) is a rational transfer function that can be realized in the controller canonical form shown in Figure 1. Every rational function  $f(D)/q(D)$ , where  $q(D)$  is delay-free, is called a *realizable function*.

Let  $g_1(D), \dots, g_c(D)$  be realizable functions. We may write

$$g_i(D) = f_i(D)/q(D), \quad i = 1, \dots, c, \quad (4)$$

where  $f_1(D), \dots, f_c(D), q(D) \in \mathbb{F}_2[D]$  and

$$\gcd(f_1(D), \dots, f_c(D), q(D)) = 1. \quad (5)$$

We define the *generalized constraint length* of the  $1 \times c$  matrix

$$\mathbf{g}(D) = (g_1(D), \dots, g_c(D)) \quad (6)$$

as

$$\nu = \max\{\deg f_1(D), \dots, \deg f_c(D), \deg q(D)\}. \quad (7)$$

Clearly,  $\mathbf{g}(D)$  can be realized with  $\nu$  delay elements in controller canonical form.

A rate  $R = b/c$  convolutional code  $\mathcal{C}$  over  $\mathbb{F}_2$  with generator matrix  $G(D)$  is the image set of a linear mapping

$$\begin{aligned} \mathbb{F}_2^b((D)) &\rightarrow \mathbb{F}_2^c((D)) \\ \mathbf{u}(D) &\mapsto \mathbf{v}(D), \end{aligned} \quad (8)$$

which can be represented as

$$\mathbf{v}(D) = \mathbf{u}(D)G(D), \quad (9)$$

where  $G(D)$  is a  $b \times c$  transfer function matrix of rank  $b$  over  $\mathbb{F}_2(D)$ . Hence, a convolutional code can be regarded as a block code over the field of Laurent series  $\mathbb{F}_2((D))$  which has  $G(D)$  as its generator matrix or as the  $\mathbb{F}_2((D))$  row space of  $G(D)$ .

A generator matrix of a convolutional code is called an *encoding matrix* of the code if it is realizable and delay-free, i.e., all of its entries  $g(D)/q(D)$  are realizable and at least one has  $g(0) \neq 0$  (c.f. [11]).

We define the *generalized  $i$ -th constraint length*  $\nu_i$  of a rational encoding matrix as the generalized constraint length of the  $i$ th row of the encoding matrix, the *generalized memory*  $m$  as

$$m = \max_{1 \leq i \leq b} \{\nu_i\}, \quad (10)$$

and the *generalized overall constraint length*  $\nu$  as

$$\nu = \sum_{i=1}^b \nu_i. \quad (11)$$

For a polynomial encoding matrix these definitions coincide with Forney's original definitions of the  $i$ th constraint length, memory, and overall constraint length [5].

Clearly, a rational encoding matrix with generalized overall constraint length  $\nu$  can be realized with  $\nu$  memory elements in controller canonical form.

### III Canonical Encoding Matrices

In [11] we defined a *minimal-basic* encoding matrix to be a basic encoding matrix whose overall constraint length  $\nu$  is minimal over all equivalent basic encoding matrices. It is clear that a polynomial encoding matrix whose overall constraint length is minimal over all equivalent polynomial encoding matrices is basic, and, hence, minimal-basic [10]. This leads naturally to the following generalization of the concept of minimal-basic encoding matrices.

A *canonical* encoding matrix is a rational encoding matrix whose generalized overall constraint length  $\nu$  is minimal over all equivalent rational encoding matrices.

In [11] we defined an encoding matrix to be *minimal* if its number of abstract states is minimal over all equivalent encoding matrices<sup>1</sup> and we proved that a minimal-basic encoding matrix is minimal. We have immediately the following two theorems.

**Theorem 1** *A minimal-basic encoding matrix is canonical.*

**Proof** Let  $G_{mb}(D)$  be a minimal-basic encoding matrix with overall constraint length  $\nu_{mb}$  and let  $G_c(D)$  be an equivalent canonical encoding matrix with overall constraint length  $\nu_c$ . Then  $\nu_c \leq \nu_{mb}$ . Since  $G_{mb}(D)$  is minimal its number of abstract states,  $2^{\nu_{mb}}$ , is minimal over all equivalent encoding matrices. Thus,  $2^{\nu_{mb}} \leq \#\{\text{abstract states of } G_c(D)\} \leq 2^{\nu_c}$ . Therefore,  $\nu_{mb} \leq \nu_c$ . Hence,  $\nu_{mb} = \nu_c$  and  $G_{mb}(D)$  is canonical.  $\square$

**Theorem 2** *A canonical encoding matrix is minimal.*

**Proof** From the proof of Theorem 1 it follows that the number of abstract states of  $G_c(D)$ ,  $2^{\nu_c}$ , is minimal over all equivalent encoding matrices. Hence,  $G_c(D)$  is minimal.  $\square$

**Lemma 3** *Let  $\mathbf{g}(D) = (g_1(D), \dots, g_c(D))$  be a  $1 \times c$  rational encoding matrix, where  $g_1(D), \dots, g_c(D) \in \mathbb{F}_2(D)$ . Write*

$$g_i(D) = f_i(D)/q(D), \quad i = 1, \dots, c, \quad (12)$$

and assume that

$$\gcd(f_1(D), \dots, f_c(D), q(D)) = 1. \quad (13)$$

Then  $\mathbf{g}(D)$  is canonical if and only if both (i) and (ii) hold:

- (i)  $\deg q(D) \leq \max\{\deg f_1(D), \dots, \deg f_c(D)\}$
- (ii)  $\gcd(f_1(D), \dots, f_c(D)) = 1$ .

**Proof** Let  $\mathbf{f}(D) = (f_1(D), \dots, f_c(D)) = \gcd(f_1(D), \dots, f_c(D))\mathbf{l}(D)$ . It is clear that  $\mathbf{g}(D)$ ,  $\mathbf{f}(D)$ , and  $\mathbf{l}(D)$  are equivalent encoding matrices.

Suppose that  $\mathbf{g}(D)$  is canonical. Then

$$\begin{aligned} \nu_g &\equiv \max\{\deg f_1(D), \dots, \deg f_c(D), \deg q(D)\} \\ &\leq \nu_f \equiv \max\{\deg f_1(D), \dots, \deg f_c(D)\}, \end{aligned} \quad (14)$$

from which (i) and  $\nu_g = \nu_f$  follow.

Moreover,

$$\nu_g = \nu_f = \deg \gcd(f_1(D), \dots, f_c(D)) + \nu_l, \quad (15)$$

where  $\nu_l$  is the constraint length of  $\mathbf{l}(D)$ . From (15), the equivalence of encoding matrices  $\mathbf{f}(D)$  and  $\mathbf{l}(D)$ , and the assumption that  $\mathbf{g}(D)$  is canonical, it follows that

$$\deg \gcd(f_1(D), \dots, f_c(D)) = 0, \quad (16)$$

---

<sup>1</sup>In [9] Forney first gave a definition of a *generalized minimal encoder* that coincides with our definition of a *minimal encoding matrix*, then he gave a redefinition that coincides with our definition of *canonical encoding matrix*. The existence of minimal encoding matrices which are not canonical is shown by Example 4.

which is equivalent to (ii).

Conversely, suppose that (i) does not hold, i.e., that

$$\deg q(D) > \max\{\deg f_1(D), \dots, \deg f_c(D)\}. \quad (17)$$

From (17) follows that

$$\nu_g = \deg q(D) > \nu_f, \quad (18)$$

and, hence, since  $\mathbf{g}(D)$  and  $\mathbf{f}(D)$  are equivalent, that  $\mathbf{g}(D)$  is not canonical.

Finally, suppose that (i) holds and that (ii) does not hold. Then from (15) follows that  $\nu_g > \nu_l$  and, since  $\mathbf{g}(D)$  and  $\mathbf{l}(D)$  are equivalent, that  $\mathbf{g}(D)$  is not canonical.  $\square$

## IV Exponential Valuations and the Defect

Let  $p(D)$  be an irreducible polynomial of  $\mathbb{F}_2[D]$ . For any nonzero  $g(D) \in \mathbb{F}_2(D)$  we can express  $g(D)$  as

$$g(D) = p(D)^{e_{p(D)}(g(D))} h(D)/d(D), \quad (19)$$

where  $e_{p(D)}(g(D)) \in \mathbb{Z}$ ;  $h(D), d(D) \in \mathbb{F}_2[D]$ ;  $\gcd(h(D), d(D)) = 1$ ; and  $p(D) \nmid h(D)d(D)$ . Furthermore,  $e_{p(D)}(g(D))$  is uniquely determined by  $g(D)$ . We define  $e_{p(D)}(0) = \infty$ . The map

$$\begin{aligned} e_{p(D)} : \mathbb{F}_2(D) &\rightarrow \mathbb{Z} \cup \{\infty\} \\ g(D) &\mapsto e_{p(D)}(g(D)) \end{aligned} \quad (20)$$

is an *exponential valuation* of  $\mathbb{F}_2(D)$  [13]. Moreover, for any nonzero  $g(D) \in \mathbb{F}_2(D)$  we can express  $g(D)$  as

$$g(D) = f(D)/q(D), \quad (21)$$

where  $f(D), q(D) \in \mathbb{F}_2[D]$ . We define

$$e_{D^{-1}}(g(D)) = \deg q(D) - \deg f(D) \quad (22)$$

and  $e_{D^{-1}}(0) = \infty$ . Then  $e_{D^{-1}}$  is also an exponential valuation of  $\mathbb{F}_2(D)$  [13].

Denote by

$$\mathcal{P}^* = \{p(D) \in \mathbb{F}_2[D] \mid p(D) \text{ is irreducible}\} \cup \{D^{-1}\}. \quad (23)$$

Then from (19) and (22) we have the *product formula* (written in additive form)

$$\sum_{p \in \mathcal{P}^*} e_p(g(D)) \deg p = 0, \quad (24)$$

for all nonzero  $g(D) \in \mathbb{F}_2(D)$  [13].

**Example 1** Let  $g(D) = (D^3 + D^5)/(1 + D + D^2)$ . Then  $e_{1+D+D^2}(g(D)) = -1$ ,  $e_{1+D}(g(D)) = 2$ ,  $e_D(g(D)) = 3$ ,  $e_{D^{-1}}(g(D)) = -3$ , and

$$e_p(g(D)) = 0 \text{ if } p \in \mathcal{P}^* \text{ and } p \neq 1 + D + D^2, 1 + D, D, D^{-1}. \quad (25)$$

It is easy to verify that

$$\sum_{p \in \mathcal{P}^*} e_p(g(D)) \deg p = 0. \quad (26)$$

□

Let  $\mathbf{g}(D) = (g_1(D), \dots, g_c(D))$ , where  $g_1(D), \dots, g_c(D) \in \mathbb{F}_2(D)$ . For any  $p \in \mathcal{P}^*$  we define

$$e_p(\mathbf{g}(D)) = \min\{e_p(g_1(D)), \dots, e_p(g_c(D))\}. \quad (27)$$

Now we can write Lemma 3 in a more symmetric form:

**Lemma 3'** Let  $\mathbf{g}(D) = (g_1(D), \dots, g_c(D))$  be a  $1 \times c$  encoding matrix over  $\mathbb{F}_2(D)$ . Then  $\mathbf{g}(D)$  is canonical if and only if

$$e_p(\mathbf{g}(D)) \leq 0, \text{ all } p \in \mathcal{P}^*. \quad (28)$$

**Proof** We will prove that (28) is equivalent to (i) and (ii) of Lemma 3. From (12), (13), and (27) it follows that

$$e_{D^{-1}}(\mathbf{g}(D)) = \deg q(D) - \max\{\deg f_1(D), \dots, \deg f_c(D)\}. \quad (29)$$

Hence,

$$e_{D^{-1}}(\mathbf{g}(D)) \leq 0 \Leftrightarrow \deg q(D) \leq \max\{\deg f_1(D), \dots, \deg f_c(D)\}. \quad (30)$$

For the second half of the proof, let  $p(D)$  be any irreducible polynomial of  $\mathbb{F}_2[D]$ . First we assume that  $p(D) | q(D)$ . Since (13) holds,  $p(D) \nmid f_i(D)$  for some  $i$ . Then we have both

$$\begin{aligned} e_{p(D)}(\mathbf{g}(D)) &= \min \left\{ e_{p(D)}(f_1(D)/q(D), \dots, f_c(D)/q(D)) \right\} = e_{p(D)}(f_i(D)/q(D)) \\ &= -e_{p(D)}(q(D)) < 0 \end{aligned} \quad (31)$$

and

$$p(D) \nmid \gcd(f_1(D), \dots, f_c(D)).$$

Now we assume that  $p(D) \nmid q(D)$ . Then

$$e_{p(D)}(\mathbf{g}(D)) = \min\{e_{p(D)}(f_1(D)), \dots, e_{p(D)}(f_c(D))\} \geq 0. \quad (32)$$

Thus,

$$\begin{aligned} e_{p(D)}(\mathbf{g}(D)) \leq 0 &\Leftrightarrow p(D) \nmid f_i(D) \text{ for some } i \\ &\Leftrightarrow p(D) \nmid \gcd(f_1(D), \dots, f_c(D)). \end{aligned} \quad (33)$$

Therefore,

$$\begin{aligned} e_p(D)(\mathbf{g}(D)) &\leq 0 \text{ for all irreducible polynomial } p(D) \\ \Leftrightarrow \gcd(f_1(D), \dots, f_c(D)) &= 1, \end{aligned} \quad (34)$$

which completes the proof.  $\square$

Forney [7] defined the *defect* of a  $1 \times c$  nonzero vector  $\mathbf{g}(D) = (g_1(D), \dots, g_c(D))$  over  $\mathbb{F}_2(D)$  to be

$$\text{def } \mathbf{g}(D) = - \sum_{p \in \mathcal{P}^*} e_p(\mathbf{g}(D)) \deg p. \quad (35)$$

When  $c = 1$ , i.e., when  $\mathbf{g}(D)$  reduces to a nonzero  $g(D) \in \mathbb{F}_2(D)$ , we have

$$\text{def } g(D) = - \sum_{p \in \mathcal{P}^*} e_p(g(D)) \deg p. \quad (36)$$

From the product formula it follows that for any nonzero  $g(D) \in \mathbb{F}_2(D)$ ,  $\text{def } g(D) = 0$ .

The following lemma shows the significance of  $\text{def } \mathbf{g}(D)$ .

**Lemma 4** Let  $\mathbf{g}(D) = (g_1(D), \dots, g_c(D))$  be a  $1 \times c$  nonzero encoding matrix over  $\mathbb{F}_2(D)$ . Write

$$g_i(D) = f_i(D)/q(D), \quad i = 1, \dots, c, \quad (37)$$

where  $f_i(D), q(D) \in \mathbb{F}_2[D]$ ,  $i = 1, \dots, c$ , and

$$\gcd(f_1(D), \dots, f_c(D), q(D)) = 1, \quad (38)$$

and assume that  $\mathbf{g}(D)$  is canonical. Then,

$$\text{def } \mathbf{g}(D) = \max\{\deg f_1(D), \dots, \deg f_c(D)\} \quad (39)$$

and  $\text{def } \mathbf{g}(D)$  is the generalized constraint length of  $\mathbf{g}(D)$ .

**Proof** We have

$$\begin{aligned} \text{def } \mathbf{g}(D) &= - \sum_{p \in \mathcal{P}^*} e_p(\mathbf{g}(D)) \deg p \\ &= -(e_{D-1}(\mathbf{g}(D)) + \sum_{p(D)|q(D)} e_{p(D)}(\mathbf{g}(D)) \deg p(D) \\ &\quad + \sum_{p(D) \nmid q(D)} e_{p(D)}(\mathbf{g}(D)) \deg p(D)) \\ &= -((\deg q(D) - \max\{\deg f_1(D), \dots, \deg f_c(D)\}) \\ &\quad - \sum_{p(D)|q(D)} e_{p(D)}(q(D)) \deg p(D) + 0), \end{aligned} \quad (40)$$

where in the last equality the first term follows from (29), the second term from (31), and the last term from (32) and Lemma 3'. The observation that

$$\deg q(D) = \sum_{p(D)|q(D)} e_{p(D)}(q(D)) \deg p(D) \quad (41)$$

and application of Lemma 3 complete the proof.  $\square$

**Example 2** Let

$$\mathbf{g}(D) = \begin{pmatrix} \frac{D}{1+D} & \frac{1+D}{1+D+D^2} & D^2 \end{pmatrix}. \quad (42)$$

By definition,

$$e_{1+D+D^2}(\mathbf{g}(D)) = \min\{0, -1, 0\} = -1. \quad (43)$$

Similarly,  $e_{1+D}(\mathbf{g}(D)) = -1$ ,  $e_D(\mathbf{g}(D)) = 0$ ,  $e_{D^{-1}}(\mathbf{g}(D)) = -2$ , and  $e_p(\mathbf{g}(D)) = 0$  if  $p \in \mathcal{P}^*$  and  $p \neq 1 + D + D^2, 1 + D, D, D^{-1}$ . It follows from Lemma 3' that  $\mathbf{g}(D)$  is canonical.

Clearly, we can express  $\mathbf{g}(D)$  as

$$\mathbf{g}(D) = \begin{pmatrix} \frac{D+D^2+D^3}{1+D^3} & \frac{1+D^2}{1+D^3} & \frac{D^2+D^5}{1+D^3} \end{pmatrix}, \quad (44)$$

which can be implemented by five delay elements in controller canonical form.  $\square$

The generalized constraint lengths of a canonical encoding matrix  $G(D)$  whose rows are  $\mathbf{g}_1(D), \dots, \mathbf{g}_b(D)$  were defined by Forney [9] as  $\text{def } \mathbf{g}_1(D), \dots, \text{def } \mathbf{g}_b(D)$ . Thus, by Lemma 4 his definition coincides with ours for such matrices. Our definition of the generalized constraint lengths in Section II is stated for more general rational encoding matrices and is more intuitive.

Let  $G(D)$  be a  $b \times c$  encoding matrix over  $\mathbb{F}_2(D)$  and let  $\mathcal{M}_b$  denote the set of all  $b \times b$  submatrices of  $G(D)$ . Forney [7] defined

$$e_p(G(D)) = \min\{e_p(|M_b(D)|) \mid M_b(D) \in \mathcal{M}_b\}, \text{ for any } p \in \mathcal{P}^* \quad (45)$$

and the *defect* of the encoding matrix  $G(D)$  to be [9]

$$\text{def } G(D) = - \sum_{p \in \mathcal{P}^*} e_p(G(D)) \deg p. \quad (46)$$

He also proved the following important result [9]:

**Theorem 5** *The defect  $\text{def } G(D)$  is an invariant of the convolutional code  $\mathcal{C}$  that is encoded by  $G(D)$ .*

**Proof** Let  $T(D)$  be a  $b \times b$  nonsingular matrix over  $\mathbb{F}_2(D)$ . Then

$$e_p(|T(D)M_b(D)|) = e_p(|T(D)|) + e_p(|M_b(D)|). \quad (47)$$

Hence,

$$e_p(T(D)G(D)) = e_p(|T(D)|) + e_p(G(D)). \quad (48)$$

It follows from (36), (46), and (48) that

$$\text{def } (T(D)G(D)) = \text{def } |T(D)| + \text{def } G(D). \quad (49)$$

But  $|T(D)| \in \mathbb{F}_2(D)$  and  $|T(D)| \neq 0$ . By the product formula (24) and (36),  $\text{def } |T(D)| = 0$ . Hence,

$$\text{def}(T(D)G(D)) = \text{def } G(D). \quad (50)$$

□

Theorem 5 motivated Forney [9] to introduce the *defect* of the code  $\mathcal{C}$  encoded by the encoding matrix  $G(D)$  to be

$$\text{def } \mathcal{C} = \text{def } G(D). \quad (51)$$

## V The Invariance of the Generalized Constraint Lengths

In [11] we gave a proof of the invariance of the constraint lengths of minimal-basic encoding matrices which is equivalent to a classical result of Kronecker's (cf. [14]). This result was also proved by Forney [7]. By a straightforward generalization of our proof we show

**Theorem 6** *The generalized constraint lengths of two equivalent canonical encoding matrices are equal one by one up to a rearrangement.*

**Proof** Let  $\mathcal{C}$  be the code encoded by two equivalent canonical encoding matrices  $G(D)$  and  $G'(D)$  with generalized constraint lengths  $\nu_1, \dots, \nu_b$  and  $\nu'_1, \dots, \nu'_b$ , respectively. Without loss of generality we assume that  $\nu_1 \leq \dots \leq \nu_b$  and  $\nu'_1 \leq \dots \leq \nu'_b$ .

Now suppose that  $\nu_i$  and  $\nu'_i$  are not equal for all  $i$ ,  $1 \leq i \leq b$ . Let  $j$  be the smallest index such that  $\nu_j \neq \nu'_j$ . Then without loss of generality we assume that  $\nu_j < \nu'_j$ . It is well-known in linear algebra that from the sequence  $\mathbf{g}_1, \dots, \mathbf{g}_j, \mathbf{g}'_1, \dots, \mathbf{g}'_b$  we can obtain a basis  $\mathbf{g}_1, \dots, \mathbf{g}_j, \mathbf{g}'_{i_{j+1}}, \dots, \mathbf{g}'_{i_b}$  of  $\mathcal{C}$  (cf. Theorem 4 on p. 169 of [15]). These  $b$  row vectors form an encoding matrix  $G''(D)$  which is equivalent to  $G'(D)$ . Let

$$\{\mathbf{g}'_1, \dots, \mathbf{g}'_b\} \setminus \{\mathbf{g}'_{i_{j+1}}, \dots, \mathbf{g}'_{i_b}\} = \{\mathbf{g}'_{i_1}, \dots, \mathbf{g}'_{i_j}\}. \quad (52)$$

From our assumptions it follows that

$$\nu'' = \sum_{l=1}^j \nu_l + \sum_{l=j+1}^b \nu'_{i_l} < \sum_{l=1}^j \nu'_l + \sum_{l=j+1}^b \nu'_{i_l} \leq \sum_{l=1}^j \nu'_{i_l} + \sum_{l=j+1}^b \nu'_{i_l} = \nu', \quad (53)$$

where  $\nu'$  and  $\nu''$  are the generalized overall constraint lengths of the encoding matrices  $G'(D)$  and  $G''(D)$ , respectively. The inequality (53) contradicts the assumption that  $G'(D)$  is canonical. This completes the proof. □

In virtue of Theorem 6 we may define the generalized constraint lengths of a convolutional code to be the generalized constraint lengths of any canonical encoding matrix that encodes the code. By Theorem 1, a minimal-basic encoding matrix is canonical. Thus, we have

**Theorem 7** *The generalized constraint lengths of a convolutional code are equal to the constraint lengths of any minimal-basic encoding matrix that encodes the code one by one up to a rearrangement.*

## VI Conditions for an Encoding Matrix to be Canonical

Before we can give four equivalent conditions for an encoding matrix to be canonical we shall prove a lemma. Let

$$G(D) = (g_{ij}(D))_{1 \leq i \leq b, 1 \leq j \leq c} \quad (54)$$

be an encoding matrix, where  $g_{ij}(D) \in \mathbb{F}_2(D)$ . Write

$$\mathbf{g}_i(D) = (g_{i1}(D), \dots, g_{ic}(D)), i = 1, \dots, b \quad (55)$$

$$g_{ij}(D) = f_{ij}(D)/q_i(D), i = 1, \dots, b; j = 1, \dots, c, \quad (56)$$

where  $f_{ij}(D), q_i(D) \in \mathbb{F}_2[D]$ ,  $i = 1, \dots, b$ ;  $j = 1, \dots, c$ , and assume that

$$\gcd(f_{i1}(D), \dots, f_{ic}(D), q_i(D)) = 1, \quad i = 1, \dots, b. \quad (57)$$

When we study polynomial encoding matrices, the  $b \times c$  (0,1)-matrix  $[G(D)]_h$  with 1 in the position  $(i, j)$  where  $\deg g_{ij}(D) = \nu_i$  and 0 otherwise plays an important role [5], [11]. As a counterpart, when we study rational encoding matrices, for each  $p \in \mathcal{P}^*$  we introduce the  $b \times c$  matrix  $[G(D)]_h(p)$  to be a matrix whose element in the position  $(i, j)$  is equal to the coefficient of the lowest term of  $g_{ij}(D)$ , written as a Laurent series of  $p$ , if  $e_p(g_{ij}(D)) = e_p(\mathbf{g}_i(D))$ , and equal to 0, otherwise. Then define  $G_1(D, p)$  by

$$G_1(D, p) = \begin{pmatrix} p^{e_p(g_{11}(D))} & & \\ & \ddots & \\ & & p^{e_p(g_{bc}(D))} \end{pmatrix} [G(D)]_h(p) \quad (58)$$

and  $G_0(D, p)$  by

$$G(D) = G_0(D, p) + G_1(D, p). \quad (59)$$

From (59) and (58) we have

**Lemma 8** *Let  $G(D)$  be a  $b \times c$  rational encoding matrix and let  $p \in \mathcal{P}^*$ . Then*

- (i)  $e_p([G(D)]_h(p)) = 0$  if and only if  $e_p(G(D)) = \sum_{i=1}^b e_p(\mathbf{g}_i(D))$ .
- (ii)  $e_p([G(D)]_h(p)) \neq 0$  if and only if  $e_p(G(D)) > \sum_{i=1}^b e_p(\mathbf{g}_i(D))$ .

□

We are now well prepared to prove the following

**Theorem 9** *Let  $G(D)$  be a  $b \times c$  rational encoding matrix with rows  $\mathbf{g}_1(D), \dots, \mathbf{g}_b(D)$ . Then the following statements are equivalent<sup>2</sup>:*

- (i)  $G(D)$  is a canonical encoding matrix.
- (ii) For all  $p \in \mathcal{P}^*$ :  $e_p(\mathbf{g}_i(D)) \leq 0$ ,  $1 \leq i \leq b$ , and  $e_p([G(D)]_h(p)) = 0$ .
- (iii) For all  $p \in \mathcal{P}^*$ :  $e_p(\mathbf{g}_i(D)) \leq 0$ ,  $1 \leq i \leq b$ , and  $e_p(G(D)) = \sum_{i=1}^b e_p(\mathbf{g}_i(D))$ .
- (iv) For all  $p \in \mathcal{P}^*$ :  $e_p(\mathbf{g}_i(D)) \leq 0$ ,  $1 \leq i \leq b$ , and  $\text{def } G(D) = \sum_{i=1}^b \text{def } \mathbf{g}_i(D)$ .
- (v) For all  $p \in \mathcal{P}^*$ :  $e_p(\mathbf{g}_i(D)) \leq 0$ ,  $1 \leq i \leq b$ , and  $[G(D)]_h(p) \bmod p$  is of full rank.

---

<sup>2</sup>The equivalence of (i) and (iii) was proved by Forney [9].

**Proof** (i  $\Rightarrow$  ii). Assume that  $G(D)$  is canonical. Suppose that  $e_p(\mathbf{g}_i(D)) \leq 0$  does not hold for some  $p \in \mathcal{P}^*$  and some  $i$ , then, by Lemma 3',  $\mathbf{g}_i(D)$  is not canonical, and, hence,  $G(D)$  is not canonical.

Suppose that  $e_p([G(D)]_h(p)) = 0$  does not hold for some  $p \in \mathcal{P}^*$ . Then, by Lemma 8, for any  $p \in \mathcal{P}^*$  such that  $e_p([G(D)]_h(p)) = 0$  does not hold, we have

$$e_p(G(D)) > \sum_{i=1}^b e_p(\mathbf{g}_i(D)) \quad (60)$$

and for any  $p \in \mathcal{P}^*$  such that  $e_p([G(D)]_h(p)) = 0$  holds, we have

$$e_p(G(D)) = \sum_{i=1}^b e_p(\mathbf{g}_i(D)). \quad (61)$$

Thus, by combining (60) and (61) we obtain

$$\begin{aligned} \text{def } G(D) &= - \sum_{p \in \mathcal{P}^*} e_p(G(D)) \deg p < - \sum_{p \in \mathcal{P}^*} \sum_{i=1}^b e_p(\mathbf{g}_i(D)) \deg p \\ &= \sum_{i=1}^b \left( - \sum_{p \in \mathcal{P}^*} e_p(\mathbf{g}_i(D)) \deg p \right) = \sum_{i=1}^b \text{def } \mathbf{g}_i(D). \end{aligned} \quad (62)$$

Hence,  $G(D)$  is not canonical.

(ii  $\Rightarrow$  iii). Follows from Lemma 8.

(iii  $\Rightarrow$  iv). Follows from (35) and (46).

(iv  $\Rightarrow$  i). By Lemma 4, the hypothesis means that  $\text{def } G(D)$  is the generalized overall constraint length of  $G(D)$ . Let  $G_c(D)$  be a canonical encoding matrix equivalent to  $G(D)$ . Then, from Theorem 5 it follows that  $\text{def } G_c(D) = \text{def } G(D)$ . By (i  $\Rightarrow$  iv),  $\text{def } G_c(D)$  is the overall constraint length of  $G_c(D)$ . Thus,  $\text{def } G_c(D)$  is minimum over all equivalent encoding matrices, and so is, of course,  $\text{def } G(D)$ . Hence,  $G(D)$  is canonical.

(ii  $\Leftrightarrow$  v).  $e_p([G(D)]_h(p)) = 0$  means that there exists at least one  $b \times b$  minor of  $[G(D)]_h(p)$  not divisible by  $p$ , which completes the proof.  $\square$

**Example 3** Consider the rational encoding matrix

$$G(D) = \begin{pmatrix} 1 & \frac{D}{1+D} & \frac{1}{1+D} \\ \frac{D^2}{1+D+D^2} & \frac{1}{1+D+D^2} & 1 \end{pmatrix}. \quad (63)$$

Clearly, for all  $p \in \mathcal{P}^*$ ,  $e_p(\mathbf{g}_i(D)) \leq 0$ ,  $i = 1, 2$ . Moreover, we have  $\text{def } \mathbf{g}_1(D) = 1$ ,  $\text{def } \mathbf{g}_2(D) = 2$ , and  $\text{def } G(D) = 3$ . Therefore condition (iv) in Theorem 9 is satisfied and we conclude that  $G(D)$  is canonical and, hence, minimal.  $\square$

**Example 4** The rational encoding matrix (cf. [11])

$$G(D) = \begin{pmatrix} 1 & 0 & \frac{1+D^2}{1+D+D^2} & \frac{D^2}{1+D+D^2} \\ 0 & 1 & \frac{D^2}{1+D+D^2} & \frac{1}{1+D+D^2} \end{pmatrix} \quad (64)$$

is systematic and, hence, minimal. We have  $\text{def } \mathbf{g}_1(D) = 2$ ,  $\text{def } \mathbf{g}_2(D) = 2$ , and  $\text{def } G(D) = 2$ . Thus,

$$\text{def } G(D) \neq \sum_{i=1}^2 \text{def } \mathbf{g}_i(D). \quad (65)$$

From (65) and Theorem 9 we conclude that  $G(D)$  is *not* canonical although it is minimal.  $\square$

**Corollary 10** *Let  $\mathcal{C}$  be a convolutional code. Then there is a canonical encoding matrix of  $\mathcal{C}$  whose generalized overall constraint length is  $\text{def } \mathcal{C}$ . Moreover, the number of memory elements in any encoder of  $\mathcal{C}$  is  $\geq \text{def } \mathcal{C}$ .*  $\square$

Corollary 10 is due to Forney [9] who proved it using the realization theory of linear systems.

## VII Conclusion and Acknowledgement

Among the rational encoding matrices that encode a convolutional code we have singled out the class of canonical encoding matrices, which can be realized by the least number of delay elements in controller canonical form among all equivalent encoding matrices. Thus the position of canonical encoding matrices within the class of rational encoding matrices corresponds to that of minimal-basic encoding matrices within the class of polynomial encoding matrices. We have shown that the set of canonical encoding matrices is a proper subset of the set of minimal rational encoding matrices. This is a generalization of our previous result that the set of minimal-basic encoding matrices is a proper subset of the set of minimal polynomial encoding matrices [10], [11].

Needless to say, we would not have been able to obtain our results if Forney [9] had not made things smooth for us.

## References

- [1] J. L. Massey and M. K. Sain, “Codes, automata, and continuous systems: Explicit interconnections”, *IEEE Trans. Autom. Control*, AC-12:644–650, 1967.
- [2] J. L. Massey and M. K. Sain, “Inverses of linear sequential circuits”, *IEEE Trans. Comput.*, C-17:330–337, 1968.
- [3] R. R. Olson, “Note on feedforward inverses for linear sequential circuits”, *IEEE Trans. Comput.*, C-19:1216–1221, 1970.
- [4] D. J. Costello, Jr., “Construction of convolutional codes for sequential decoding”, Techn. Rpt. EE-692, U. Notre Dame, 1969.
- [5] G. D. Forney, Jr., “Convolutional codes I: Algebraic structure”, *IEEE Trans. Inform. Theory*, IT-16:720–738, 1970.

- [6] G. D. Forney, Jr., “Structural analyses of convolutional codes via dual codes”, *IEEE Trans. Inform. Theory*, IT-19:512–518, 1973.
- [7] G. D. Forney, Jr., “Minimal bases of rational vector spaces, with applications to multivariable systems”, *SIAM J. Control*, 13:493–520, 1975.
- [8] P. Piret, *Convolutional Codes: An Algebraic Approach*, MIT Press, Cambridge, Mass, 1988.
- [9] G. D. Forney, Jr., “Algebraic structure of convolutional codes, and algebraic system theory”, *Mathematical System Theory*, A.C. Antoulas, Ed., Springer-Verlag, Berlin, 527–558, 1991.
- [10] R. Johannesson and Z. Wan, “Submodules of  $F[x]^n$  and convolutional codes”, Proceedings of the First China-Japan International Symposium on Ring Theory, Oct. 20–25, 1991, Guilin, China, 1991.
- [11] R. Johannesson and Z. Wan, “A linear algebra approach to minimal convolutional encoders”, *IEEE Trans. Inform. Theory*, IT-39:1219–1233, 1993.
- [12] H.-A. Loeliger and T. Mittelholzer, “Convolutional codes over groups”. Submitted to *IEEE Trans. Inform. Theory*, 1992.
- [13] N. Jacobson, *Basic Algebra II*, 2nd ed., Freeman, New York, 1989.
- [14] T. Kailath, *Linear systems*, Prentice Hall, Englewood Cliffs, N.J., 1980.
- [15] G. Birkhoff and S. MacLane, *A Survey of Modern Algebra*, rev. ed., MacMillan, New York, 1953.

# A Comparison Of Error Patterns Corrected By Block Codes And Convolutional Codes

J. Justesen

Institute of Circuit Theory and Telecommunications  
Technical University of Denmark

## Abstract

The purpose of this paper is to suggest that when a fair comparison of a block code and a convolutional code is made, which means that good codes with the same rate and distance are compared, the performance is almost the same. However, the convolutional code has a small advantage which is directly related to the structure of the code.

## I Introduction

The possible advantages of convolutional codes over block codes have frequently been suggested. At times Jim Massey might jokingly claim that anything which could be done with a block code could be done better with a convolutional code. However, colleagues and students were often reminded about the need for precision in the definition and analysis of convolutional codes and their distances [1].

The distance properties of convolutional codes and their relation to correctable errors is a much more complicated subject than the distances of block codes. A summary of distances and distance bounds for convolutional codes was given by Massey in [2], and this paper also indicates that more results in this direction were being developed. What was missing in this early work was mainly the notion of a trellis. However, when this concept was introduced, and a nice algebraic approach based on finite state machines was described by Viterbi [3], it was not used much for studying distances of specific codes.

With the introduction of the Viterbi algorithm, the number of states became a commonly used measure of complexity of decoding. If trellis decoding is also used for decoding block codes, it becomes possible to make a comparison of the complexities of different codes with the same distance. The paper by Zyablov and Sidorenko [4] indicates that convolutional codes are in some sense optimal.

The present analysis was at first related to Unit Memory convolutional (UM) codes. In our work on the distances of such codes [5], we found that although the approach was not in principle different from the usual analysis of convolutional codes, distance properties of specific UM codes were often much more easily understood. For this reason a UM code will be used for the example in this paper.

The result of this paper is that when a fair comparison of a block code and a convolutional code is made, which means that good codes with the same rate and distance are compared, the performance is almost the same. However, the convolutional code has a small advantage which is directly related to the structure of the code.

## II Correctable Error Patterns for Block Codes and Convolutional Codes.

In this section we describe the sets of error patterns that can be corrected by block codes and by convolutional codes. In addition we introduce the intersection of these sets, which will be called the basic set of error patterns. As a first approximation, the performance is related to the correction of errors in the basic set, which is common to the two classes of codes.

In the analysis of block codes, the received symbols are usually described as a random vector of length equal to the blocklength of the code. This is a natural approach when the transmitted blocks are mutually independent and the channel is memoryless. However in order to compare the two code structures, we must in both cases consider long sequences of received symbols. As a result the set of correctable errors will depend on a random segmentation of the sequence into blocks.

*Example 1.* In the examples we shall use the  $(24, 12, 8)$  block code. In order to simplify the discussion, we shall consider the received bits to be organized in 8-bit bytes. The code will correct three errors in any three bytes. The remaining 1771 syndromes could be used for correcting some errors of weight 4, but all these cosets contain more than one such error pattern, and we shall not consider decoding in these cases. When the received sequence is considered to consist of 8-bit bytes with a random distribution of errors, the correction of a particular error pattern may depend on the segmentation of the sequence into 24-bit blocks. We shall assume the segmentation to start in any particular byte with probability  $1/3$ , and with this distribution we shall calculate the number of error patterns of a given weight that the code will correct. In a single byte all patterns with at most three errors are corrected. In two bytes we can always correct three errors, but with probability  $1/3$  the bytes belong to different blocks, and three errors can be corrected in each byte. Similarly three bytes belong to a single block with probability  $1/3$  and are separated into two blocks with probability  $2/3$ . Four bytes always belong to two blocks. We shall consider a set of errors that can be corrected for any segmentation, and this will turn out to be the basic set

of correctable errors. This set is defined by the number of errors in any window of length  $j$  bytes. For the code considered, an error pattern is in the set if and only if the weight in any window is at most 3 for  $j \leq 3$  and at most  $j$  for any length  $j > 3$ . These error patterns are not the only ones that can always be corrected; the block code will correct any error pattern such that  $j$  bytes contain at most  $3 \times Ij/3J$  errors. Both sets include all error patterns of weight 3 and all error patterns with at most four errors in four bytes but only three errors in three bytes. However, for any particular segmentation there are error patterns of weight 4, 5, and 6 in two or three bytes that can be corrected. If three bytes are segmented into two blocks, four errors can be corrected if they are not in the same block. Thus the average number of error patterns of weight 4 in three blocks is  $(2/3)(4480+3360+896)=5824$ .

*Example 2.* In this example we consider the (8,4) UM code with memory 3 (bits). The row distances of this code are  $\max\{8, 2j + 4\}$  for  $j$  encoded blocks (this could be  $j$  or  $j - 1$  information blocks). Thus the number of errors that can be corrected in  $j$  received blocks is  $t(j) = \max\{3, j + 1\}$  in  $j$  blocks. If the number of errors in any window of length  $j$  is at most  $t(j)$ , there is a unique closest transmitted sequence. Comparing this restriction to the sets discussed in the previous example, we find that the basic set of error patterns, the set that both codes correct, is the one we described earlier. However, the convolutional code will accept an additional error for  $j > 2$ . In particular four errors in three bytes are corrected if there are at most three errors in any window of two bytes. The number of such error patterns is 7056. The better performance of the UM code is directly related to the fact that it corrects more error patterns of weight 4.

Considering that we have chosen an almost perfect block code for the comparison, it may seem surprising that any code can correct more errors. However, the use of long received sequences rather than a random received vector changes the picture somewhat. When an error pattern with the errors concentrated on a small number of bytes is considered, the memory of the UM code introduces three additional syndrome bits, which account for the extra error correcting capacity. In the block code there is also an ability to correct more errors, but here it is related to the random segmentation of the error pattern, which may introduce an extra block and thus additional syndromes.

Clearly most of the received sequences will be in the basic set, and thus to a first approximation the performance of the two codes will be equal. However, in a more accurate comparison the performance of the UM codes turns out to be slightly better due to the fact that the other correctable error patterns have higher probability than the patterns corrected by the block code.

The examples are greatly simplified by the assumption that the sequences are synchronized in bytes. If the sequence is randomly segmented into bytes, the average number of correctable error patterns will change, but the effect is the same for both codes.

It is not possible to make a precise comparison of the complexity of decoding the two codes. Several different methods are available with different trade-offs between table sizes and the number

of simple logic operations. The differences are in all cases small, but the UM code has a slight advantage unless the variable decoding delay is considered to be a serious disadvantage.

The codes of Examples 1 and 2 were chosen to have the same basic set of error patterns. For other parameter values it may be difficult to find a pair of codes with matching distances. However, a comparison with bounds for long codes indicates that the examples reflect the typical situation.

Consider a long  $(N, Nr, D)$  block code with rate  $r$  and  $D = NH - 1(1-r)$ , i.e. an average code with minimum distance given by the Varshamov-Gilbert bound. For a long  $(n, rn)$  convolutional code with the same rate, the row distance [5] is

$$d_{j+M} = n(j + M)H^{-1}(1 - rj/(j + M)) \quad (1)$$

where  $M$  is the memory of the code measured in blocks (the index  $j + M$  refers to  $j$  input blocks and  $j + M$  output blocks). For large values of  $j$ , this function converges to a straight line with slope  $H^{-1}(1 - r)$ . This line intersects the  $d$ -axis in the point  $(0, nMb)$ , where  $b$  is a function of  $r$  with positive values. As an approximation we may take

$$d_{j+M} \geq \max\{d_{free}, jnH^{-1}(1 - r) + b\} \quad (2)$$

By taking suitable values of  $N$  and  $M$  we can make  $D = d_{free}$ . In this case we get the same basic set of error patterns for the block code and the convolutional code. However, due to the positive value of  $b$ , additional errors can be corrected in sequences of length  $jn$  whenever the row distance is greater than  $d_{free}$ .

*Example 3.* For long codes of rate 1/2, the number of errors in the basic set is  $njH^{-1}(1/2)/2 = nj \times 0.055$ . The number  $b$  is 0.28 , i.e. for a given memory it is not important whether the convolutional code is a unit memory or multi memory code. However, the free distance depends on the structure of the code, and we shall assume a unit memory code with free distance .44n. Thus if  $N = 4n$ , we get  $D = d_{free}$ . In this case, it follows from Equation 2 that the convolutional code corrects all error patterns with less than 0.22n errors in one or two subblocks and  $n(0.055j + 0.085)$  errors in  $j$  subblocks,  $j > 2$ . The block code always corrects 0.22n errors in four subblocks, and heavier error patterns will be corrected if they do not occur within a single block. If the bit error probability is small, the probability of decoding failure depends on the number of errors of weight slightly greater than 0.22n which are not corrected. For the convolutional code these errors have length at most 3n, while errors of length 4n cause errors in the block code.

Clearly the conclusion in Example 3 is also true for other rates because a positive value of  $b$  always makes the low weight error patterns shorter for the convolutional code. The expurgated bound on the error exponent is directly related to the minimum distance, and we may therefore expect codes with equal minimum distance to have approximately the same performance on good channels. However, for an average code, the number of codewords of weight slightly greater than

the minimum distance is approximately

$$2^{-N(H(d/N)-(1-R))}$$

for a code of rate  $R$  and blocklength  $N$ . It follows from the convexity of the entropy function that this number increases with increasing values of  $N$ . Thus even though the block code and the critical terminated code in the convolutional code have equal distances, the number of low-weight codewords in the convolutional code is smaller and thus the performance may be expected to be slightly better.

We have considered long average codes in order to demonstrate that for codes with equal distances, the number of low-weight error patterns that cause decoding errors is typically smaller for convolutional codes. At this time there is no realistic way of decoding long codes up to these distances, and for this reason it is not really possible to discuss the decoding complexity. The minimum distances and the corresponding expurgated error bounds for unit memory codes were discussed in [6].

### III Computational Methods and Examples

In Example 2 we had to calculate the number of sequences of length  $J$  blocks which satisfy the property that the number of errors in any  $j$  consecutive blocks is at most  $t(j)$ . Even when  $t(j)$  is a simple function of  $j$  (other than  $aj$  for some constant  $a$ ), it is not a standard combinatorial problem to find this number. However, it may be calculated by a modification of the algorithm used in [5] for deciding if a given sequence satisfies the system of weight constraints.

For some UM codes the number of correctable errors in  $j$  subblocks may be expressed as

$$t(j) = aj + b/2, \quad j > 1.$$

In this case the states of the algorithm are defined by the number

$$f(k) = \max\{w[e(k)], f(k-1) - a + w[e(k)]\}$$

where  $w[e(k)]$  indicates the weight of the error pattern in subblock  $k$ . For each value of  $f(k)$  less than or equal to  $a + b/2$ , the number of error patterns of each weight may be calculated from the numbers at depth  $k$  and the number of transitions. A value of  $f$  greater than  $a + b/2$  indicates that some window contains more than  $aj + b/2$  errors, which is allowed only for  $j = 1$ .

If the linear approximation is valid only for longer sequences, additional states are needed in the algorithm.

*Example 4.* For the code discussed in Example 2, the weight constraint is checked by calculating

$$f(k) = \max\{w[e(k)], f(k-1) - 1 + w[e(k)]\}$$

where  $f(k) < 3$  with the exception that a byte of weight 3 may occur when  $f(k - 1) = 0$ .

The weight constraint for the basic set is checked by a similar algorithm, but slightly more memory is needed in this case. The calculation of number of error patterns corrected by the block code is straightforward.

The number of corrected error patterns in the basic set is compared to the errors corrected by the block code and the convolutional code in the following table:

Basic set

| Weight | Number of blocks |      |       |        |        |
|--------|------------------|------|-------|--------|--------|
|        | 2                | 3    | 4     | 5      | 6      |
| 1      | 16               | 24   | 32    | 40     | 48     |
| 2      | 120              | 276  | 496   | 780    | 1128   |
| 3      | 560              | 2024 | 4960  | 9880   | 17296  |
| 4      | 0                | 0    | 16528 | 63152  | 157536 |
| 5      | 0                | 0    | 0     | 160448 | 777664 |

UM Code

| Weight | Number of blocks |      |       |        |         |
|--------|------------------|------|-------|--------|---------|
|        | 2                | 3    | 4     | 5      | 6       |
| 1      | 16               | 24   | 32    | 40     | 48      |
| 2      | 120              | 276  | 496   | 780    | 1128    |
| 3      | 560              | 2024 | 4960  | 9880   | 17296   |
| 4      | 0                | 7056 | 30640 | 84320  | 185760  |
| 5      | 0                | 0    | 92736 | 455232 | 1387392 |

Block Code

| Weight | Number of blocks |       |       |        |         |
|--------|------------------|-------|-------|--------|---------|
|        | 2                | 3     | 4     | 5      | 6       |
| 1      | 16               | 24    | 32    | 40     | 48      |
| 2      | 120              | 276   | 496   | 780    | 1128    |
| 3      | 560              | 2024  | 4960  | 9880   | 17296   |
| 4      | 560              | 5824  | 27616 | 79527  | 179782  |
| 5      | 1045             | 14933 | 92885 | 412690 | 1293778 |

## IV Conclusions

The minimum distance is often a good measure of the performance of a code (Concatenated codes are important exceptions, since minimum weight error events are extremely rare). As a first approximation we argue that block codes and convolutional codes with the same distances have approximately the same performance, because both correct the same basic set of errors. However, due to the difference in structure, the number of low-weight errors that cause decoding failure is

smaller for a convolutional code than for a block code. The difference is most easily demonstrated by calculating the number of error patterns that cause decoding failure when minimum distance decoding is used for a block code and a unit memory code with the same rate and minimum distance. For short codes, the decoding complexities are often comparable for codes with the same distance.

## References

- [1] J.L. Massey, "Theory and practice of error control codes", (book review), *IEEE Trans. Info. Th.*, IT-31, pp. 553-554, 1985.
- [2] J.L. Massey, *Some algebraic and distance properties of convolutional codes*, in H.B. Mann (Ed.): *Error Correcting Codes*, pp. 89-110, Wiley, New York, 1968.
- [3] A.J. Viterbi, "Convolutional codes and their performance in communication systems", *IEEE Trans. Commun. Tech.*, COM-19, pp. 751-772, 1971.
- [4] V.V. Zyablov and V.R. Sidorenko, "Bounds on the complexity of trellis decoding of linear block codes", to appear in *Problemy Peredachi Informatsii*.
- [5] J. Justesen, "Bounded distance decoding of unit memory codes", *IEEE Trans. Info. Th.*, IT-39, 1993.
- [6] C. Thommesen and J. Justesen, "Bounds on distances and error exponents of unit memory codes", *IEEE Trans. Info. Th.*, IT-29, pp. 637-649, 1983.

# Encounters with the Berlekamp-Massey Algorithm \*

Thomas Kailath †

## Abstract

In 1969, J. Massey published a now-famous paper showing, among other things, that an iterative algorithm introduced by Berlekamp for decoding BCH codes also solved the problem of finding a shortest-length feedback shift register circuit for generating a given finite sequence of digits. This nice physical interpretation opened the door to connections with many other problems, including the minimal partial realization problems of linear system theory, Padé approximations and continued fractions, the fast algorithms of Levinson and Schur for Toeplitz matrices, inverse scattering, VLSI implementations, etc. This paper is an informal account of some of the different contexts in which the Berlekamp-Massey algorithm have been encountered in the work of the author and his students.

## I Introduction

It is a pleasure to have this opportunity to reflect in an informal way on my several personal and professional encounters with Jim Massey, beginning with overlapping graduate student tenures at MIT, overlapping consultancies with the affectionately remembered Art Kohlberg of Melpar, Inc., and various later contacts at Notre Dame, UCLA and ETH. However, I only began to follow some of Jim's technical work more closely when in the early seventies I got interested in linear system theory and noticed the studies of Massey and Sain, inspired by problems in convolutional codes, on the inversion of multivariable systems. Of course various connections between coding and systems are evident, but it waited for Jim to take the first deep steps in bringing them together. (Since then, Dave Forney has exploited and extended the connections in striking fashion, making important contributions to both fields.) Jim's best known contribution in this area is the Berlekamp-Massey algorithm for decoding BCH codes.

Over the years I have had various encounters with this algorithm, which has turned out to have interesting connections with a number of things rather different from algebraic coding theory, such as fast algorithms for recursive Hankel matrix factorization, (generalized) Lanczos recursions and Lanczos orthogonal polynomials, Levinson and Schur algorithms, inverse scattering for certain generalized transmission lines, parallelizable forms for VLSI implementation and others.

I remember the excitement at MIT in 1960 when in perhaps the first university course on algebraic coding theory, Wes Peterson described Reed-Solomon codes, and their striking

---

\*This work was supported in part by the Army Research Office under Grant DAAH04-93-G-0029.

†Dept. of Electrical Engineering, Stanford University, Stanford, CA 94305-4055.

independent generalization by Bose and Chaudhuri in the U.S. and by Hocquenghem in France. [I also remember Gus Solomon coming in as a substitute lecturer one day and introducing himself as N. Wiener; in the shocked silence, he explained that ‘N’ was for Nathan. For a reference on some classical result, the answer was *Acta Pyramidica*. Coding theory has had a rich cast of characters.] It was already clear at that time that the real problem was in the decoding and Peterson (1960) was the first to give a reasonably efficient procedure, requiring  $\mathcal{O}(t^3)$  multiplications (on a serial machine) for decoding a  $t$ -error correcting BCH code. Then in his remarkable book, Berlekamp (1968) presented a procedure requiring only  $\mathcal{O}(t^2)$  multiplications, based on solving what he called the *key equation*:

$$S(z)\lambda(z) = \omega(z) \pmod{z^{2t}} \quad (1)$$

where  $S(z)$  is a given polynomial formed from the syndromes  $\{S_j, j = 1, \dots, 2t\}$  as

$$S(z) = \sum_{j=1}^{2t} S_j z^{j-1} \quad (2)$$

and the error-location polynomial  $\lambda(z)$  and the error-evaluator polynomial  $\omega(z)$  are to be determined as the unique coprime polynomials of degree  $\leq t$  and  $\leq t-1$ , respectively. This may be all too cryptic for non-coding-theorists, but will be all we need to know for later discussions. In fact, it was Jim Massey who reinterpreted the key equation in more widely accessible linear systems language, and used the circuits intuition to add some refinements (Massey, 1969). The final algorithm is now universally referred to as the Berlekamp-Massey Algorithm (BMA) and has been very widely studied since then, with papers on it continuing to appear quite regularly; an INSPEC search showed 11 papers in 1989-1992 with BMA in their titles.

## II Massey’s Interpretation

To obtain a more physical picture, we introduce  $z^{-1}$  as the usual unit-time-delay operator of discrete-time system theory (Jim used  $D$ , following a notation introduced by Dave Huffman in studying sequential circuits) and rewrote the key equation as

$$S(z^{-1})\lambda(z^{-1}) = \omega(z^{-1}) + a(z^{-1})z^{-2t}$$

or

$$\begin{aligned} z^{-1}S(z^{-1}) &= \frac{z^{-1}\omega(z^{-1})}{\lambda(z^{-1})} + b(z^{-1})z^{-2t} \\ &= \frac{\omega^\#(z)}{\lambda^\#(z)} + b(z^{-1})z^{-2t} \end{aligned} \quad (3)$$

where

$$\begin{aligned} \lambda^\#(z) &= \text{the reciprocal polynomial of } \lambda(z) \\ &= z^{\deg \lambda} \lambda(z^{-1}) \end{aligned}$$

and similarly for  $\omega^\#(z)$ .

In linear systems language,  $\omega^\#(z)/\lambda^\#(z)$  is the transfer function of a linear system that when excited by an impulse gives as a response the sequence  $\{0, S_1, \dots, S_{2t}, \dots\}$ . Therefore, solving the key equation is equivalent to finding a minimal order rational function the first  $2t$  terms of whose impulse response are equal to the syndrome sequence. Over the binary field, this is now a feedback shift register (FSR) realization problem.

With the physical problem in mind, Massey was able to recast and refine Berlekamp's solution to yield the following result.

### Berlekamp-Massey Algorithm

*Consider the following slight rearrangement of the key equation,*

$$(1 + zS(z))\lambda(z) = \bar{\omega}(z) \bmod z^{2t+1} \quad (4)$$

where  $\bar{\omega}(z) = \omega(z) + \lambda(z)$ . Then we can find

$$\bar{\omega}(z) = \bar{\omega}^{(2t)}(z), \quad \lambda(z) = \lambda^{(2t)}(z),$$

by running the following recursions: for  $k = 1$  to  $2t$ ,

$$\begin{bmatrix} \bar{\omega}^{(k)}(z) \\ v^{(k)}(z) \end{bmatrix} = \Gamma_k \begin{bmatrix} \bar{\omega}^{(k-1)}(z) \\ v^{(k-1)}(z) \end{bmatrix}, \quad \begin{bmatrix} \bar{\omega}^{(0)}(z) \\ v^{(0)}(z) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and

$$\begin{bmatrix} \lambda^{(k)}(z) \\ \tau^{(k)}(z) \end{bmatrix} = \Gamma_k \begin{bmatrix} \lambda^{(k-1)}(z) \\ \tau^{(k-1)}(z) \end{bmatrix}, \quad \begin{bmatrix} \lambda^{(0)}(z) \\ \tau^{(0)}(z) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

where

$$\Gamma_k = \begin{bmatrix} 1 & -z\Delta_k \\ \Delta^{-1}\gamma_k & (1 - \gamma_k)z \end{bmatrix}, \quad (5)$$

$$\Delta_k = \sum_{l=0}^t S_{k-l} \lambda_l^{(k-1)}, \quad (6)$$

Also  $\gamma_k$  is a control variable, equal to 0 or 1, determined as follows: Let

$$\bar{N}_0 = 0, \quad N_k = \bar{N}_{k-1} + 1,$$

where

$$\bar{N}_k = \begin{cases} -N_k, & \text{if } N_k > 0 \text{ and } \Delta_k \neq 0 \\ N_k, & \text{otherwise.} \end{cases}$$

Then

$$\gamma_k = \begin{cases} 1, & \text{if } N_k > 0 \text{ and } \Delta_k \neq 0 \\ 0, & \text{otherwise.} \end{cases}$$

**Remarks:** It can be checked that the BMA requires only  $5t^2$  multiplications. In 1975, Sugiyama *et al.* gave a somewhat different solution (for decoding the more general class of Goppa codes), based on the use of the venerable Euclidean algorithm (ca 300BC). [It

has been noted that in the 6<sup>th</sup> century, the Indian mathematician Brahmagupta solved a diophantine equation similar to the key equation by using the Euclidean algorithm.] Though the approach of Sugiyama *et al.* is now generally regarded as being easier to follow than the original arguments of Berlekamp and Massey, their algorithm is more expensive, taking  $7.5t^2$  multiplications. (There are several other works on related issues, see, *e.g.*, [8], [40], [41]). In fact, however, there can be various methods of implementing (*i.e.*, computing the quotient in) the Euclidean algorithm, and in his Ph.D. thesis, Citron (1986) showed how a particular choice can lead to exactly the BMA. The rest of this note is devoted to explaining the background to Citron's work, and to describing some of his results, which for several reasons have unfortunately remained unpublished. In particular, Citron developed a version particularly studied for parallel implementation in VLSI, *viz.*, an implementation with identical locally connected processing elements laid out in a regular fashion. He called his version the Microlevel Euclidean Algorithm or MLE.

## Early Stanford Work on the BMA

My own interest in the BMA began with my first students in the linear systems area, Bradley Dickinson (Ph.D., 1974) and Martin Morf (Ph.D., 1975). By then it had been realized that over the field of real (rather than binary) numbers, Massey's realization problem had a long and rich history, related to Padé approximations, continued fractions, etc. Also in 1965, B.L. Ho, a student of R.E. Kalman, had already addressed the minimal realization problem (over the reals), basing his analysis on solving linear systems of equations of the form

$$\begin{bmatrix} S_1 & S_2 & \dots & S_t \\ S_2 & \dots & \dots & S_{t+1} \\ \vdots & & & \vdots \\ S_t & \dots & \dots & S_{2t-1} \end{bmatrix} \begin{bmatrix} \lambda_t \\ \lambda_{t-1} \\ \vdots \\ \lambda_1 \end{bmatrix} = - \begin{bmatrix} S_{t+1} \\ S_{t+2} \\ \vdots \\ S_{2t} \end{bmatrix}, \quad (7)$$

where  $\lambda(z) = 1 + \lambda_1 z + \dots + \lambda_t z^t$ . Once  $\lambda(z)$  is known,  $\omega(z) = 1 + \omega_1 z + \dots + \omega_t z^t$  can be formed by (back substitution) from the triangular set of equations

$$\begin{bmatrix} \mathbf{O} & S_1 & & \\ & S_1 & S_2 & \\ & \ddots & \vdots & \\ S_1 & S_2 & \dots & S_t \end{bmatrix} \begin{bmatrix} \lambda_t \\ \lambda_{t-1} \\ \vdots \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_t \end{bmatrix} \quad (8)$$

We can rewrite (7) as

$$S_j = - \sum_{i=1}^t \lambda_i S_{j-i}, \quad j = (t+1) : 2t, \quad (9)$$

which shows that the  $\lambda_i$  can be regarded as the coefficients of a feedback shift register (FSR) for forming the  $\{S_j\}$ . Solving the key equation, assuming  $t$  errors have occurred, is equivalent to finding a length  $t$  FSR that matches the known syndrome sequence (when the FSR is initialized with the first  $t$  syndromes).

This is exactly the problem solved by Massey, except that he (following Berlekamp) also gave a fast recursive way of solving the equations (7). The point, as noted by Berlekamp, is that to solve the linear equations (7) by the standard method of Gaussian eliminations would require  $\mathcal{O}(t^3)$  multiplications, compared to the  $\mathcal{O}(t^2)$  of the BMA. However, note that the coefficient matrix in (7) is a structured matrix: it is constant along the antidiagonals, *i.e.*, it is a Hankel matrix (see, *e.g.*, the famous book on matrix theory of Gantmakher, especially Ch. 15, which was in part the inspiration for the Ho-Kalman method). The number of multiplications can be reduced to  $\mathcal{O}(t^2)$  by properly exploiting this structure, and this is exactly what Berlekamp and Massey did by devising a recursive (in  $t$ ) method of solution. Moreover a recursive method avoids an apparent difficulty with the above equations, *viz.* that (7) implies that the number of errors,  $t$ , is known *a priori*.

Just as the coding theorists had not been aware of earlier systems work, in system theory it was only in 1971 that J. Rissanen published a method of solving the Hankel equations recursively, but still at a cost  $\mathcal{O}(t^3)$ . In other words, he didn't fully exploit the Hankel structure, though he did do so later, in 1974, but under the assumption that the leading minors of the Hankel coefficient matrix were all nonzero. However, this is not the situation in any interesting partial realization problem, where a low-order rational fraction is sought to match a long sequence of transfer-function (or impulse response) coefficients.

I had become interested in studying multivariable system theory around this time. Rissanen was at IBM Research in San Jose and used at that time to visit me fairly regularly to discuss a variety of common technical interests. In system theory, unlike coding theory, it is the multivariable case that is of more interest and offers more challenges. It seemed natural to test our understanding of multivariable system theory by trying to extend Rissanen's interesting work to the matrix partial realization problem, where the syndromes  $S_j$  are matrices rather than scalars. My students and I found a solution, though it was rather detailed and complicated, which is perhaps why it was accepted quite quickly with minimal comments (Dickinson *et al.* (1984)). However I was dissatisfied with the complexity of our solution and kept (among many other things) looking for more insight into even the scalar problem.

The key was provided by a casual browsing, for reasons I can't exactly remember now, in a Dover paperback by Householder, Theory of Matrices in Numerical Analysis. On pp. 20-23 of this book, there appear Hankel matrices and a fast  $\mathcal{O}(t^2)$  way of factoring them using the so-called Lanczos algorithm. The Lanczos algorithm is a famous method for orthogonalizing a set of so-called Krylov vectors of the form  $K = [b, Ab, \dots, A^{n-1}b]$ ; this structure was exploited to speed-up the Gram-Schmidt technique. The connection with our problem is that the Gramian matrix,  $KK^*$ , is a Hankel matrix. The Lanczos algorithm is wellknown in numerical analysis as a method for efficiently computing a few eigenvalues and eigenvectors of a large matrix (see, *e.g.*, Parlett (1980)). Recently Parlett (1992) has used system theory ideas to better explain special problems arising in the application of the algorithm – they turn out to be related to nonminimality (*i.e.*, loss of controllability or observability or both) of an associated linear system. Another interesting application has been for fast subspace computation in some new high-resolution direction-finding algorithms (MUSIC, ESPRIT, WSF, etc.), which are now being applied to various communication problems – see Xu and Kailath (1994). It is not unlikely that these works may have some coding theory implications as well!

In any case, even in the mid-seventies Householder's discussion seemed interesting enough that I suggested to my third student in the linear systems area (S.Y. Kung, Ph.D. 1977) that he try to figure out a connection to minimal realization and to the BMA. This turned out to be a fruitful venture.

The solution Householder described applied only to so called strongly nonsingular (or strongly regular) (Hankel) matrices, *viz.*, those with all nonzero leading minors. In this case, there was the following result. Rewrite (7) as

$$M(t, t+1) \begin{bmatrix} \lambda_t \\ \vdots \\ \lambda_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

where  $M(t, t+1)$  is the  $t \times (t+1)$  Hankel matrix with first row  $\{s_1, \dots, s_{t+1}\}$ . The key step in solving the equations (10) is to factor  $M_{t,t+1}$  as

$$M(t, t+1) P(t, t+1) = Q(t, t+1) \quad (11)$$

where  $P$  is an upper triangular matrix with unit diagonal, and  $Q$  is lower triangular (with nonzero diagonal elements because of the strong nonsingularity assumption). If the columns of  $P$  are denoted by  $\{p_0, \dots, p_t\}$  then it can be shown that the column  $p_k$  will match the first  $2k$  syndromes. Now Householder showed, following Lanczos (1951), that the  $p_k$  could be recursively determined by using the following 3-term polynomial recursion, where  $p_k(z)$  is associated with the vector  $p_k$  in the usual way ( $p_k(z) = [1 \ z \ \dots \ z^k] p_k$ ),

$$p_{k+1}(z) = (z - \lambda_k) p_k(z) - \delta_k p_{k-1}(z). \quad (12)$$

Such three-term recursions are routinely encountered in the theory of orthogonal polynomials (see, *e.g.*, Szegő (1939)), and in fact the polynomials  $p_k(z)$  above are often called the Lanczos orthogonal polynomials. The initial polynomials  $p_0(z)$  and  $p_1(z)$  are easy to find, and the constants  $\lambda_k$  and  $\delta_k$  can be determined by forming certain inner products. However Kung noted that they could also be found without forming inner products by simultaneously propagating the columns of  $Q$ , and working with a certain extension of the Hankel matrix  $M$ , *viz.*,

$$\widehat{M} = M(2t, 2t+1) \text{ with } S_{2t+1} = 0 = \dots = S_{3t}.$$

This was an important fact, whose significance was only appreciated several years later. In Kung's thesis, this was merely noted as an alternative to the BMA, of somewhat less interest because the number of computations, though still  $\mathcal{O}(t^2)$ , had a higher leading coefficient.

However the three-term recursions break down when any leading minors are zero, which is usually the case in partial realization problems (where a system of order less than  $t$  can often match more than  $2t$  terms of a given sequence; *e.g.*,  $(1 - az^{-1})^{-1}$  matches the input sequence  $1, a, a^2, \dots$ ). Kung found a nice way of generalizing the earlier arguments to apply to this case by using generalized Lanczos recursions of the form

$$p_{k+\eta_k}(z) = \lambda_k(z) p_k(z) - \delta_k p_{k-1}(z) \quad (13)$$

where  $\lambda_k(z)$  is now a polynomial of degree  $\eta_k$  (rather than of degree 1); the indices  $\eta_k$  are determined by the pattern of zero minors in  $\tilde{M}$  in a way that we do not have space to describe here (just as we did not describe the computation of  $\gamma_k$  and  $\delta_k$  in (12)). Again the coefficients of  $\gamma_k(z)$  can be found in different ways, in particular with or without using inner products. Moreover in the zero-minors case, the polynomials  $\lambda_k(z)$  are not unique, and it turns out that a particularly natural choice leads to the classical BMA. The above approach and the connections with matrix factorization clearly showed the nature of the nonuniqueness; the polynomial language also gives a nice compact description of the algorithm. Incidentally, these results extend the classical theory of orthogonal polynomials (see, *e.g.* Szegő (1939)), based on the assumption of strong regularity of the moment matrices. The appropriate generalizations were apparently first found by A. Magnus (1962) via a different approach, as noted by Gragg and Lindquist (1985). This paper, and earlier ones of Gragg (1974) and Kalman (1979), have various points of contact with the results in Kung's Ph.D. thesis (1977). The continuing interest in this problem is shown by the very recent paper of Pal and Kailath (1994).

However the main contribution of this part of Kung's thesis was in showing how to get a simpler solution to the case of matrix syndromes by using a reduction to a certain so called polynomial echelon form (see Kailath (1980), Sec. 6.7 and Example 6.7-7). We wrote up these two sets of results — for the scalar and matrix problems. But the papers were taken without much change from the corresponding chapters of a somewhat hurriedly completed thesis (I was on sabbatical in Belgium, and Kung already had a job in the fall), and so the referees suggested considerable revision, which we never got around to doing. [I did clean up the scalar paper for use in Ch. 5 of my book on Linear Systems, but dropped the material in galley proof, to keep the book below 700 pages; in retrospect, an unfortunate mistake because after 1980 there was a resurgence of interest in the partial realization algorithm and the Lanczos-type solutions.] One reason was that there were already several different derivations of the BMA in the literature and there did not seem to be a pressing reason for one more, even though we had a slightly different (inner-product-free) version even in the scalar case.

So I dropped the subject and my two other students in Linear Systems, G. Verghese (Ph.D, 1979) and B. Lévy (Ph.D, 1980) worked on other problems. However in 1985 I returned to the BMA and to Kung's work in a roundabout way, via the study of Toeplitz matrices. For related work from a slightly different point of view, see the work of Blahut (1985). Also Chun and Kailath (1991).

### III Levinson and Schur Algorithms for Toeplitz Matrices

Toeplitz matrices seem to arise even more frequently than Hankel matrices, *e.g.*, as covariance matrices of stationary discrete-time processes. [They can be converted to Hankel matrices by reversing the order of the columns, but this makes it difficult to obtain recursions in the size.] In this connection, a fast algorithm of Levinson (1947) became famous in connection with significant commercial applications in geophysics and in speech processing. The Levinson algorithm is based on the so called 2-term (rather than 3-term) recursions derived by Szegő (1939) for polynomials orthogonal on the unit circle. To solve Toeplitz

equations of the form

$$a_n T_n = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \quad (14)$$

where  $T_n = [c_{|i-j|}]$ , we use the recursions

$$\begin{bmatrix} a_{m+1}(z) \\ a_m^\sharp(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_m \\ -k_m & 1 \end{bmatrix} \begin{bmatrix} a_m(z) \\ a_m^\sharp(z) \end{bmatrix}, \quad \begin{bmatrix} a_0(z) \\ a_0^\sharp(z) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (15)$$

where  $k_m$  found via an inner product operation using the  $\{c_i\}$  and the coefficients of  $a_m(z)$ . This can be checked to be an  $\mathcal{O}(m^2)$  algorithm as well.

$$k_{m+1} = (a_{m,m}c_1 + \dots + a_{m,1}c_m + c_{m+1}/\sigma_m^2) \quad (16)$$

$$\sigma_{m+1}^2 = \sigma_m^2(1 - |k_{m+1}|^2), \quad \sigma_0^2 = c_0 \quad (17)$$

As can be seen by comparing *e.g.*, (5)-(6) of the BMA with (15)-(16) above, the Levinson algorithm and the BMA have several similarities. Berlekamp's derivation is in fact in the same spirit as Levinson's – the  $t$ -th order solution is trivially extended to give a trial  $(t+1)$ -th order solution, then the resulting mismatch of the right-hand sides is used to appropriately modify the trial solution. There is also a three-term form of the Levinson recursion, similar to the Lanczos form of the BMA (*cf.* (12)).

Now in 1978, a slight variation of the Levinson algorithm was found, in which the coefficients  $\{k_m\}$  were found *without* forming inner products. This actually arose from a somewhat accidental encounter at Stanford with a former student, Patrick Dewilde, of one of my colleagues in circuit theory (R. Newcomb). Patrick was teaching system theory and numerical linear algebra at Leuven in Belgium; he is now at Delft in Holland. Some work on a scattering theory formulation of Kalman filtering problems (Riccati equations also arise in transmission-line theory) had led us to some problems in circuit theory. So, when I bumped into Patrick in 1976, as he was briefly passing through Stanford (for reasons that make another story itself), I asked him for help. This meeting led to a long and fruitful collaboration, continuing to this day. Dewilde led my student, A. Vieira (Ph.D., 1977), and me to a famous 1917 paper of I. Schur (on testing when a power series is bounded in the unit disc), which Patrick had encountered in his postdoctoral work on network theory, working with H. Helson at UC Berkeley. The Levinson algorithm can be interpreted as giving a fast algorithm for factoring the inverse of Toeplitz matrices. The Schur algorithm gives a fast method for the triangular factorization of Toeplitz matrices themselves, and thereby to an alternative solution of Toeplitz linear equations. The basic structure of the two algorithms is the same, *viz.*, Eq. (15), except for initial conditions and for a different, inner-product-free, method of computing the key recursion coefficients  $k_{m+1}$  (*i.e.*, not as in Eq. (16)-(17)).

However the Schur algorithm remained in the background for a few years, till in the early eighties our group began to look at problems in designing VLSI systems (systolic arrays) for various signal processing problems (see, *e.g.* [33], [20]). In VLSI design, inner products are to be avoided, because while the individual products can be formed in parallel, summing them up will take at least time  $\mathcal{O}(\log t)$  and require global (nonlocal) wiring connections in an IC implementation. Therefore parallel implementation of inner-product-based algorithms, such as the BM algorithm or the Levinson algorithm, will take  $\mathcal{O}(t \log t)$  steps, as

compared to the  $\mathcal{O}(t)$  steps we would expect. Here is where the alternative inner-product-free Schur algorithm for Toeplitz matrices shines: while the Schur algorithm is slightly more expensive (it has a higher coefficient than the Levinson algorithm) in flop counts, with parallel computation it can be much faster and better adapted to VLSI implementation (only local connections).

I mentioned this fact in a seminar course, and Todd Citron (Ph.D., 1986) immediately picked it up. Todd had been taking a coding theory class, though his targeted thesis area was sensor array signal processing (adaptive beamforming, etc, on which he already had a publication). Todd pointed out to me that the difference between the BM and the generalized Lanczos algorithms was precisely the same as between the Levinson and Schur algorithms. The BMA used inner products and was somewhat faster on a serial machine; however on a parallel machine, the generalized Lanczos algorithm (*cf.*, Eq. (13)) would only require  $\mathcal{O}(t)$  steps as compared to  $\mathcal{O}(t \log t)$  for the BMA!

Inspired by this observation, Todd switched thesis topics and in about a year completed a thesis on the BMA, emphasizing the development of architectures matched to VLSI implementation. To reach this goal, he first showed that the Euclidean algorithm method of Sugiyama *et al.* was in fact closely related to the (inner-product-free version of the) generalized Lanczos algorithm (GLA). The drawback to the Euclidean algorithm approach was that it is not easy to see the impact of algorithmic changes on the implementation architecture. This difficulty would no doubt have been overcome, but new insight into the question came from another direction. Another student, Alfred (Freddy) Bruckstein (Ph.D., 1984) was completing a thesis on inverse scattering theory, which among other things, turned out to provide a very nice physical (transmission-line) interpretation for the Levinson and Schur algorithms applied to Toeplitz matrices and equations. The two algorithms turn out to correspond to two basic approaches to inverse transmission line problems, called *layer-adjoining* and *layer-peeling* (see Bruckstein and Kailath, 1987). Freddy was intrigued by the thought of a similar physical approach to Hankel equations, and Todd and he soon figured this out. Now one had a ‘generalized’ transmission line, with none of the usual *energy-conservation* properties but with the critical property of *causality*. With this model they were able to show that the BMA and the GLA again corresponded to layer-adjoining and layer-peeling (see [12] and [5]).

The circuits intuition associated with these arguments was skillfully exploited by Citron to examine various architectural tradeoffs, leading finally to an algorithm he called the Microlevel Euclidean (MLE) algorithm. It is difficult to elaborate on this algorithm without going into a lot of detail, and this is not the place to do so. Interested readers may consult Todd’s 1986 Ph.D. thesis. Soon after submitting the thesis, Todd returned to Hughes Communication Systems Division, where he has been kept too busy to worry about papers and publication. From time to time we meet or talk and briefly discuss some new angles on the problem, but then again get absorbed in different things. So there will probably be no other source for it.

Moreover Ian Blake told me just a few days ago that there was now a Berlekamp-Welch decoding algorithm that didn’t require the *key* equation, *sic transit gloria mundi*. That probably means that our particular view will no longer be relevant to the decoding problem, but like hope it still lives on with the promise of shedding more light on the study of Toeplitz and related matrices. The related matrices are in fact Bezoutian matrices, which

are encountered in the problem of GCD computation for polynomials. This of course was the problem for which the Euclidean algorithm was invented!

And so the story goes on and on. I was going to say eternally, but I recall a footnote in a paper of Dave Forney's on how Jim Massey had explained to him that the word eternal was often misapplied to sequences that began at some fixed time and then went on forever. These were, in theology, called *aveternal* to distinguish them from *eternal* (pre-existing and everlasting). I would not wish aveternity on Jim (or on anyone), but perhaps we can all join in invoking for him the old Polish blessing, "May you live 120 years."

## References

- [1] Berlekamp, E.R., *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.
- [2] Blahut, R.E., "Algebraic Fields, Signal Processing and Error Control," *Proc. IEEE*, vol. 73, no. 5, pp. 874-893, May 1985.
- [3] Blahut, R.E., *Fast Algorithms for Digital Signal Processing*, Addison-Wesley, 1985.
- [4] Bruckstein, A.M., "Scattering Models in Signal Processing," *Ph.D. Dissertation*, Stanford University, 1984.
- [5] Bruckstein, A.M., T.K. Citron, and T. Kailath, "On Inverse Scattering and Partial Realizations," *Int'l. J. Control*, vol. 48, no. 4, pp. 1537-1550, Oct. 1988.
- [6] Bruckstein, A.M. and T. Kailath, "Inverse Scattering for Discrete Transmission-Line Models," *SIAM Review*, vol. 29, no. 3, pp. 359-389, Sept. 1987.
- [7] Bruckstein, A.M. and T. Kailath, "An Inverse Scattering Framework for Several Problems in Signal Processing," *IEEE ASSP Mag.*, vol. 4, no. 1, pp. 6-20, Jan. 1987.
- [8] Cheng, U., "On the Continued Fraction and Berlekamp's Algorithm," *IEEE Trans. Info. Theory*, vol. IT-30, no. 3, pp. 541-544, May 1984.
- [9] Chun, J. and T. Kailath, "Divide-and-Conquer Solutions of Least-Squares Problems for Matrices with Displacement Structure," *SIAM J. Matrix Anal. Appl.*, vol. 12, no. 1, pp. 128-145, Jan. 1991.
- [10] Citron, T.K., "Algorithms and Architectures for Error Correcting Codes," *Ph.D. Dissertation*, Stanford University, 1986.
- [11] Citron, T.K. and T. Kailath, "Method and Means for Error Detection and Correction in High Speed Data Transmission Codes," Patent, 1987.
- [12] Citron, T.K., A.M. Bruckstein, and T. Kailath, "An Inverse Scattering Approach to the Partial Realization Problem," *19th CDC*, Las Vegas, Dec. 1984.
- [13] Citron, T.K. and T. Kailath, "Euclid's Algorithm, Scattering Theory, and a VLSI Architecture for Decoding Reed-Solomon Codes," *presented at 1985 Info. Theory Symp.*

- [14] Dewilde, P., A. Vieira, and T. Kailath, ‘On a Generalized Szegő-Levinson Realization Algorithm for Optimal Linear Prediction based on a Network Synthesis Approach,’ *IEEE Trans. Circ. Sys.*, vol. CAS-25, no. 9, pp. 663-675, Sept. 1978.
- [15] Dickinson, B., “Properties and Applications of Matrix Fraction Descriptions of Linear Systems,” *Ph.D. Dissertation*, Stanford University, 1974.
- [16] Dickinson, B., M. Morf, and T. Kailath, “A Minimal Realization Algorithm for Matrix Sequences,” vol. AC-19, no. 1, pp. 31-38, Feb. 1974.
- [17] Gragg, B. and A. Lindquist, “On the Partial Realization Problem,” *Linear Algebra Appl.*, vol. 50, pp. 277-319, 1985.
- [18] Ho, B.L. and R.E. Kalman, “Effective Construction of Linear State-Variable Models from Input/Output Descriptions,” *Proc. Third Allerton Conf.*, pp. 449-459, 1965. Also in *Regelungstechnik*, vol. 14, pp. 545-548, 1966.
- [19] Householder, A.S., *The Theory of Matrices in Numerical Analysis*, Dover, New York, 1964.
- [20] Jagadish, H.V., S.K. Rao, and T. Kailath, “Array Architectures for Iterative Algorithms,” *Proc. IEEE*, vol. 75, no. 9, pp. 1304-1321, Sept. 1987.
- [21] Kailath, T., *Linear Systems*, Prentice-Hall, New Jersey, 1980.
- [22] Kalman, R.E., “On Partial Realizations, Transfer Functions and Canonical Forms,” *Acta Polytech. Scand.*, vol. MA-31, pp. 9-32, 1979.
- [23] Kung, S.Y., “Multivariable and Multidimensional Systems: Analysis and Design,” *Ph.D. Dissertation*, Stanford University, 1977.
- [24] Lanczos, C., “Solution of Systems of Linear Equations by Minimized Iterations,” *J. Res. Nat. Bur. Stand.*, no. 49, pp. 35-53, 1952.
- [25] Levinson, N., “The Wiener RMS (Root-Mean-Square) Error Criterion in Filter Prediction and Design,” *J. Math. Phys.*, vol. 25, pp. 261-278, 1947.
- [26] Magnus, A., “Certain Continued Fractions Associated with the Padé Table,” *Math. Zeitschr.*, vol. 78, pp. 361-374, 1962.
- [27] Massey, J.L., “Shift-Register Synthesis and BCH Decoding,” *IEEE Trans. Info. Theory*, vol. IT-15, no. 1, pp. 122-127, Jan. 1969.
- [28] Morf, M., “Fast Algorithms for Multivariable Systems,” Ph.D. Dissertation, Stanford University, 1974.
- [29] Pal, D. and T. Kailath, “Fast Triangular Factorization and Inversion of Hermitian Toeplitz and Related Matrices With Arbitrary Rank Profile,” *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 4, pp. 1016-1042, Oct. 1993.
- [30] Parlett, B.N., *The Symmetric Eigenvalue Problem*, Prentice-Hall, New Jersey, 1980.

- [31] Parlett, B.N., "Reduction to Tridiagonal Form and Minimal Realizations," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 2, pp. 567-593, April 1992.
- [32] Peterson, W.W., "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes," *IRE Trans. Info. Theory*, pp. 459-470, Sept. 1960.
- [33] Rao, S.K. and T. Kailath, "Regular Iterative Algorithms and Their Implementation on Processor Arrays," *Proc. IEEE*, vol. 76, no. 3, pp. 259-282, March 1988.
- [34] Rissanen, J., "Recursive Identification of Linear Systems," *SIAM J. Control*, vol. 9, no. 3, pp. 420-430, 1971.
- [35] Rissanen, J., "Solution of Linear Equations with Hankel and Toeplitz Matrices," *Numer. Math.*, vol. 22, pp. 361-366, 1974.
- [36] Schur, I., "Über potenzreihen die im innern des Einheitskreises beschränkt sind," *Journal für die Reine Angewandte Mathematik*, vol. 147, pp. 202-232, 1917. English translation in *I. Schur Methods in Operator Theory and Signal Processing*, Operator Theory: Advances and Applications, I. Gohberg (ed.), vol. 18, pp. 31-88, Birkhäuser Verlag, Basel, 1986.
- [37] Sugiyama, Y., M. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes," *Information and Control*, vol. 27, pp. 87-99, 1975.
- [38] Szegő, G., *Orthogonal Polynomials*, Amer. Math. Socy., Rhode Island, 1939. (3rd ed., 1967).
- [39] Vieira, A., "Matrix Orthogonal Polynomials, with Applications to Autoregressive Modeling and Ladder Forms," *Ph.D. Dissertation*, Stanford University, 1977.
- [40] Welch, L.R. and R.A. Scholtz, "Continued Fractions and Berlekamp's Algorithm," *IEEE Trans. Info. Theory*, vol. IT-25, no. 1, pp. 19-27, Jan. 1979.
- [41] Wolf, J.K., "Redundancy, the Discrete Fourier Transform, and Impulse Repsonse Noise Cancellation," *IEEE Trans. Comm.*, vol. COM-31, no. 3, pp. 458-461, March 1983.
- [42] Xu, G. and T. Kailath, "Fast Estimation of Principal Eigenspace Using Lanczos Algorithm," *SIAM J. Matrix Anal.*, July 1994.

# Using Zech's Logarithm to Find Low-Weight Parity Checks for Linear Recurring Sequences

G. J. Kühn

Department of Electrical  
and Electronic Engineering  
University of Pretoria  
0002 PRETORIA, South Africa

W. T. Penzhorn

Department of Electrical  
and Electronic Engineering  
University of Pretoria  
0002 PRETORIA, South Africa

## Abstract

The fast correlation attack described by Meier and Staffelbach on certain stream ciphers requires that the number of taps of the feedback polynomial must be small, typically less than ten. The attack can be extended to feedback polynomials with an arbitrary number of taps if an efficient algorithm is available to find low-weight polynomial multiples of the feedback polynomial. Given an arbitrary polynomial of degree  $k$  over  $GF(2)$ , a method based on the “birthday paradox” can find weight-4 polynomial multiples of the polynomial having degree  $\leq 2^{k/4}$ . The computational complexity is  $O(2^{k/2})$ , and a table size of  $O(2^{k/2})$  is used. In this paper a technique based on Zech’s logarithm, using a significantly smaller table size but at the cost of increased computational complexity, is described. It is shown that weight-4 polynomials of degree  $O(2^{k/3})$  can be found requiring a table size of  $O(2^{k/3})$  and computational complexity  $O(2^{k/3}) + DL[O(2^{k/3})]$ , where the second term denotes the complexity of computing discrete logarithms in  $GF(2^k)$  of  $O(2^{k/3})$  field elements. If weight-4 polynomials of degree  $\leq O(2^{k/4})$  are sought, the table size is  $O(2^{3k/8})$  and the computational complexity  $O(2^{3k/8}) + DL[O(2^{3k/8})]$ . To find the discrete logarithm of field elements the use of Coppersmith’s algorithm is suggested.

## I Introduction

In cryptography the need to find low-weight polynomial multiples of an arbitrary polynomial arises from a method described by Meier and Staffelbach [1] to attack running-key stream ciphers based on the nonlinear combining of a fixed number of linear feedback shift registers (LFSRs). In particular, they have suggested an efficient algorithm, referred to as Algorithm B, which exploits the correlation known to exist between the stream cipher output and a LFSR. The algorithm has asymptotic complexity  $O(k)$ , when the number of taps  $t$  of the LFSR is fixed. However, if  $t$  grows linearly with LFSR length  $k$ , then the algorithm has complexity exponential in  $k$ . Consequently, the algorithm is only suitable for LFSRs with relatively few taps, i.e. parity checks having low weight ( $<\approx 10$ ).

Several researchers have proposed extensions of the original algorithm [3,4,5,6,7], so as to overcome this limitation.

In their paper Meier and Staffelbach propose the use of a coincidence method to find low-weight polynomial multiples of the LFSR feedback polynomial  $f(x)$ , which may be an arbitrary polynomial of degree  $k$  with coefficients in  $GF(2)$ . A table of polynomials  $x^a \oplus x^b \bmod f(x)$ , where  $0 \leq a, b \leq 2^{k/4}$ , is generated. Each reduced polynomial  $\bmod f(x)$  has degree  $k-1$  or less. The "birthday paradox" [8] states that there is probability better than  $1 - e^{-1}$  of at least one coincidence in the list. If the coincidence is  $x^a \oplus x^b \equiv x^c \oplus x^d$ , then  $x^a \oplus x^b \oplus x^c \oplus x^d = 0 \bmod f(x)$  is a polynomial multiple of  $f(x)$  of degree  $\leq 2^{k/4}$ . The size of the table, as well as the computational complexity, is  $O(2^{k/2})$ .

In this paper we develop a method based on Zech's logarithm which requires a significantly smaller table size, but at the cost of increased computational complexity.

## II Approach Based on Zech's Logarithm

Let  $x$  be a primitive element in  $GF(2^k)$ . Then Zech's logarithm is defined by

$$x^{z(i)} = 1 \oplus x^i \quad (1)$$

where  $i, z(i) \in \{0, 1, 2, \dots, 2^k - 2\} \cup \{-\infty\}$ . Thus  $z(i)$  is the logarithm of  $1 \oplus x^i$  to the base  $x$ . By convention  $z(0) = -\infty$ . The following properties of  $z(i)$  are easily proved [9]:

1.  $z(z(i)) = i$  (self-inverse).
2.  $z(2i) = 2z(i) \bmod 2^k - 1$ .
3.  $z(-i) = z(i) - i \bmod 2^k - 1$ .

Define  $z_i = z(i)$  and let  $Z_N = \{z_i, 1 \leq i \leq N\}$ . We assume that  $z_i \in Z_N$  is randomly distributed over the interval  $(0, L-1)$ , where  $L = 2^k$ , and that the intervals between adjacent values have a negative exponential distribution with rate parameter  $N/L$ . The assumption is motivated by the observed random distribution of  $z(i)$  in numerical calculations over finite fields  $GF(2^k)$  for small values of  $k$ , viz.  $k = 8, \dots, 24$ .

**Finding weight-3 polynomials.** The polynomial  $x^{z(i)} \oplus x^i \oplus 1$  is a multiple of  $f(x)$  having weight 3. Let  $\nu_{\max}$  be the maximum degree of any weight-3 polynomial that is useful in the cryptanalysis algorithm. The probability that  $z(i) \leq \nu_{\max}$  under the negative exponential distribution assumption is

$$P[z(i) \leq \nu_{\max}] \approx 1 - e^{-\nu_{\max}N/L} \quad (2)$$

If  $\nu_{\max} = O(2^{k/4})$ , then the required table size for a probability greater than  $1 - e^{-1}$  of finding a weight-3 polynomial is  $N = O(k^{3/4})$ . This large table size would limit the technique to values of  $k$  less than approximately 60.

**Finding weight-4 polynomials.** Suppose two values  $i, j$  are found such that  $z_i - z_j = d > 0$ . Then

$$x^{z_i} \oplus x^i \oplus 1 \oplus x^d (x^{z_j} \oplus x^j \oplus 1) = x^i \oplus x^{j+d} \oplus x^d \oplus 1 \quad (3)$$

is a weight-4 polynomial having degree  $\nu = \max(i, j+d)$ . If  $0 < i, j \leq N$ , then  $\nu_{\max} \leq N+d$ . The probability that two adjacent Zech logarithm values are within a distance  $d$  of each other is

$$P(z_i - z_j = d; N, L) = 1 - e^{-dN/L}, \quad d > 0 \quad (4)$$

The expected number of pairs spaced at a distance of  $d$  is

$$m = N(1 - e^{-dN/L}) \approx dN^2/L \quad (5)$$

Choosing  $d = N = L^{1/3}$  gives a table size of  $O(2^{k/3})$  and an expected weight-4 polynomial of degree  $\nu_{\max} \leq 2.2^{k/3}$ . The computational complexity is  $O(2^{k/3}) + DL[O(2^{k/3})]$ , where  $DL[O(2^{k/3})]$  is the complexity of computing the discrete logarithms of  $O(2^{k/3})$  field elements. The complexity to compute discrete logarithms will be discussed in the next section.

If the maximum degree of the weight-4 polynomial must be  $\nu_{\max} \leq O(L^{1/4})$ , the required parameters are obtained by setting  $d = L^{1/4}$  in Equation (5), giving the values  $\nu_{\max} \leq O(2^{k/4})$ ,  $N = O(2^{3k/8})$  and computational complexity  $O(2^{3k/8}) + DL[O(2^{3k/8})]$ .

### III Coppersmith's Algorithm for Discrete Logarithms in $GF(2^k)$

Let  $f(x)$  be a primitive polynomial of degree  $k$  over  $GF(2)$ . The elements of  $GF(2^k)$  are all polynomials over  $GF(2)$  of degree at most  $k-1$ . Any field element  $a(x) \in GF(2^k)$  can be represented as  $a(x) \equiv x^s \pmod{f(x)}$ . The *discrete logarithm problem* may be formulated as follows: given any  $a(x) \in GF(2^k)$ , find the unique integer  $s$ ,  $0 \leq s \leq 2^k-2$ , such that  $a(x) \equiv x^s \pmod{f(x)}$ . Coppersmith [2] has developed an efficient algorithm for the calculation of the discrete logarithm for any arbitrary field element in  $GF(2^k)$ . The algorithm consists of two stages.

**Stage I.** In the first stage, the discrete logarithms of all irreducible polynomials of degree less than or equal to a suitable bound  $b$  are calculated and placed in a database. A polynomial  $a(x)$  is said to be *smooth with respect to  $b$*  (or simply *smooth*) when  $a(x)$  is the product of irreducible factors having degree at most  $b$ . The asymptotic running time for this stage is  $O(\exp((1 + O(1))k^{1/3} \ln^{2/3} k))$ .

**Stage II.** Next, the discrete logarithm of a given field element  $a(x)$  is calculated. The given  $a(x)$  is processed until it is smooth, i.e. decomposable into irreducible factors of degree at most  $b$ . The corresponding discrete logarithms for each factor are then obtained from the database calculated in Stage I. The asymptotic running time for Stage II to compute one logarithm is given approximately as  $O(\exp(\sqrt{k/b} \ln k/b + \ln^2 k))$ .

Using estimates from [10] for  $k = 80$  and  $b = 20$ , the number of operations required to set up the database is approximately  $6 \times 10^9$ . This amount of computing can be done by a supercomputer in less than a day.

The more important consideration in this paper is the time required for the Stage II calculations, as they are repeated for each element in the table of Zech's logarithms. A substantial improvement in the speed of the second stage is proposed in [11]. Assuming  $k = 80$  and a bound  $b = 20$ , it is found from [10] that approximately ten extended Euclidean operations over  $GF(2)$  per table entry are required. Taking a table size of  $2^{80/3} \approx 10^8$  entries of 80-bit words, the total number of operations is approximately  $10^9$ . This number is expected to be computed within a day using a supercomputer.

## IV Conclusion

In this paper we have presented a systematic method based on Zech's logarithms for finding low-weight parity checks for linear recurring sequences generated by linear feedback shift registers. Finding weight-3 checks would only be feasible for shift register lengths of up to 60 bits. Weight-4 checks can be obtained by combining two weight-3 checks, which can be found by means of Zech's logarithm and calculation of the discrete logarithm in  $GF(2^k)$  with Coppersmith's algorithm. This method is independent of the number of feedback taps of the LFSR, and its applicability is limited to a large extent by the computational complexity of finding discrete logarithms of finite field elements. The proposed method was successfully applied to the case of  $GF(2^{31})$ , and it is expected to be feasible for shift register lengths of approximately 80 to 100 bits.

The authors would like to thank Mr. J. H. J. Filter for fruitful discussions and helpful comments in the preparation of this paper.

## References

- [1] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers ", *J. Cryptology*, pp. 159-176, 1989.
- [2] D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two", *IEEE Trans. Inform. Theory* , vol. IT-30, no. 4, pp. 587-594, 1984.
- [3] V. Chepyzhov and B. Smeets, "On a fast correlation attack on stream ciphers", *Advances in Cryptology, EUROCRYPT '91*, LNCS no. 547, Springer- Verlag, pp. 176-185, 1991.
- [4] M. J. Mihaljevic and J. Golic, "A fast iterative algorithm for shift register initial state reconstruction given the noisy output sequence", *Advances in Cryptology, AUSCRYPT '90*, LNCS no. 453, Springer- Verlag, pp. 165-175, 1990.
- [5] M. J. Mihaljevic and J. Golic, "A comparison of cryptanalytic principles based on iterative error-correction," *Advances in Cryptology, EUROCRYPT '91*, LNCS no. 547, Springer- Verlag, pp. 527-531, 1991.
- [6] K. Zeng and M. Huang, "On the linear syndrome method in cryptanalysis", *Advances in Cryptology, CRYPTO '88*, LNCS no. 405, Springer-Verlag, pp. 469-478, 1990.

- [7] K. Zeng and C. H. Yang, “An improved linear syndrome algorithm in cryptanalysis with applications”, *Advances in Cryptology, CRYPTO '90*, LNCS no. 405, Springer-Verlag, pp. 469-478, 1990.
- [8] K. Nishimura and M. Sibuya, “Probability to meet in the middle”, *J. Cryptology*, Vol. 2, No. 1, pp 13-22, 1990.
- [9] K. Huber, “Some comments on Zech’s logarithm”, *IEEE Trans. Inform. Theory*, vol. IT-36, no. 4, pp. 946-950, 1990.
- [10] A. M. Odlyzko, “Discrete logarithms and their cryptographic significance”, *Proc. EUROCRYPT '84*, LNCS, Springer-Verlag, pp. 224-314, 1985.
- [11] I. F. Blake, R. Fuji-Hara, R. C. Mullin, and S. A. Vanstone, “Computing logarithms in finite fields of characteristic two”, *SIAM J. Alg. Disc. Methods*, 5, pp. 276-285, 1985.

# Higher Order Derivatives and Differential Cryptanalysis

Xuejia Lai

\* $R^3$  Security Engineering AG  
CH-8607 Aathal, Switzerland

## Abstract

High-order derivatives of multi-variable functions are studied in this paper as a natural generalization of the basic concept used in differential cryptanalysis. Possible applications of such derivatives in cryptology are discussed.

## I Introduction

In 1990, just after Jim and I published at Eurocrypt'90 our new block cipher PES [1], the previous version of the IDEA cipher, differential cryptanalysis was proposed by Biham and Shamir [2] as a chosen-plaintext attack to find the secret-key of an iterated block cipher. In [3], Biham and Shamir showed that the full 16-round DES can be broken by differential cryptanalysis using only  $2^{47}$  encryptions, which is the first reported attack that finds the secret-key of DES with fewer encryptions than exhaustive key-search. Besides DES, differential cryptanalysis has been applied successfully to many other ciphers [4, 5], for example, the block ciphers FEAL, Khafre, REDOC, LOKI and Lucifer. Like every cipher designer who heard a new attack that can be applied to his cipher, we started to apply the differential cryptanalysis to our own proposal. As I talked to Jim about the first result of the analysis, he noticed immediately that the use of the properties of a Markov chain is essential to differential cryptanalysis. This observation led to the concept of "Markov Cipher" which we presented in [6], where it was shown that the differential cryptanalysis of many (if not all) practical block ciphers can be formulated in terms of the theory of Markov chains. For example, the implication of the only lemma in Biham and Shamir's paper is that DES is a Markov cipher. In particular, a fairly tight lower bound on the complexity of a differential cryptanalysis attack on a cipher can be derived in terms of parameters of the Markov chain. Although differential cryptanalysis requires far too many chosen plaintexts to be a practical attack on a cipher, the complexity of a differential cryptanalysis attack is the best measure of its cryptographic strength that is known today.

In this paper, we consider a possible generalization of the original (first-order) differential cryptanalysis in terms of higher-order derivative, which is defined in Section 2, where some

---

\*This work was carried out at the Signal and Information Processing Laboratory of the Swiss Federal Institute of Technology, Zürich, Switzerland

general properties of derivatives are studied. In Section 3 we consider some special properties of derivatives of binary functions. Section 4 discusses the applications of such higher order derivatives in cryptology. In particular, we consider the relationship among derivatives, differential cryptanalysis and linear structure of cryptographic functions.

## II Higher Order Derivatives

**Definition** Let  $(S, +)$  and  $(T, +)$  be Abelian groups. For a function  $f : S \rightarrow T$ , the *derivatives of f at point a*  $\in S$  is defined as

$$\Delta_a f(x) = f(x + a) - f(x).$$

Note that the derivative of  $f$  is itself a function from  $S$  to  $T$ , we can define the *i-th* ( $i > 1$ ) *derivative of f at*  $(a_1, a_2, \dots, a_i)$  as

$$\Delta_{a_1, \dots, a_i}^{(i)} f(x) = \Delta_{a_i}(\Delta_{a_1, \dots, a_{i-1}}^{(i-1)} f(x))$$

where  $\Delta_{a_1, \dots, a_{i-1}}^{(i-1)} f(x)$  being the  $(i-1)$ -th derivative of  $f$  at  $(a_1, a_2, \dots, a_{i-1})$ . The 0-th derivative of  $f(x)$  is defined to be  $f(x)$  itself.

For example, for  $i = 2$ , we have

$$\begin{aligned} \Delta_{a_1, a_2}^{(2)} f(x) &= \Delta_{a_2}(\Delta_{a_1} f(x)) \\ &= \Delta_{a_2}(f(x + a_1) - f(x)) \\ &= (f(x + a_1 + a_2) - f(x + a_2)) - (f(x + a_1) - f(x)) \\ &= f(x + a_1 + a_2) - f(x + a_1) - f(x + a_2) + f(x). \end{aligned}$$

It then follows that

$$f(x + a_1 + a_2) = \Delta_{a_1, a_2}^{(2)} f(x) + \Delta_{a_1} f(x) + \Delta_{a_2} f(x) + f(x).$$

In general, one can show the following result.

### Proposition 1

$$f(x + a_1 + a_2 + \dots + a_n) = \sum_{i=0}^n \sum_{1 \leq j_1 < \dots < j_i \leq n} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x). \quad (1)$$

**Proof** For  $n = 1$  it follows from the definition. Suppose (1) holds for  $n - 1$ . Then

$$\Delta_{a_1, \dots, a_n}^{(n)} f(x) \quad (2)$$

$$= \Delta_{a_1, \dots, a_{n-1}}^{(n-1)} f(x \oplus a_n) - \Delta_{a_1, \dots, a_{n-1}}^{(n-1)} f(x) \quad (3)$$

$$= \left( f(x + a_n + a_1 + \dots + a_{n-1}) - \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \dots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x + a_n) \right) \quad (4)$$

$$- \Delta_{a_1, \dots, a_{n-1}}^{(n-1)} f(x) \quad (5)$$

$$= f(x + a_1 + \cdots + a_n) - \left( \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x + a_n) \right) \quad (6)$$

$$- \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x) + \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x) + \Delta_{a_1, \dots, a_{n-1}}^{(n-1)} f(x) \quad (7)$$

$$= f(x + a_1 + \cdots + a_n) - \left( \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} (f(x + a_n) - f(x)) \right) \quad (8)$$

$$+ \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x) + \Delta_{a_1, \dots, a_{n-1}}^{(n-1)} f(x) \quad (9)$$

$$= f(x + a_1 + \cdots + a_n) - \left( \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} (\Delta_a f(x)) \right) \quad (10)$$

$$+ \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x) + \Delta_{a_1, \dots, a_{n-1}}^{(n-1)} f(x) \quad (11)$$

$$= f(x + a_1 + \cdots + a_n) - \left( \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}, a_n}^{(i+1)} f(x) \right) \quad (12)$$

$$+ \sum_{i=0}^{n-2} \sum_{1 \leq j_1 < \cdots < j_i \leq n-1} \Delta_{a_{j_1}, \dots, a_{j_i}}^{(i)} f(x) + \Delta_{a_1, \dots, a_{n-1}}^{(n-1)} f(x) \quad (13)$$

$$= f(x + a_1 + \cdots + a_n) - \left( \sum_{i=0}^{n-1} \sum_{1 \leq j_1 < \cdots < j_i \leq n} \Delta_{a_{j_1}, \dots, a_{j_i}, a_n}^{(i)} f(x) \right) \quad (14)$$

□

Some basic properties of the derivative are as follows.

$$\Delta_a(f + g) = \Delta_a f + \Delta_a g \quad (15)$$

$$\Delta_a(f(x)g(x)) = f(x + a)\Delta_a g(x) + (\Delta_a f(x))g(x) \quad (16)$$

Equation (15) is rather obvious and Equation (16) can be obtained as follows:

$$\begin{aligned} \Delta_a(f(x)g(x)) &= f(x + a)g(x + a) - f(x)g(x) \\ &= f(x + a)(g(x + a) - g(x)) + (f(x + a) - f(x))g(x) \\ &= f(x + a)\Delta_a g(x) + (\Delta_a f(x))g(x). \end{aligned}$$

□

**Proposition 2** Let  $\deg(f)$  denote the nonlinear degree of a multivariable polynomial function  $f(x)$ . Then

$$\deg(\Delta_a f(x)) \leq \deg(f(x)) - 1. \quad (17)$$

**Proof** It follows from equation (15) and (16), from the facts that  $\deg(f+g) \leq \max(\deg(f), \deg(g))$  and  $\deg(fg) \leq \deg(f) + \deg(g)$  and that the derivative of a linear function is a constant. □

### III Derivatives of Binary Functions

In what follows, we will consider only the binary functions and the group operation is the *bitwise XOR*, denoted by  $\oplus$ .

**Proposition 3** *Let  $L[a_1, a_2, \dots, a_i]$  be the list of all  $2^i$  possible linear combinations of  $a_1, a_2, \dots, a_i$ . Then,*

$$\Delta_{a_1, \dots, a_i}^{(i)} f(x) = \sum_{c \in L[a_1, a_2, \dots, a_i]} f(x \oplus c) \quad (18)$$

**Proof** We prove it by induction. For  $i = 1$  it is obvious. Suppose (18) holds for  $i - 1$ , then by definition,

$$\Delta_{a_1, \dots, a_i}^{(i)} f(x) = \Delta_{a_i}(\Delta_{a_1, \dots, a_{i-1}}^{(i-1)} f(x)) \quad (19)$$

$$= (\Delta_{a_1, \dots, a_{i-1}}^{(i-1)} f(x \oplus a_i)) \oplus (\Delta_{a_1, \dots, a_{i-1}}^{(i-1)} f(x)) \quad (20)$$

$$= \left( \sum_{c \in L[a_1, a_2, \dots, a_{i-1}]} f(x \oplus c \oplus a_i) \right) \oplus \left( \sum_{c \in L[a_1, a_2, \dots, a_{i-1}]} f(x \oplus c) \right) \quad (21)$$

$$= \sum_{c \in L[a_1, a_2, \dots, a_i]} f(x \oplus c). \quad (22)$$

□

**Corollary 4** *Derivatives of binary function is independent of the order in which the derivation is taken, i.e., for any permutation  $p(j)$  of index  $j$ ,*

$$\Delta_{a_1, \dots, a_i}^{(i)} f(x) = \Delta_{a_{p(1)}, \dots, a_{p(i)}}^{(i)} f(x) \quad (23)$$

The above results lead to the lower bound on the probability of derivative function taking on a certain value. This probability is essential in cryptanalysis using such derivatives.

**Proposition 5** *For function  $f : F_2^n \rightarrow F_2^n$  and linearly independent  $a_1, a_2, \dots, a_i$  in  $F_2^n$  and for any  $b \in F_2^n$ ,  $P(\Delta_{a_1, \dots, a_i}^{(i)} f(x) = b)$  is either 0 or at least  $2^{i-n}$  if  $x$  is uniformly random.*

**Proof** From equation (18), if at input  $x_0$  the derivative takes on value  $b$ , then at all  $(2^i)$  inputs  $x_0 \oplus c$ ,  $c \in L[a_1, a_2, \dots, a_{i-1}]$ , the derivative will take on value  $b$ . □

**Proposition 6** *If  $a_i$  is linearly dependent of  $a_1, a_2, \dots, a_{i-1}$ , then  $\Delta_{a_1, \dots, a_i}^{(i)} f(x) = 0$ .*

**Proof** If  $a_i$  is linearly dependent of  $a_1, \dots, a_{i-1}$ , then  $a_i$  is contained in the list  $L[a_1, a_2, \dots, a_{i-1}]$ . Thus, in (21) we have

$$\sum_{c \in L[a_1, a_2, \dots, a_{i-1}]} f(x \oplus c \oplus a_i) = \sum_{c \in L[a_1, a_2, \dots, a_{i-1}]} f(x \oplus c),$$

which implies that the derivative is zero. □

The above result shows that derivatives should be computed at the points that are linearly independent. Otherwise the higher-order derivatives will be trivially zero; such cases are of no interest for our purpose.

**Proposition 7** For any function  $f : F_2^n \rightarrow F_2^m$ , the  $n$ th derivative of  $f$  is a constant. If  $f : F_2^n \rightarrow F_2^n$  is invertible, then  $(n - 1)$ -th derivative of  $f$  is a constant.

**Proof** Each component function of  $f$  can have a nonlinear degree at most  $n$ . When  $f$  is invertible, the nonlinear degree of its component function is at most  $n - 1$ . The proof then follows from Proposition 2.  $\square$

**Example.** For

$$f(x_1, x_2, x_3, x_4) = x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_2x_3x_4,$$

we compute the second derivative at  $(0001, 1010)$ .

$$\Delta_{0001} f(x_1, x_2, x_3, x_4) = f(x_1, x_2, x_3, x_4 \oplus 1) \oplus f(x_1, x_2, x_3, x_4) = x_1x_2 \oplus x_2x_3,$$

$$\Delta_{1010}(x_1x_2 \oplus x_2x_3) = x_2 \oplus x_2 = 0.$$

Thus,  $\Delta_{(0001, 1010)}^{(2)} f(x_1, x_2, x_3, x_4) = 0$ . Note that  $(0001)$  and  $(1010)$  are linearly independent and function  $f$  has a nonlinear degree 3.

## IV Cryptographic Significance of Derivatives

**Differential cryptanalysis and derivatives** The basic concept of differential cryptanalysis is the probability of differentials. A *differential* is a couple  $(a, b)$ , where  $a$  is the difference of a pair of distinct inputs  $x$  and  $x^*$  and where  $b$  is a possible difference for the resulting outputs  $y = f(x)$  and  $y^* = f(x^*)$ . The *probability of an differential*  $(a, b)$  is the conditional probability that  $b$  is the difference  $\Delta y$  of the outputs given that the input pair  $(x, x^*)$  has difference  $\Delta x = a$  when the  $x$  is uniformly random. We denote this differential probability by  $P(\Delta y = b | \Delta x = a)$ . If the “difference” is defined by the group operation “+”, i.e., if  $\Delta x = x - x^*$ , then

$$P(\Delta y = b | \Delta x = a) = P(f(x + a) - f(x) = b) = P(\Delta_a f = b). \quad (24)$$

**Proposition 8** The probability of a differential  $(a, b)$  is the probability that the first derivative of function  $f(x)$  at point  $a$  takes on value  $b$  when  $x$  is uniformly random.

The success of differential cryptanalysis is based on the fact that many practical block ciphers are obtained from iterating a cryptographically weak round function. If the difference of a pair of inputs to the last round can be anticipated with a high probability, then the secret key used in the last round can usually be derived from the pair of outputs and from the difference at the input. By using high-order derivatives, the basic idea of differential cryptanalysis can be generalized to the case when more than two inputs are used simultaneously for deriving the secret key: *if a (nontrivial)  $i$ -th derivative of  $(r-1)$  round function takes on a value with high probability, then it is possible to derive the key for the last round from the known  $2^i$  outputs and from the anticipated derivative value.* Although some independent preliminary experiments [7, 8] indicated that cryptanalysis using high order derivative may not be more powerful than the first-order differential cryptanalysis, we expect, however, that the derivative will provide new measurement for the strength of cryptographic functions.

**Linear structure and derivatives** A function  $f$  is said to have a *linear structure* if there is a nonzero  $a$ , such that  $f(x+a) - f(x)$  remains invariant for all  $x$ . The study of linear structure has lead to attacks on cipher functions [9, 10] and to the nonlinearity criteria for cryptographic functions [11, 12]. From the definitions, it is easy to see that function  $f(x)$  has a linear structure if and only if there is a nonzero  $a$  such that the derivative of  $f(x)$  at  $a$  is a constant, or equivalently, if and only if function  $f$  has a differential of probability one. The relationship among the concepts of differential, linear structure and derivatives then suggests the following:

**A new design principle for cryptographic functions.** For each small  $i$ , the nontrivial  $i$ -th derivatives of function should take on each possible value roughly uniform. In particular a binary function from  $F_2^n$  to  $F_2^n$ , the nontrivial  $i$ -th derivatives should take on each possible value with probability about  $2^{i-n}$ .

## References

- [1] X. Lai and J. L. Massey, “A Proposal for a New Block Encryption Standard”, *Advances in Cryptology – EUROCRYPT’90, Proceedings*, LNCS 473, pp. 389-404, Springer-Verlag, Berlin, 1991.
- [2] E. Biham and A. Shamir, “Differential Cryptanalysis of DES-like Cryptosystems”, *Advances in Cryptology – CRYPTO’90, Proceedings*, LNCS 537, pp. 2-21, Springer-Verlag, Berlin 1991.
- [3] E. Biham and A. Shamir, “Differential Cryptanalysis of the full 16-round DES”, *Abstracts of CRYPTO’92*.
- [4] E. Biham and A. Shamir, “Differential Cryptanalysis of FEAL and N-Hash”, *Advances in Cryptology – EUROCRYPT’91, Proceedings*, LNCS 547, pp. 1-16, Springer-Verlag, Berlin 1991.
- [5] E. Biham and A. Shamir, “Differential Cryptanalysis of Snelru, Khafre, REDOC-II, LOKI and Lucifer”, *Advances in Cryptology – CRYPTO’91, Proceedings*, LNCS 576, pp. 156-171, Springer-Verlag, Berlin 1992.
- [6] X. Lai, J. L. Massey and S. Murphy, “Markov Ciphers and Differential Cryptanalysis”, *Advances in Cryptology – EUROCRYPT’91, Proceedings*, LNCS 547, pp. 17-38, Springer-Verlag, Berlin, 1991.
- [7] C. Harpes, “Notes on High Order Differential Cryptanalysis of DES,” Internal report, Signal and Information Processing Laboratory, Swiss Federal Institute of Technology, August 12, 1993.
- [8] E. Biham, “Higher Order Differential Cryptanalysis,” (Preliminary draft) August 13, 1993.
- [9] D. Chaum and J.H. Evertse, “Cryptanalysis of DES with a reduced number of rounds,” *Advances in Cryptology - CRYPTO’85, Proceedings*, pp. 192–211, Springer-Verlag, 1986.

- [10] J.H. Evertse, “Linear structures in block ciphers,” *Advances in Cryptology - EUROCRYPT’87, Proceedings*, pp. 249–266, Springer-Verlag, 1988.
- [11] W. Meier and O. Staffelbach, “Nonlinearity criteria for cryptographic functions,” *Advances in Cryptology - EUROCRYPT’89, Proceedings*, pp. 549–562, Springer-Verlag, 1990.
- [12] K. Nyberg, “On the construction of highly nonlinear permutations,” *Advances in Cryptology - EUROCRYPT’92, Proceedings*, pp. 92–98, Springer-Verlag, 1993.

# Coded MPSK Modulation for the AWGN and Rayleigh Fading Channels \*

Shu Lin

Department of Electrical Engineering  
University of Hawaii at Manoa  
Honolulu, Hawaii 96822, U.S.A.

Sandeep Rajpal

Department of Electrical Engineering  
University of Hawaii at Manoa  
Honolulu, Hawaii 96822, U.S.A.

Do Jun Rhee

Department of Electrical Engineering  
University of Hawaii at Manoa  
Honolulu, Hawaii 96822, U.S.A.

Dedicated to Professor James L. Massey for his contributions in coding theory and its applications.

## Abstract

This paper investigates two methods of constructing bandwidth efficient MPSK modulation codes for the AWGN and Rayleigh fading channels. The first method is the multilevel coding method devised by Imai and Hirakawa. The multilevel coding method is a powerful technique for constructing bandwidth efficient modulation codes systematically with arbitrarily large distance parameters from Hamming distance component (block or convolutional) codes in conjunction with proper bits-to-signal mapping through signal set partitioning. Particularly, it provides the flexibility to coordinate the distance parameters of a code such that the best performance for a given channel can be attained. Furthermore, the multilevel modulation codes constructed by this method allow the use of multistage decoding procedures that provide good trade-offs between performance and decoding complexity. The second method is to construct TCM codes using convolutional codes with good free branch distance in conjunction with the multilevel coding technique.

Some good MPSK modulation codes for both the AWGN and Rayleigh fading channels are constructed and their error performances are given.

## I Introduction

The multilevel method devised by Imai and Hirakawa [1] is a powerful technique for constructing bandwidth efficient modulation codes systematically with arbitrarily large distance parameters from Hamming distance component codes (block or convolutional, binary

---

\*This research was supported by NSF Grants NCR-911540, BCS-9021435, and NASA Grant NAG 5-931.

or nonbinary) in conjunction with proper bits-to-signal mapping through signal set partitioning. Particularly, it provides the flexibility to coordinate the distance parameters of a code such that the best performance for a given channel can be attained. Furthermore, the multilevel modulation codes constructed by this method allow the use of multistage decoding procedures that provide good trade-offs between error performance and decoding complexity.

In this paper, constructions of multilevel bandwidth efficient modulation codes for both the AWGN and Rayleigh fading channels are presented. Distance parameters, such as the minimum squared Euclidean distance, minimum symbol distance, and minimum product distance, which determine the error performance of a multilevel modulation code, are expressed in terms of the minimum Hamming distance of the component codes. Guidelines for constructing good multilevel modulation codes for either the AWGN channel or the Rayleigh fading channel are presented. Error performances of some good multilevel modulation codes for either the AWGN or the Rayleigh fading channel are given. These codes achieve high performance with low decoding complexity.

The organization of this paper is as follows. In Section II, a brief review of the multilevel method for constructing multilevel modulation codes is given. Distance parameters of a multilevel modulation code are defined and expressed in terms of the minimum Hamming distances of the component codes. Guidelines for constructing good modulation codes for either the AWGN or the Rayleigh fading channel are presented. Some good basic multilevel modulation codes and their error performances using multistage decoding are given. These codes achieve significant coding gains over uncoded reference systems with low decoding complexity. In Section III, the construction of TCM codes using convolutional codes with good free *branch distance* in conjunction with the multilevel coding technique is presented.

## II Multilevel Coded Modulation

In a coded modulation system, information sequences are encoded into signal sequences over a certain modulation signal set. These signal sequences form a modulation code. In the following, we first define some important distance parameters of a modulation code and then give a brief review of the basic multilevel method for constructing modulation codes.

Let  $C$  be a modulation code of length  $n$  with signals from a certain modulation signal space  $S$ . The error performance of  $C$  depends on several distance parameters. Let  $d^2(\mathbf{x}, \mathbf{y})$  denote the *squared Euclidean distance* between two code sequences,  $\mathbf{x}$  and  $\mathbf{y}$ , in  $C$ . The *minimum squared Euclidean distance* of  $C$ , denoted  $d_E^2[C]$ , is defined as follows:

$$d_E^2[C] \triangleq \min \{ d^2(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in C \text{ and } \mathbf{x} \neq \mathbf{y} \} \quad (2.1)$$

The *Hamming distance* between two code sequences  $\mathbf{x}$  and  $\mathbf{y}$ , denoted  $\delta_H(\mathbf{x}, \mathbf{y})$ , is the number of different symbols between the two sequences. The *minimum distance* of  $C$ , denoted  $\delta_H[C]$ , is defined as the minimum distance between any two code sequences in the code. The *product distance* between  $\mathbf{x}$  and  $\mathbf{y}$  denoted by  $\Delta_p^2(\mathbf{x}, \mathbf{y})$  is defined as follows:

$$\Delta_p^2(\mathbf{x}, \mathbf{y}) = \prod_{k=1, x_k \neq y_k}^n d^2(x_k, y_k) \quad (2.2)$$

where  $d^2(x_k, y_k)$  is the *squared Euclidean distance* between  $k$ th signals,  $x_k$  and  $y_k$ , of  $\mathbf{x}$  and  $\mathbf{y}$ . The *minimum product distance* of  $C$ , denoted  $\Delta_p^2[C]$ , is the minimum product distance between any two code sequences with symbol distance  $\delta_H[C]$  in the code.

For the AWGN channel, the error performance of a code depends primarily on its minimum squared Euclidean distance and path multiplicity[2]. For the Rayleigh fading channel, the error performance of a code depends primarily on its minimum symbol distance, minimum product distance, and path multiplicity [3], [4]. It depends on the minimum squared Euclidean distance to a lesser degree.

The basic multilevel method for constructing modulation codes consists of five steps: (1) selection of a modulation signal set  $S$ ; (2) labeling of signal points by strings of labeling symbols through signal set partitioning; (3) selection of component codes; (4) combining component codes into a code over a signal label set; and (5) label-to-signal mapping to form a multilevel modulation code.

In this paper, we only consider the construction of modulation codes over the MPSK signal constellation. Generalization of the construction to QAM signal constellation is straightforward. Let  $S$  denote the two-dimensional  $2^\ell$ -PSK signal constellation with unit energy. Label each of the  $2^\ell$  signal points by a unique string of  $\ell$  bits,  $(a_1, a_2, \dots, a_\ell)$ , where  $a_1$  is the first labeling bit and  $a_\ell$  is the last labeling bit. The signal labeling is achieved by using the following partition chain,  $2^\ell$ -PSK/ $2^{\ell-1}$ -PSK/ $\dots$ /QPSK/BPSK. At the first partition level, the  $2^\ell$ -PSK signal constellation is partitioned into two  $2^{\ell-1}$ -PSK signal sets, one is labeled with “0” and other is labeled with “1”. At the second partition level, each  $2^{\ell-1}$ -PSK signal set is partitioned into two  $2^{\ell-2}$ -PSK sets, one is labeled with “0” and the other is labeled with “1”. The partition process continues until the  $\ell$ -level is reached where each subset contains a single signal point which is labeled with a unique sequence of  $\ell$  bits. The partitioning and labeling process of an 8-PSK signal constellation is shown in Figure 1. For  $0 < i \leq \ell$ , let  $d_i$  be the *intraset distance*[2] (the minimum squared Euclidean distance among signal points) of the  $2^{\ell-i+1}$ -PSK signal set. It is clear that the intraset distance of a set at the  $(i-1)$ th partition level is  $d_i$ , and the intraset distance increases as the partition level increases. This monotonically increasing property of the intraset distances,  $d_1, d_2, \dots, d_\ell$ , is a key to the construction of bandwidth efficient modulation codes. The labeling strings formed from the above partitioning process have the following important property: for  $0 < i \leq \ell$ , two signal points with labels identical at the first  $i-1$  bit positions but different at the  $i$ th bit position are at least at a squared Euclidean distance  $d_i$  apart. The intraset distances of the partition chain 8-PSK/QPSK/BPSK for an 8-PSK signal set are:  $d_1 = 0.586$ ,  $d_2 = 2.0$ , and  $d_3 = 4.0$  respectively. The above partitioning process was first devised by Ungerboeck in his construction of TCM codes[2].

Suppose block component codes are used for the code construction. For  $1 \leq i \leq \ell$ , let  $A_i$  be a binary  $(n, k_i, \delta_i)$  linear block code of length  $n$ , dimension  $k_i$ , and minimum Hamming distance  $\delta_i$ . Let

$$\begin{aligned} \mathbf{v}^{(1)} &= (v_1^{(1)}, v_2^{(1)}, \dots, v_j^{(1)}, \dots, v_n^{(1)}) \\ \mathbf{v}^{(2)} &= (v_1^{(2)}, v_2^{(2)}, \dots, v_j^{(2)}, \dots, v_n^{(2)}) \\ &\vdots \\ \mathbf{v}^{(\ell)} &= (v_1^{(\ell)}, v_2^{(\ell)}, \dots, v_j^{(\ell)}, \dots, v_n^{(\ell)}) \end{aligned} \tag{2.3}$$

be  $\ell$  codewords in  $A_1, A_2, \dots, A_\ell$  respectively. We form the following  $n$ -tuple:

$$\begin{aligned} \mathbf{v}^{(1)} * \mathbf{v}^{(2)} * \dots * \mathbf{v}^{(\ell)} &\triangleq (v_1^{(1)} v_1^{(2)} \dots v_1^{(\ell)}, v_2^{(1)} v_2^{(2)} \dots v_2^{(\ell)}, \dots, v_j^{(1)} v_j^{(2)} \dots v_j^{(\ell)}, \\ &\quad \dots, v_n^{(1)} v_n^{(2)} \dots v_n^{(\ell)}) \end{aligned} \quad (2.4)$$

For  $1 \leq j \leq n$ , we take  $v_j^{(1)} v_j^{(2)} \dots v_j^{(\ell)}$  as the label for a signal point in the  $2^\ell$ -PSK signal constellation. Let  $\lambda(\cdot)$  be the mapping which maps the label  $v_j^{(1)} v_j^{(2)} \dots v_j^{(\ell)}$  into its corresponding signal point  $s_j$ , i.e.,  $\lambda(v_j^{(1)} v_j^{(2)} \dots v_j^{(\ell)}) = s_j$ . Then

$$\begin{aligned} \lambda(\mathbf{v}^{(1)} * \mathbf{v}^{(2)} * \dots * \mathbf{v}^{(\ell)}) &\triangleq (\lambda(v_1^{(1)} v_1^{(2)} \dots v_1^{(\ell)}), \lambda(v_2^{(1)} v_2^{(2)} \dots v_2^{(\ell)}), \dots, \lambda(v_j^{(1)} v_j^{(2)} \dots v_j^{(\ell)}), \\ &\quad \dots, \lambda(v_n^{(1)} v_n^{(2)} \dots v_n^{(\ell)})) \end{aligned} \quad (2.5)$$

is a sequence of  $n$   $2^\ell$ -PSK signals. Let

$$C \triangleq \lambda[A_1 * A_2 * \dots * A_\ell] = \{\lambda(\mathbf{v}^{(1)} * \mathbf{v}^{(2)} * \dots * \mathbf{v}^{(\ell)}) : \mathbf{v}^{(1)} \in A_1, \mathbf{v}^{(2)} \in A_2, \dots, \mathbf{v}^{(\ell)} \in A_\ell\} \quad (2.6)$$

Then  $C$  is an  $\ell$ -level  $2^\ell$ -PSK block modulation code of length  $n$  and dimension  $k = k_1 + k_2 + \dots + k_\ell$ . Since  $k_1 + k_2 + \dots + k_\ell$  information bits are encoded into a code sequence of  $n$   $2^\ell$ -PSK signals, the spectral efficiency is  $\eta[C] = (k_1 + k_2 + \dots + k_\ell)/n$  bits/symbol. In the above construction,  $\ell$  component codes are used and each component code contributes one level of labeling.

The distance parameters of an  $\ell$ -level  $2^\ell$ -PSK modulation code can be expressed in terms of the minimum Hamming distances of its component codes.

**Distance Theorem:** Let  $d_E^2[C]$ ,  $\delta_H[C]$  and  $\Delta_p^2[C]$  denote the minimum squared Euclidean distance, minimum symbol distance and minimum product distance of an  $\ell$ -level  $2^\ell$ -PSK code,  $C = \lambda[A_1 * A_2 * \dots * A_\ell]$ , respectively. Then

$$(1) \quad d_E^2[C] = \min\{\delta_i d_i : 1 \leq i \leq \ell\}, \quad (2.7)$$

$$(2) \quad \delta_H[C] = \min\{\delta_i : 1 \leq i \leq \ell\}, \quad (2.8)$$

- (3) Let  $k$  be the smallest integer in the index set  $I = \{1, 2, \dots, \ell\}$  for which  $\delta_k = \delta_H[C]$ . Then

$$\Delta_p^2[C] = (d_k)^{\delta_k} \quad (2.9)$$

where  $d_1, d_2, \dots, d_\ell$  are the intraset distances of the partition chain

$$2^\ell\text{-PSK}/2^{\ell-1}\text{-PSK}/\dots/\text{QPSK}/\text{BPSK}. \quad \Delta \Delta$$

The proof of (2.7) can be found in [1], [5], [6]. Equations (2.8) and (2.9) are simply generalization of the results for 3-level 8-PSK modulation codes given in Lemma 1 of [7].

From the expressions for the minimum squared Euclidean distance and product distance of a multilevel modulation code given by (2.7) and (2.9), it is clear why the intraset distances should be kept as large as possible during the signal set partitioning and labeling process. The distance theorem provides general guidelines for constructing good multilevel

modulation codes for both the AWGN and fading channels. For the AWGN channel, the error performance of a modulation code depends mainly on its minimum squared Euclidean distance. In this case, expression (2.7) should be used as a guideline for code construction. For a given minimum squared Euclidean distance, the component codes should be chosen to maximize the spectral efficiency and minimize the decoding complexity and path multiplicity. For the Rayleigh fading channel, the error performance of a modulation code depends strongly on its minimum symbol and product distances. Both these distances should be as large as possible. In designing modulation codes for the Rayleigh fading channel, expressions of (2.8) and (2.9) should be used as the design guidelines.

The above multilevel code construction method and distance theorem can be generalized to QAM signal constellation in a straightforward manner. The component codes may be either block or convolutional codes, binary or nonbinary. The signal set partition can be either binary or nonbinary. Furthermore, it is not necessary to use  $\ell$  component codes in the construction of a multilevel modulation code over a signal set with  $2^\ell$  signals [8]. A modulation code is called an  $m$ -level code if  $m$  component codes are used in the construction.

In the following, two good 3-level 8-PSK block modulation codes for the AWGN and Rayleigh fading channels are constructed using the distance theorem as the general guideline. The error performance of these codes based on multistage soft-decision decoding [9],[10] are given. In the code construction, the component codes are chosen to have simple trellis structure so that the Viterbi decoding algorithm can be used at each decoding stage.

**Example 2.1:** Suppose we want to design a 3-level 8-PSK modulation code with minimum squared Euclidean distance 8 for the AWGN channel. From expression (2.7), we find that the three component codes must have minimum Hamming distances at least, 14, 4, and 2 respectively. In choosing the component codes to achieve the required minimum squared Euclidean distance, the overall decoding complexity, spectral efficiency, phase symmetry, and other factors must also be taken into consideration. One possible choice of the component codes is: (1)  $A_1 = (32, 6, 16)$ , the first order Reed-Muller(RM) code of blocklength 32; (2)  $A_2 = (32, 26, 4)$ , the third-order RM code of blocklength 32; and (3)  $A_3 = (32, 31, 2)$ , the even parity-check code of blocklength 32. With this choice, the resultant three-level 8-PSK code of blocklength 32,  $C(1) = \lambda[A_1 * A_2 * A_3]$ , has the following parameters:  $\eta[C(1)] = 63/32 = 1.969$  bits/symbol,  $d_E^2[C(1)] = 8$ ,  $\delta_H[C(1)] = 2$ , and  $\Delta_p^2[C(1)] = 16$ . This code has a spectral efficiency almost the same as the uncoded QPSK, 2 bits/symbol.

The reason to choose RM codes as component codes in the code construction is that they have relatively simple trellis diagrams [11],[12] and hence can be decoded with the soft-decision Viterbi decoding algorithm.

The  $(32, 6, 16)$  RM code has a simple 16-state and four-section trellis diagram[11]. This trellis diagram is loosely connected and consists of eight parallel and structurally identical two-state sub-trellis diagrams without cross connections between them [11],[12]. As a result, eight identical and simple two-state Viterbi decoders can be built to decode this code in parallel at the first stage of decoding of the 8-PSK code  $C(1)$ . This parallel structure of the trellis diagram not only simplifies the decoding complexity but also speeds up the decoding process.

The  $(32,26,4)$  RM code has a 16-state and four-section trellis diagram which consists of

two parallel and structurally identical eight-state sub-trellis diagrams without cross connections between them. The parallel structure in the trellis allows us to devise two identical and simple eight-state Viterbi decoders to decode the  $(32, 26, 4)$  RM code in parallel at the second stage of decoding of the 8-PSK code  $C(1)$ . The even parity-check  $(32, 31, 2)$  code has a two-state trellis diagram. Therefore, the decoding of this component code is also very simple. We see that the overall decoding complexity for the 8-PSK code  $C(1)$  is quite simple.

The bit error performance of  $C(1)$  with suboptimal and multistage decoding over the AWGN channel is shown in Figure 2. We see that the code achieves a 3.5 dB real coding gain over the uncoded QPSK at the bit-error-rate of  $10^{-6}$ . Since the code has a minimum distance 2 and a large minimum product distance 16, it also performs well over the Rayleigh fading channel as shown in Figure 3. The code achieves a 10.63 dB coding gain over the uncoded QPSK at the bit-error-rate of  $10^{-3}$ . The simulation is carried out assuming independent fading, no channel state information and squared Euclidean distance as the decoding metric.

Overall,  $C(1)$  provides good performance for both the AWGN and Rayleigh fading channels with relatively simple decoding complexity. Furthermore, it is invariant under  $45^\circ$  degree phase rotation which is important for synchronization purpose.

**Example 2.2:** Now consider the design of a three-level 8-PSK block modulation code for the Rayleigh fading channel. Suppose we want to construct a code with minimum distance 4 and minimum product distance greater than 1. From expression (2.8) in the distance theorem, we find that the smallest minimum Hamming distance of the component codes must be 4. From expression (2.9) in the distance theorem, we find that for the minimum product distance to be greater than 1, the first component code must not be the code with the smallest minimum Hamming distance. In this case, either the second or the third component code should be chosen to have the smallest minimum Hamming distance. A possible choice of the component codes is: (1)  $A_1 = (32, 16, 8)$ , the second-order RM code of length 32; (2)  $A_2 = (32, 26, 4)$ , the third-order RM code of blocklength 32; and (3)  $A_3 = A_2 = (32, 26, 4)$ . The resultant three-level 8-PSK code, denoted  $C(2) = \lambda[A_1 * A_2 * A_3]$ , has the following parameters:  $\eta[C(2)] = 2.125$  bits/symbol,  $d_E^2[C(2)] = 4.688$ ,  $\delta_H[C(2)] = 4$ , and  $\Delta_p^2[C(2)] = 16$ .

The first component code has a 64-state and four-section trellis diagram which consists of eight parallel and structurally identical eight-state sub-trellis diagrams without cross connections. As a result, eight identical eight-state Viterbi decoders can be devised to decode this code at the first stage of decoding of  $C(2)$ . The second and third component codes are both the third-order RM code of length 32 whose decoding complexity has already been discussed in the previous example. Again, we see that the overall decoding complexity for  $C(2)$  is quite simple.

The error performance of this code over the Rayleigh fading channel is shown in Figure 3. The code achieves a 12.39 dB coding gain over the uncoded QPSK at the bit error rate of  $10^{-3}$  with higher bandwidth efficiency. This code outperforms  $C(1)$  over the Rayleigh fading channel because it has larger minimum symbol distance. This code is invariant under  $90^\circ$  phase rotation.

More powerful multilevel MPSK codes can be constructed by using longer component

codes. However, this increases the decoding complexity drastically even with multistage decoding. One approach to overcome the complexity problem is to use short multilevel codes as the building blocks for constructing more powerful modulation codes or coded modulation systems to achieve high performance with reduced decoding complexity, such as concatenated coded modulation [13],[14].

### III Multilevel TCM Codes

The multilevel coding method presented in the previous section can be used for constructing TCM codes using convolutional codes as the component codes or using both convolutional and block codes as the component codes. In this section, we consider the construction of TCM codes using convolutional codes with good free branch distance as the component codes.

Let  $C$  be a rate  $k/n$  binary convolutional code. Using trellis representation, a code sequence in  $C$  is a path in the code trellis diagram consisting of a sequence of branches, each branch consists of  $n$  coded bits. For two coded sequences  $\mathbf{u}$  and  $\mathbf{v}$  in  $C$ , the *branch distance* between them, denoted  $d_b(\mathbf{u}, \mathbf{v})$ , is defined as the number of branches for which  $\mathbf{u}$  and  $\mathbf{v}$  differ. The *minimum free branch distance* of  $C$ , denoted  $d_{\text{B-free}}$ , is defined as the minimum branch distance between any two code sequences in  $C$ , i.e.,

$$d_{\text{B-free}} \triangleq \min\{d_b(\mathbf{u}, \mathbf{v}) : \mathbf{u}, \mathbf{v} \in C \text{ and } \mathbf{u} \neq \mathbf{v}\}$$

For a rate  $k/n$  feedforward binary convolutional code of total encoder memory  $\gamma$ , its minimum free branch distance  $d_{\text{B-free}}$  is upper bounded by  $1 + \lfloor \gamma/k \rfloor$ . A search has been performed on rate  $1/2$ ,  $2/3$  and  $3/4$  codes to find the best ones in terms of minimum free branch distance and the nearest neighbors. Some best codes are given in Tables 1.A, 2.A and 3. Most of the codes meet the upper bound on the minimum free branch distance. A search was also performed to find the optimal rate  $1/2$  and rate  $2/3$  branch distance convolutional codes for 4-PSK and 8-PSK transmission respectively, over the Rayleigh fading channel. The optimal codes obtained by the search are listed in Tables 1.B and 2.B.

Convolutional codes of good minimum free branch distances are quite suitable for constructing TCM codes with good minimum symbol distances. Let  $S$  be a modulation signal set with  $2^\ell$  signal points and  $C$  a rate- $k/n$  convolutional code of minimum free branch distance  $d_{\text{B-free}}$ . If each group of  $n$  coded bits is one-to-one mapped into a signal point in  $S$ , we obtain a TCM code over  $S$  with minimum symbol distance equal to the minimum free branch distance of the convolutional code  $C$ . If the resultant TCM code also has good minimum product distance, the code would perform well over the Rayleigh fading channel.

**Example 3.1:** Suppose the first code given in Table 2.B is used for constructing an 8-PSK TCM code. This code is a rate  $2/3$ , feedforward convolutional code with total encoder memory 4, minimum free branch distance 3, and generator matrix

$$G(D) = \begin{pmatrix} D^2 & 1 & D \\ 1 & D + D^2 & D \end{pmatrix}$$

Suppose natural mapping as shown in Figure 1 is used and every three coded bits are mapped into an 8-PSK signal point. The resultant modulation code, denoted  $C(3)$  is a 16-state 8-PSK TCM code with spectral efficiency two bits per symbol. The minimum symbol distance of this TCM code is 3. We also find that the minimum product and squared Euclidean distances of this code are 4.6864 and 5.17 respectively. Furthermore, this code has only two nearest neighbors in terms of both minimum symbol and product distances. It turns out that this code has the same distance parameters as those of the 16-state Ungerboeck code ( or the 16-state Schlegel-Costello code ) [2],[4] which has been regarded as optimal for the Rayleigh fading channel. The 16-state Ungerboeck code was actually designed for the AWGN channel and the convolutional code used in the construction is a rate-2/3 feedback convolutional code. The error performances of the TCM code constructed in this example and the 16-state Ungerboeck code over the Rayleigh channel are shown in Figure 4. The code achieves a 12.89 dB coding gain over the uncoded QPSK at the bit-error-rate of  $10^{-3}$  and performs slightly better than the 16-state Ungerboeck code for SNR greater than 11.5 dB, but the Ungerboeck code gives slightly better error performance for SNR less than 11.5 dB. For the AWGN channel, the Ungerboeck code performs slightly better than the code in this example for SNR less than 7 dB, and their error performance curves overlap with each other for SNR greater than 7 dB. This is due to the fact that the Ungerboeck code has slightly better squared Euclidean distance profile.

**Example 3.2:** In this example, we consider the construction of a two-level 8-PSK TCM code for the Rayleigh fading channel using two component codes, a convolutional code and a block code. The convolutional component code, denoted  $A_1$ , is a rate-1/2 code of constraint length 6 with generator matrix ( the fifth code in Table 1.A )

$$G(D) = [1 + D + D^4, D + D^2 + D^3 + D^5]$$

This code has minimum free *branch* distance 6, minimum free distance 7, and a 32-state trellis diagram. The block component code, denoted  $A_2$ , is the (32, 26, 4) RM code which has a four-section 16-state trellis diagram consisting of two parallel and structurally identical eight-state sub-trellis diagrams without cross connections between them.

At each time unit, the two code bits at the output of the convolutional code encoder form the first two label bits for an 8-PSK signal point, the block component code contributes the third label bit as shown in Figure 5. The resultant two-level 8-PSK code, denoted  $C(4) = \lambda[A_1 * A_2]$ , is a TCM code with the following parameters:  $\eta[C] = 1.8125$  bits/symbol, minimum symbol distance  $\delta_H[C] = 4$ , minimum product distance  $\Delta_p^2[C] = 256$ , and minimum free squared Euclidean distance  $d_{free}^2 = 5.516$ .

The error performance of this code over the Rayleigh fading channel with two-stage decoding is shown in Figure 4. It achieves an impressive 15.01 dB real coding gain over the uncoded QPSK at the bit error rate of  $10^{-3}$ . This coding gain is achieved at the expense of a 9.375% bandwidth expansion. The decoding complexity is reasonably simple.

## IV Conclusion

In this paper, we have shown that the multilevel coding method is a powerful technique for constructing modulation codes for both the AWGN and Rayleigh fading channels. Vari-

ous multilevel 8-PSK modulation codes have been constructed and they achieve significant coding gains with relatively simple decoding complexity.

## References

- [1] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error Correcting Codes," *IEEE Trans. on Information Theory*, Vol. IT-23, No. 3, 1977.
- [2] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Information Theory*, Vol. IT-28, No. 1, pp. 55-67, 1982.
- [3] E. Biglieri, D. Divsalar, P.J. McLane, and M.K. Simon, *Introduction to Trellis-Coded Modulation with Applications*, Macmillan, 1991.
- [4] C. Schlegel and D.J. Costello, Jr., "Bandwidth Efficient Coding for Fading Channels: Code Construction and Performance Analysis," *IEEE Journal Select Areas Communications*, Vol. SAC-7, No. 9, pp. 1356-1368, 1989.
- [5] V.V. Ginzburg, "Multidimensional Signals for a Continuous Channel," *Problemy Peredachi Informatsii*, Vol. 20, No. 1, pp. 28-46, 1984.
- [6] S.I. Sayegh, "A Class of Optimum Block Codes in Signal Space," *IEEE Trans. on Communications*, Vol. COM-30, No. 10, pp. 1043-1045, 1986.
- [7] J. Wu and S. Lin, "Multilevel Trellis MPSK Modulation Codes for the Rayleigh Fading Channel," *IEEE Trans. on Communications*, Vol. 41, No. 9, 1993.
- [8] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Multi-Level Block Modulation Codes," *IEEE Trans. on Information Theory*, Vol. IT-37, No. 4, 1991.
- [9] A.R. Calderbank, "Multi-Level Codes and Multistage Decoding," *IEEE Trans. on Communications*, Vol. COM-37, No. 3, pp. 222-229, 1989.
- [10] T. Takata, S. Ujita, T. Kasami and S. Lin, "Multistage Decoding of Multilevel Block M-PSK Modulation Codes and Its Performance Analysis," *IEEE Trans. on Information Theory*, Vol. 39, No. 4, 1993.
- [11] G. D. Forney, Jr., "Coset Codes II: Binary Lattices and Related Codes," *IEEE Trans. on Information Theory*, Vol. IT-34, pp. 1152-1187, September 1988, Part II.
- [12] T. Kasami, T. Takata, T. Fujiwara and S. Lin, "On Structural Complexity of the L-Section Minimal Trellis Diagrams for Binary Linear Block Codes," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, 1993.
- [13] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "A Concatenated Coded Modulation Scheme for Error Control," *IEEE Trans. on Communications*, Vol. COM-38, No. 6, pp. 752-763, 1990.
- [14] B. Vucetic, "Bandwidth Efficient Concatenated Coding Schemes for Fading Channels," *IEEE Trans. on Communications*, Vol. 41, No. 1, 1993.

**Table 1.A Optimum Branch Distance Rate 1/2 Codes,  
obtained by optimizing Branch Distance  
and Number of Nearest Neighbors**

| $\gamma^{\dagger}$ | G  | $d_{B\text{-free}}^{\ddagger}$ | $N_{B\text{-free}}^{\triangle}$ | $d_{H\text{-free}}^{\square}$ | $N_{H\text{-free}}^{*}$ |
|--------------------|--|--------------------------------|---------------------------------|-------------------------------|-------------------------|
| 1                  | $\begin{pmatrix} 4 \\ 2 \end{pmatrix}_8$       | 2                              | 1                               | 2                             | 1                       |
| 2                  | $\begin{pmatrix} 5 \\ 2 \end{pmatrix}_8$       | 3                              | 1                               | 3                             | 1                       |
| 3                  | $\begin{pmatrix} 5 \\ 64 \end{pmatrix}_8$      | 4                              | 1                               | 5                             | 1                       |
| 4                  | $\begin{pmatrix} 44 \\ 32 \end{pmatrix}_8$     | 5                              | 2                               | 5                             | 1                       |
| 5                  | $\begin{pmatrix} 62 \\ 35 \end{pmatrix}_8$     | 6                              | 2                               | 7                             | 3                       |
| 6                  | $\begin{pmatrix} 51 \\ 664 \end{pmatrix}_8$    | 7                              | 4                               | 8                             | 2                       |
| 7                  | $\begin{pmatrix} 344 \\ 532 \end{pmatrix}_8$   | 8                              | 6                               | 9                             | 2                       |
| 8                  | $\begin{pmatrix} 622 \\ 575 \end{pmatrix}_8$   | 8                              | 1                               | 10                            | 4                       |
| 9                  | $\begin{pmatrix} 355 \\ 6244 \end{pmatrix}_8$  | 9                              | 1                               | 11                            | 2                       |
| 10                 | $\begin{pmatrix} 3576 \\ 6322 \end{pmatrix}_8$ | 10                             | 3                               | 12                            | 2                       |

$\dagger$  : Total encoder memory

$\ddagger$  : Minimum free branch distance

$\triangle$  : Number of codewords with branch distance  $d_{B\text{-free}}$

$\square$  : Free Hamming distance

$*$  : Number of codewords with Hamming distance  $d_{H\text{-free}}$

Note: The generator matrix coefficients have been listed in octal, with the lowest degree coefficient on the left and the highest on the right, e.g.,  $(34)_8 \equiv D + D^2 + D^3$ .

**Table 1.B Optimum Branch Distance Rate 1/2 Codes,  
optimized for 4-PSK transmission  
over the Rayleigh fading channel**

| $\gamma^{\dagger}$ | G   | $d_{\text{B-free}}^{\ddagger}$ | $\Delta_p^2 \diamond$ |
|--------------------|---|--------------------------------|-----------------------|
| 1                  | $\begin{pmatrix} 4 \\ 2 \\ 8 \end{pmatrix}$     | 2                              | 8.0                   |
| 2                  | $\begin{pmatrix} 5 \\ 2 \\ 8 \end{pmatrix}$     | 3                              | 32.0                  |
| 3                  | $\begin{pmatrix} 7 \\ 24 \\ 8 \end{pmatrix}$    | 4                              | 64.0                  |
| 4                  | $\begin{pmatrix} 64 \\ 52 \\ 8 \end{pmatrix}$   | 5                              | 128.0                 |
| 5                  | $\begin{pmatrix} 66 \\ 37 \\ 8 \end{pmatrix}$   | 6                              | 128.0                 |
| 6                  | $\begin{pmatrix} 77 \\ 224 \\ 8 \end{pmatrix}$  | 7                              | 256.0                 |
| 7                  | $\begin{pmatrix} 552 \\ 364 \\ 8 \end{pmatrix}$ | 8                              | 1024.0                |
| 8                  | $\begin{pmatrix} 706 \\ 251 \\ 8 \end{pmatrix}$ | 8                              | 4096.0                |

$\diamond$  : Minimum product distance

**Table 2.A Optimum Branch Distance Rate 2/3 Codes,  
obtained by optimizing Branch Distance  
and Number of Nearest Neighbors**

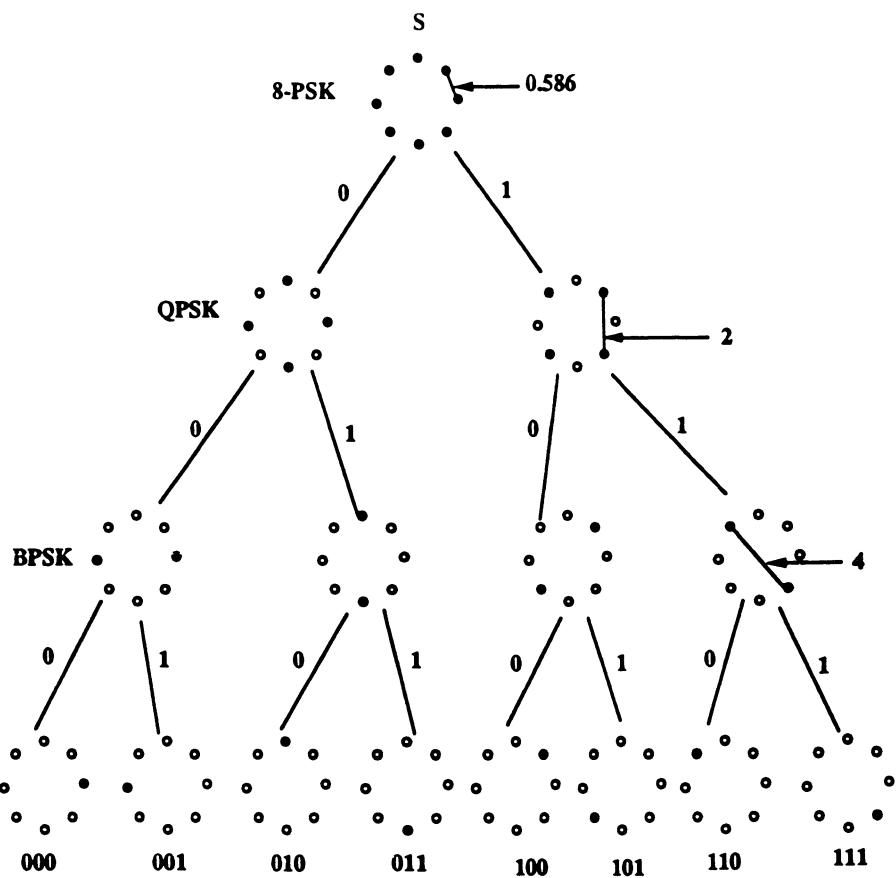
| $\gamma^{\dagger}$ | G   | $d_{B\text{-free}}^{\ddagger}$ | $N_{B\text{-free}}^{\Delta}$ | $d_{H\text{-free}}^{\ddagger}$ | $N_{H\text{-free}}^*$ |
|--------------------|---|--------------------------------|------------------------------|--------------------------------|-----------------------|
| 2                  | $\begin{pmatrix} 6 & 2 & 6 \\ 2 & 4 & 4 \end{pmatrix}_8$      | 2                              | 4                            | 3                              | 2                     |
| 4                  | $\begin{pmatrix} 0 & 4 & 3 \\ 7 & 5 & 0 \end{pmatrix}_8$      | 3                              | 5                            | 3                              | 1                     |
| 6                  | $\begin{pmatrix} 0 & 54 & 64 \\ 54 & 74 & 14 \end{pmatrix}_8$ | 4                              | 7                            | 6                              | 3                     |
| 8                  | $\begin{pmatrix} 76 & 26 & 46 \\ 64 & 0 & 36 \end{pmatrix}_8$ | 5                              | 14                           | 6                              | 1                     |
| 10                 | $\begin{pmatrix} 75 & 57 & 0 \\ 66 & 64 & 55 \end{pmatrix}_8$ | 6                              | 30                           | 6                              | 1                     |

**Table 2.B Optimum Branch Distance Rate 2/3 Codes,  
optimized for 8-PSK transmission  
over the Rayleigh fading channel**

| $\gamma^{\dagger}$ | G   | $d_{B\text{-free}}^{\ddagger}$ | $\Delta_p^{2.0}$ |
|--------------------|---|--------------------------------|------------------|
| 4                  | $\begin{pmatrix} 1 & 4 & 2 \\ 4 & 3 & 2 \end{pmatrix}_8$    | 3                              | 4.69             |
| 6                  | $\begin{pmatrix} 7 & 34 & 5 \\ 64 & 14 & 0 \end{pmatrix}_8$ | 4                              | 7.99             |

**Table 3 Optimum Branch Distance Rate 3/4 Codes**

| $\gamma^{\dagger}$ | G  | $d_{B\text{-free}}^{\ddagger}$ | $N_{B\text{-free}}^{\Delta}$ | $d_{H\text{-free}}^{\ddagger}$ | $N_{H\text{-free}}^*$ |
|--------------------|--|--------------------------------|------------------------------|--------------------------------|-----------------------|
| 3                  | $\begin{pmatrix} 0 & 6 & 6 & 2 \\ 6 & 6 & 2 & 4 \\ 6 & 2 & 2 & 2 \end{pmatrix}_8$        | 2                              | 11                           | 3                              | 3                     |
| 6                  | $\begin{pmatrix} 7 & 1 & 0 & 4 \\ 5 & 7 & 1 & 7 \\ 0 & 5 & 6 & 7 \end{pmatrix}_8$        | 3                              | 16                           | ;                              | 8                     |
| 9                  | $\begin{pmatrix} 74 & 2 & 34 & 0 \\ 44 & 7 & 74 & 74 \\ 54 & 0 & 4 & 74 \end{pmatrix}_8$ | 4                              | 30                           | 5                              | 1                     |



**Figure 1** The 8-PSK signal set and its partition chain 8-PSK/QPSK/BPSK

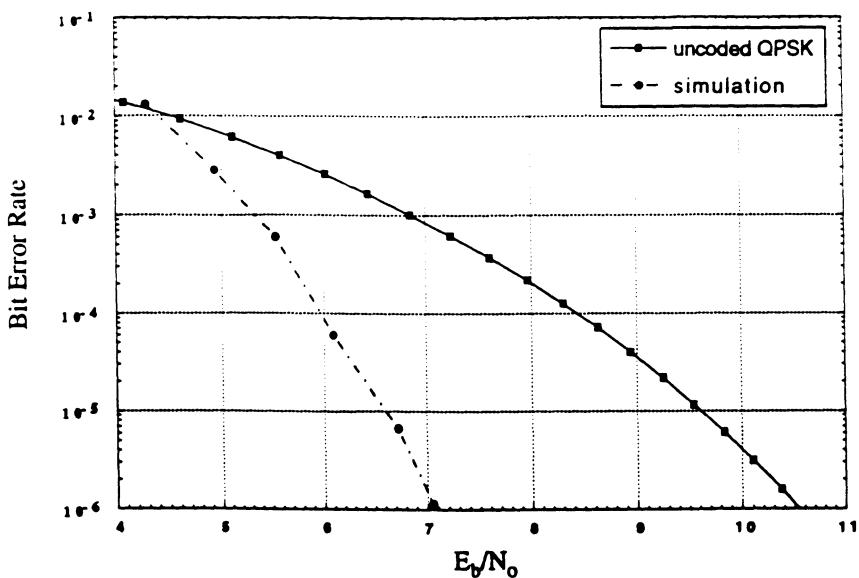


Figure 2 Bit-error performance of the basic 3-level modulation code C(1)

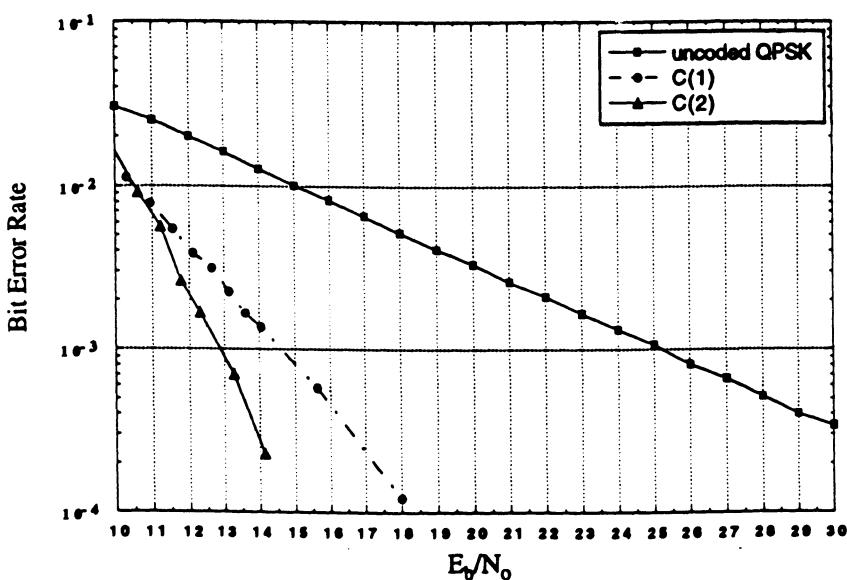


Figure 3 Bit-error performance of the basic 3-level modulation codes, C(1) and C(2)

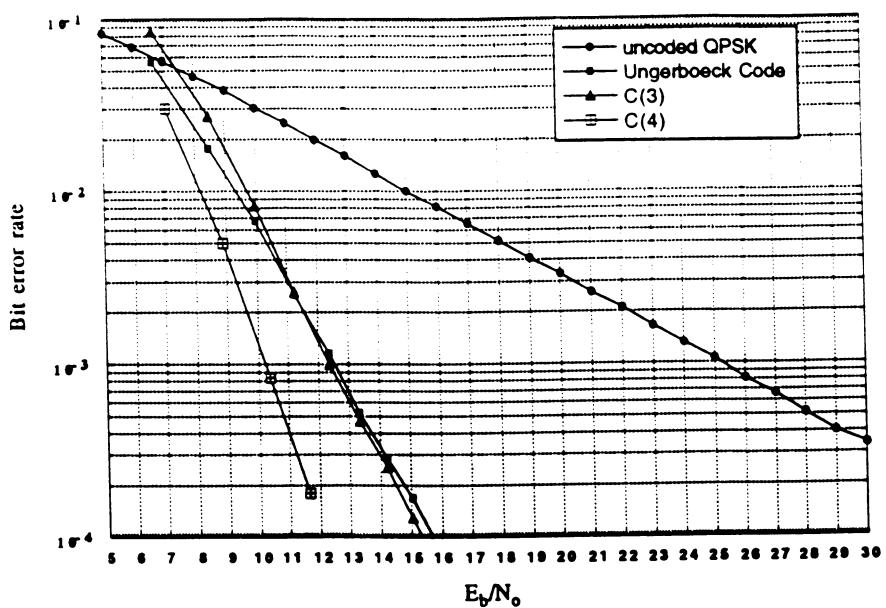


Figure 4 Bit error performance of the TCM codes, C(3) and C(4)

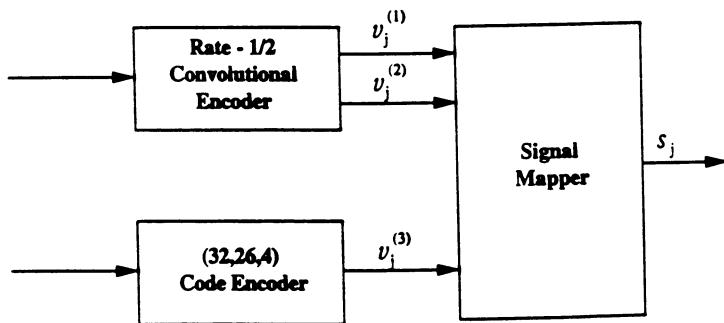


Figure 5 C(4)-Encoder

# On the Basic Averaging Arguments for Linear Codes

Hans-Andrea Loeliger  
ISY / Information Theory  
Linköping University  
S-58183 Linköping, Sweden

## Abstract

Linear codes over  $F_q$  are considered for use in detecting and in correcting the additive errors in some subset  $E$  of  $F_q^n$ . (The most familiar example of such an error set  $E$  is the set of all  $n$ -tuples of Hamming weight at most  $t$ .) In this set-up, the basic averaging arguments for linear codes are reviewed with emphasis on the relation between the combinatorial and the information-theoretic viewpoint. The main theorems are (a correspondingly general version of) the Varshamov-Gilbert bound and a ‘random-coding’ bound on the probability of an ambiguous syndrome. These bounds are shown to result from applying the same elementary averaging argument to two different packing problems, viz., the combinatorial ‘sphere’ packing problem and the probabilistic ‘Shannon packing’. Some applications of the general bounds are indicated, e.g., hash functions and Euclidean-space codes, and the connection to Justesen-type constructions of asymptotically good codes is outlined.

## I Introduction

This paper is essentially a tutorial review of the basic averaging arguments for linear codes. The main results that are proved are (a version of) the Varshamov-Gilbert bound and a ‘random-coding’ bound for linear codes.

This will hardly sound exciting — the mentioned venerable bounds belong to the very foundations of coding theory and have been proved and generalized in dozens of ways. What, then, is the purpose of this paper?

One of the origins of this paper is the deep confusion in which I once was put by the following ‘paradox’. Consider the binary symmetric channel with crossover probability  $\varepsilon$ , whose capacity is  $C \triangleq 1 - h(\varepsilon)$ , where  $h(\varepsilon) \triangleq -\varepsilon \log_2 \varepsilon - (1-\varepsilon) \log(1-\varepsilon)$  is the binary entropy function. It is well known from basic information theory that, for a fixed rate  $R$  less than (but arbitrarily close to)  $C$ , almost all binary linear codes of rate  $R$  and sufficiently large blocklength  $n$  have vanishingly small error probability on this channel. One could thus expect that, for  $R$  close to  $C$  and large  $n$ , most codes have a relative minimum distance  $d/n$  of about  $2\varepsilon$ . We would thus have  $R \approx 1 - h(d/2n)$ , which coincides with the asymptotic Hamming upper bound. However, the asymptotic Hamming bound is well known not to be achievable; in fact, asymptotically, almost all binary linear codes are known to have rates close to that of the Varshamov-Gilbert bound, viz.,  $R = 1 - h(d/n)$ .

This ‘paradox’ exposes a conceptual gap between information theory and combinatorial coding theory of which engineers should be aware, but over which most textbooks pass with silence. (A commendable exception is the classical text by Peterson and Weldon [1, Chapt. 4.3].)

Therefore, the primary purpose of this paper is to illuminate the relation between the information-theoretic and the combinatorial view of coding. We shall see that the Varshamov-Gilbert bound and a probabilistic random-coding bound are obtained from applying the same basic averaging argument to two different sphere packing problems, viz., the combinatorial packing of rigid spheres and the probabilistic ‘Shannon packing’, where the spheres are allowed to overlap slightly.

(It is appropriate here to mention that the intimate relation between the Varshamov-Gilbert bound and random coding à la Shannon was apparently first noticed in the Ph. D. thesis of Jim Massey [2].)

A basic feature of our exposition is that we will consider arbitrary sets  $E \subseteq F_q^n$  of additive error patterns, not just the standard case where  $E$  is the set of  $n$ -tuples of Hamming weight at most  $t$ . One motivation for this generality are applications such as burst error correction, multiaccess communications, hash functions, constrained codes, and Euclidean-space coding, which will be cursorily reviewed in Section IV. However, the consideration of an arbitrary error set  $E$  is believed to be valuable also from a purely pedagogical viewpoint.

The paper is structured as follows. In Section II, the essence of the averaging arguments of this paper is summarized in the form of a definition and three elementary lemmas. The core of the paper is Section III, where (a form of) the Varshamov-Gilbert bound and our probabilistic random-coding bound are derived. The proof of the latter — the only nontrivial proof of this paper — is so simple that it could well be used as an exercise in a first course on algebraic coding theory. These results are then discussed in Section IV, where also some applications are indicated and the connections to Justesen-type constructions of asymptotically good codes are outlined.

To conclude this introduction, I would like to mention that the material of this paper dates back to the time when I was a graduate student of Jim Massey, and he has repeatedly encouraged me to publish it. Here it is — happy birthday!

## II Three Averaging Lemmas

Let  $F_q$  denote the finite field with  $q$  elements. As usual, an  $(n, k)$  *q-ary linear code* is a  $k$ -dimensional subspace of the vector space  $F_q^n$  of  $n$ -tuples over  $F_q$ , and the *rate* of such a code is the fraction  $k/n$ . For any subset  $E$  of  $F_q^n$ , the set  $\{e \in E : e \neq 0\}$  of nonzero elements will be denoted by  $E^*$ .

The averaging arguments of this paper hold for every set of codes that is balanced in the following sense.

**Definition 1** *A nonempty set  $\mathcal{C}$  of linear  $(n, k)$  codes over  $F_q$  is balanced if every nonzero element of  $F_q^n$  is contained in the same number, denoted by  $N_{\mathcal{C}}$ , of codes from  $\mathcal{C}$ .*

It is rather obvious that, for fixed  $n$ ,  $k$ , and  $q$ , the set of *all* linear  $(n, k)$  codes over  $F_q$  is balanced. Further examples of balanced sets of codes will be given in Section IV.

The term ‘balanced’ stems from [3], where, however, it is used with a slightly different meaning, viz., as a property of a set of affine encoders (rather than of linear codes). Very similar sets of codes were also considered in [4]. It was noted in these references that balancedness is a combinatorial version of *pairwise-independence*, which is the key property underlying the usual random coding arguments, cf. [5, Chapt. 6.2]. For sufficiently symmetric channels (i.e., for ‘regular’ channels in the sense of [3]), the average error probability taken over a balanced set of codes is thus upper bounded by Gallager’s celebrated random coding bound [3], [5]. In Section III, we will see that a similar random coding bound can be derived very easily.

The multiplicity  $N_{\mathcal{C}}$  of every nonzero  $q$ -ary  $n$ -tuple in a balanced set  $\mathcal{C}$  of  $(n, k)$  linear codes is related to the number of codes  $|\mathcal{C}|$  by

$$|\mathcal{C}|(q^k - 1) = N_{\mathcal{C}}(q^n - 1). \quad (1)$$

(This is proved by counting the total number of nonzero codewords of all codes in  $\mathcal{C}$ , each with its multiplicity.) It will be useful to remember that, for all positive integers  $n, k$ , and  $q$  such that  $q > 1$  and  $k \leq n$ ,

$$\frac{q^n - 1}{q^k - 1} \geq \frac{q^n}{q^k} \quad (2)$$

and the inequality is strict for  $k < n$ .

**Lemma 1 (Basic Averaging Lemma)** *Let  $f(\cdot)$  be an arbitrary real valued<sup>1</sup> function defined on  $F_q^n$ ; let  $\mathcal{C}$  be a balanced set of linear  $(n, k)$  codes over  $F_q$ . Then the average, over all codes  $C$  in  $\mathcal{C}$ , of the sum  $\sum_{c \in C^*} f(c)$  (over all nonzero codewords) is given by*

$$\frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \sum_{c \in C^*} f(c) = \frac{q^k - 1}{q^n - 1} \sum_{v \in (F_q^n)^*} f(v).$$

**Proof:** It follows from the definition of a balanced set of codes that

$$\sum_{C \in \mathcal{C}} \sum_{c \in C^*} f(c) = N_{\mathcal{C}} \sum_{v \in (F_q^n)^*} f(v),$$

and the lemma follows from (1).  $\square$

We will use Lemma 1 primarily in the more special form of the following lemma.

**Lemma 2 (Average Intersection Cardinality Lemma)** *Let  $\mathcal{C}$  be a balanced set of linear  $(n, k)$  codes over  $F_q$ ; let  $E$  be an arbitrary subset of  $F_q^n$ . Then the average cardinality of  $C^* \cap E$  over all codes  $C$  in  $\mathcal{C}$  is given by*

$$\frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} |C^* \cap E| = \frac{q^k - 1}{q^n - 1} |E^*|. \quad (3)$$

**Proof:** Define  $f : F_q^n \rightarrow \{0, 1\}$  as  $f(v) = 1$  if  $v \in E$  and  $f(v) = 0$  otherwise and apply Lemma 1.  $\square$

---

<sup>1</sup>The range of  $f(\cdot)$  can actually be more general.

The last lemma of this section shows that, under a slightly stronger balancing condition for  $\mathcal{C}$ , the cardinality of  $C^* \cap E$  is close to the average value (3) for most codes  $C$  in  $\mathcal{C}$ .

**Lemma 3 (Intersection Variance Lemma)** *Let  $\mathcal{C}$  be a set of linear  $(n, k)$  codes over  $F_q$  that is doubly balanced, i.e., it is balanced and every pair of linearly independent elements of  $F_q^n$  is contained in the same number of codes from  $\mathcal{C}$ . Let  $E$  be an arbitrary nonempty subset of  $F_q^n$  with at least one nonzero element. Then*

$$\frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} |C^* \cap E|^2 - \left( \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} |C^* \cap E| \right)^2 < (q-1) q^{k-n} |E^*|.$$

It then follows from Chebyshev's inequality that the fraction of codes  $C$  in  $\mathcal{C}$  for which

$$\left| |C^* \cap E| - \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} |C^* \cap E| \right| \geq \gamma q^{k-n} |E^*|$$

is at most  $(q-1)^2/\gamma^2$ . Since Lemma 3 will not be referred to in the sequel, the proof is omitted and the interested reader is referred to [6].

### III Error Detection and Correction

Consider the following textbook situation. A transmitter selects a codeword  $c$  from an  $(n, k)$  linear code  $C$  over  $F_q$  and sends it over a noisy channel. The channel adds an error pattern  $e \in F_q^n$  to the transmitted codeword, and the task of the receiver is to estimate  $c$  from  $c + e$ .

It is clear that, for  $k > 0$ ,  $c$  cannot be recovered from  $c + e$  for all possible codewords  $c \in C$  and all possible error patterns  $e \in F_q^n$ . Therefore, we restrict our attention to a set  $E \subset F_q^n$  of typical error patterns (where ‘typical’ is meant informally). The most popular choice for  $E$  is the discrete ball  $S_{n,r} \triangleq \{v \in F_q^n : d(v, 0) \leq r\}$ , where  $d(\cdot, \cdot)$  denotes either Hamming distance or any other suitable metric on  $F_q^n$ , but the arguments below hold for arbitrary  $E \subseteq F_q^n$ .

We will use the notation

$$H_q(E) \triangleq 1/n \log_q |E| \tag{4}$$

for the ‘entropy’ of the error patterns  $E$  or of any other subset of  $F_q^n$ . (If a statistical channel model for the noise is available, then, for a reasonable choice of  $E$ , the set-theoretic entropy (4) is, of course, closely related to, and asymptotically identical with, the information-theoretic entropy of the noise. In fact, even in a purely combinatorial context, the set-theoretic entropy (4) can sometimes be closely approximated by the information-theoretic entropy of a suitable auxillary probability distribution, cf. [7, Sec. 8], [8].)

At this point, the purely combinatorial viewpoint and the probabilistic (information-theoretic) viewpoint begin to differ. We begin with the former.

We say that the code  $C$  *corrects* all errors in  $E$  if  $c$  (and thus also  $e$ ) can be recovered from  $c + e$  for all codewords  $c \in C$  and all error patterns  $e \in E$ . This is clearly impossible if  $q^k |E| > q^n$ , which is expressed in the following classical bound.

**Proposition 1 (Hamming Bound)** *Let  $E$  be an arbitrary nonempty subset of  $F_q^n$ ; let  $C \subseteq F_q^n$  be a code (not necessarily linear) with  $q^k$  codewords that corrects all errors in  $E$ . Then*

$$|E| \leq q^{n-k}$$

*or, equivalently,  $H_q(E) \leq 1 - k/n$ .*

It is convenient for the following argument to consider also error detection. A linear code *detects* all errors in a set  $E$  of error patterns if and only if  $C^* \cap E = \emptyset$ .

It is easily seen that a linear code  $C$  corrects all errors in  $E$  if and only if  $C \cap \Delta E = \{0\}$ , where  $\Delta E \triangleq \{e - e' : e, e' \in E\}$  is the set of all differences of elements of  $E$ . Thus  $C$  *corrects* all errors in  $E$  if and only if it *detects* all errors in  $\Delta E$ .

We now use Lemma 2 to obtain an existence proof for error detecting and correcting codes. Let  $\mathcal{C}$  be a balanced set of linear  $(n, k)$  codes over  $F_q$ , and let  $E$  be an arbitrary nonempty subset of  $F_q^n$ . If the right side of (3) is less than 1, then, on the average over all codes in  $\mathcal{C}$ ,  $|C^* \cap E| < 1$ , which implies the existence of at least one code  $C$  in  $\mathcal{C}$  that detects all errors in  $E$ ; a sufficient condition for this is  $|E^*| < q^{n-k}$ . We have proved the following theorem, which is one form of the classical Varshamov-Gilbert bound.

**Theorem 1 (Varshamov-Gilbert Bound)** *Let  $\mathcal{C}$  be a balanced set of linear  $(n, k)$  codes over  $F_q$  and let  $E$  be an arbitrary subset of  $F_q^n$  that contains the all zero vector. If*

$$|E| \leq q^{n-k} \tag{5}$$

*or, equivalently, if  $H_q(E) \leq 1 - k/n$ , then there exists a code in  $\mathcal{C}$  that detects all errors in  $E$ ; if*

$$|\Delta E| \leq q^{n-k} \tag{6}$$

*or, equivalently, if  $H_q(\Delta E) \leq 1 - k/n$ , then there exists a code in  $\mathcal{C}$  that corrects all errors in  $E$ .*

The following corollary is a bit closer to the usual formulations of the Varshamov-Gilbert bound.

**Corollary 1** *Let  $d(\cdot, \cdot)$  be a metric on  $F_q^n$  that satisfies  $d(v, v') = d(v - v', 0)$  for all  $v, v' \in F_q^n$  (i.e.,  $d(\cdot, \cdot)$  is translation invariant, as is, e.g., Hamming distance). Then there exists a code in  $\mathcal{C}$  with minimum distance  $d$  such that*

$$|S_{n,d}| > q^{n-k}. \tag{7}$$

**Proof:** Due to the assumed property for  $d(\cdot, \cdot)$ , it suffices to consider the distances from the all zero codeword. Assume that the largest minimum distance  $d$  of any code in  $\mathcal{C}$  satisfies  $|S_{n,d}| \leq q^{n-k}$ . The first part of Theorem 1 then implies the existence of a code  $C \in \mathcal{C}$  such that  $C^* \cap S_{n,d} = \emptyset$ . The minimum distance of  $C$  is thus larger than  $d$ , a contradiction.  $\square$

When applied to Hamming distance, Corollary 1 is slightly weaker than Gilbert's bound [9], which in turn is slightly weaker than Varshamov's bound [10, pp. 33–34]. However, all

forms agree asymptotically for  $n \rightarrow \infty$ , which is the case of primary interest. The minor nonasymptotic weakness of Corollary 1 seems to be the price for the generality of Theorem 1.

We now reconsider the situation from an information-theoretic (probabilistic) viewpoint. We assume that both the transmitted codeword  $\mathbf{c}$  and the error pattern  $\mathbf{e}$  are random variables (indicated by bold type); the former takes values in a  $q$ -ary linear code  $C$  and the latter takes values in  $F_q^n$ . We further assume that  $\mathbf{e}$  and  $\mathbf{c}$  are independent.

In this probabilistic set-up, Lemma 1 has an immediate interpretation for error detection.

**Proposition 2** *Let  $C$  be a balanced set of linear  $(n, k)$  codes over  $F_q$ . The arithmetic average, over all codes  $C$  in  $\mathcal{C}$ , of the probability  $P_{ue} \triangleq \sum_{c \in C^*} P(\mathbf{e} = c)$  of an undetectable error is given by*

$$\overline{P_{ue}} = \frac{q^k - 1}{q^n - 1} (1 - P(\mathbf{e} = 0)),$$

which implies  $\overline{P_{ue}} \leq q^{k-n}$ .

The published bounds of this type are usually restricted to particular additive channels such as, e.g., the binary symmetric channel [11], [12]. It is therefore remarkable that the upper bound  $q^{k-n}$  of Proposition 2 holds independently of the probability distribution of  $\mathbf{e}$ .

A comparison of Proposition 2 with the first part of Theorem 1 reveals a striking difference between the probabilistic viewpoint of the former and the combinatorial viewpoint of the latter: According to Theorem 1, error detection (with ‘average’ linear codes) costs  $H_q(E)$  in code rate, whereas Proposition 2 makes clear that, for  $n \rightarrow \infty$ , arbitrarily reliable error detection is possible with code rates arbitrarily close to one.

We now consider error correction. As before, we restrict our attention to a set  $E \subseteq F_q^n$  of typical, i.e., high probability error patterns. (Again, ‘typical’ is not meant formally — the derivation below holds for arbitrary sets  $E \subseteq F_q^n$ .)

The event that the received vector  $\mathbf{y} \triangleq \mathbf{c} + \mathbf{e}$  can be written in more than one way as  $\mathbf{y} = c + e$  with  $c \in C$  and  $e \in E$  will be called an *ambiguity*. Let  $P_{amb|E}$  denote the probability of an ambiguity, conditioned on the event that  $\mathbf{e}$  is in  $E$ . It is easily seen that further conditioning on the transmitted codeword does not change the probability of an ambiguity, i.e.,  $P_{amb|E} = P(\text{ambiguity} \mid \mathbf{e} \in E \text{ and } \mathbf{c} = c)$  for all  $c \in C$ .

Consider a decoder that operates according to the following rule. If  $\mathbf{y}$  has a unique decomposition  $\mathbf{y} = c + e$  with  $c \in C$  and  $e \in E$ , then  $\mathbf{y}$  is decoded to  $c$ . We need not specify the decoder action for any other case. For any such decoder, the probability  $P_e$  of a decoding error is bounded by

$$P_e \leq P_{amb|E} + P(\mathbf{e} \notin E). \quad (8)$$

In general, it is very difficult to compute  $P_{amb|E}$ . However, its average, over all codes of a balanced set of codes, is readily bounded as follows.

**Theorem 2 (Random Coding Bound)** *The arithmetic average, over all codes of a balanced set of  $q$ -ary linear  $(n, k)$  codes, of  $P_{amb|E}$  is bounded by*

$$\overline{P_{amb|E}} \leq q^{k-n} |E| \quad (9)$$

or, equivalently, by  $\overline{P_{amb|E}} \leq q^{-n[1-k/n-H_q(E)]}$ .

**Proof:** Since  $P_{amb|E}$  is independent of the transmitted codeword, we can assume that the all-zero codeword is transmitted. Any given received  $y$  has a unique decomposition  $y = c + e$ , with  $c = 0$  and  $e \in E$ , if and only if  $(y - E) \cap C = \{0\}$ . Therefore,  $P_{amb|E}$  is bounded by

$$P_{amb|E} \leq \sum_{y \in F_q^n} P(y = y \mid \mathbf{c} = 0 \text{ and } \mathbf{e} \in E) \cdot |C^* \cap (y - E)|,$$

which is the critical step of the proof. The rest is straightforward: averaging over all codes in  $\mathcal{C}$  and applying Lemma 2 yields

$$\begin{aligned} \overline{P_{amb|E}} &= \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} P_{amb|E} \\ &\leq \sum_{y \in F_q^n} P(y = y \mid \mathbf{c} = 0 \text{ and } \mathbf{e} \in E) \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} |C^* \cap (y - E)| \\ &= \sum_{y \in F_q^n} P(y = y \mid \mathbf{c} = 0 \text{ and } \mathbf{e} \in E) \frac{q^k - 1}{q^n - 1} |(y - E)^*| \\ &\leq q^{k-n} |E|. \end{aligned}$$

□

Note that Theorem 2 implies Shannon's channel coding theorem for all ergodic  $q$ -ary additive channels: if  $E$  are the typical error patterns (where, this time, 'typical' is meant more formally, cf. [13]), it is clear that both right-hand terms of (8) vanish for  $n \rightarrow \infty$  and fixed code rate  $k/n$ , provided only that the entropy  $H_q(E)$  of the errors tends to a limit below  $1 - k/n$ . Note, however, that Theorem 2 makes sense even without assuming ergodic errors and is in this sense more general than the usual information-theoretic random coding bounds.

## IV Discussion

The striking formal similarity among the results of Section III makes it easy to compare the combinatorial 'sphere' packing problem with the information-theoretic 'Shannon packing', as was promised in the introduction. Thereafter, we will sketch how the generality of the two theorems of Section III makes them useful for a variety of applications. Finally, some balanced classes of linear codes are briefly mentioned and the connection to the asymptotically 'good' codes of Justesen [14] and of Delsarte and Piret [3] is outlined.

### Shannon Packing vs. Rigid-Sphere Packing

How many Hamming spheres — or, more generally, arbitrary 'objects'  $E \subseteq F_q^n$  — can be packed into  $F_q^n$  such that they do not overlap? An obvious upper bound is the Hamming bound (Proposition 1), and the Varshamov-Gilbert bound (Theorem 1) gives a lower bound: if the spheres are centered on the codewords of a linear code, Theorem 1 gives the code rate where, on the average over all codes of a balanced set of codes, the intersection of the

spheres drops below one. The asymptotically best possible packing rate (for spheres) is not known in general — its determination is considered by mathematicians as one of the main goals of coding theory — but it is known to be closer to the Varshamov-Gilbert bound than to the Hamming bound.

A different packing problem was (implicitly) introduced — and solved — by Shannon: the requirement that the spheres do not intersect is relaxed to the condition that the volume of the intersection of any sphere with its neighbors is at most a fraction  $\varepsilon$  of the volume of the sphere. A very sharp and general solution to this ‘Shannon packing’ problem is Theorem 2, applied to the case that  $e$  is uniform over  $E$ : no matter how the objects  $E$  are shaped and for any positive  $\varepsilon$ , the asymptotically achievable packing rate coincides with that for cubes of the same volume, i.e., with the Hamming bound! Moreover, the same answer still holds if the interior of  $E$  is weighted by an arbitrary probability distribution.

It can not be overemphasized that the latter type of packing (i.e., the Shannon packing) is the more important one for most engineering applications. In particular, the restriction of most algebraic decoding algorithms to *spherical* decoding regions (i.e., bounded-distance decoding, cf. [15]) is one of the reasons for the limited practical usefulness of much of algebraic coding theory. (The other main problem of algebraic decoding is the well-known difficulty to use ‘soft-decision’ reliability information.) In fact, it is evident that, in all successful applications of coding to channels with moderate to high noise level, the decoding regions are far from being spherical; good examples are convolutional codes and concatenated codes of all kinds.

## Applications

The generality of Theorems 1 and 2 makes them useful in a large variety of applications. For the correction of burst errors or in similar situations, all that needs to be done is to specify the set  $E$  of error patterns and to evaluate its cardinality.

An interesting application are multiaccess systems. Any of a collection of users can lump the activity of the other users together with the channel noise into a set  $E$  of possible interference patterns. It is clear from Theorem 2 that the full sum capacity of one  $q$ -ary symbol per channel use is achievable if each user independently and randomly selects a code from a balanced set of codes.

The syndrome mapping  $F_q^n \rightarrow F_q^r$  of a linear  $(n, k = n - r)$  code can be viewed as a linear source encoder for the error patterns. In this way, any result on linear codes has an interpretation in source coding and vice versa, cf. [16].

The primary application area of linear source coding is the technique of hashing in computer science: a set  $E \subseteq F_2^n$  of ‘keys’ is coded into  $F_2^r$  by means of a hash function  $F_2^n \rightarrow F_2^r$ , where  $r < n$  is so small (typically in the range 8–16) that decoding can be done by table lookup. (Large keys are used, however, for cryptographic purposes.) In fact, results essentially equivalent to those of Section III have been published in the literature on hash functions, and our notion of a balanced set of linear codes is closely related to (the specialization to linear hash functions of) the notion of a universal class of hash functions, cf. [17].

Another application area is shaping of linear codes; i.e., we use a linear code  $C$  for error correction but allow only the codewords  $C \cap E_s$  that satisfy certain constraints, which are

specified by a set  $E_s \subset F_q^n$  of allowed words. For example, it is easily established from the arguments of this paper that constrained subcodes of linear codes can achieve any rate less than  $H(E_s) - H(E_e)$  on any  $q$ -ary channel with additive errors of entropy  $H(E_e)$ , cf. [6]. (The specialization of this result to the binary symmetric channel and runlength-limited codes has been reported in [18].)

The averaging arguments of this paper can also be applied to Euclidean-space codes and lattices. E.g., it is clear that Theorem 1, as well as Corollary 1, applies to  $M$ -PSK ( $M$ -ary phase-shift keying) when  $M$  is a prime and linear codes over  $F_M$  are used. In fact, it was shown in [19] that the Minkowski-Hlawka theorem — the basic asymptotic existence theorem for lattices — can be derived from Lemma 1 (and essentially the same proof is outlined in [20, pp. 534–535]). Further applications to Euclidean-space codes and lattices are given in [6].

## Balanced Classes of Codes and Explicit Constructions of Asymptotically Good Codes

An interesting method to construct balanced sets of linear codes over  $F_q$  is due to Delsarte and Piret [3]. The space  $F_q^n$  of  $q$ -ary  $n$ -tuples is identified with the field  $F_{q^n}$ . Let  $F_{q^k}$  be a subfield of  $F_{q^n}$ . For any nonzero  $v \in F_{q^n}$ , let  $\bar{v} \triangleq \{av : a \in F_{q^k}\}$  be the set of all  $F_{q^k}$  multiples of  $v$ , which is clearly a linear  $(n, k)$  code over  $F_q$ . But the set  $\mathcal{C} \triangleq \{\bar{v} : v \in F_{q^n}^*\}$  of these codes is a partition of  $F_{q^n}^*$  and therefore balanced with  $N_{\mathcal{C}} = 1$ .

This set  $\mathcal{C}$  was used in [3] for a Justesen-type concatenated code construction where the inner code varies over all codes in  $\mathcal{C}$  and the outer code is a Reed-Solomon code; it was shown that, for regular channels, such code constructions allow the derivation of an upper bound on error probability of the form  $P_e < q^{-NE(R)}$ , where  $N$  and  $R$  are the blocklength and the rate, respectively, of the concatenated code, and where the exponent  $E(R)$  is positive for all rates  $R$  less than channel capacity. (Since this construction relies on the goodness of the inner codes  $\mathcal{C}$ , the exponent  $E(R)$  is smaller than Gallager's exponent.)

We conclude the discussion of balanced sets of codes by pointing out that weaker versions of Theorems 1 and 2 can often be proved for sets of codes that are only ‘almost balanced’. In particular, an asymptotic version of Definition 1 suffices to prove the asymptotic Varshamov-Gilbert bound and information-theoretic random coding bounds. Apart from the celebrated codes from algebraic geometry [21] [20], all classes of codes that have so far been shown to satisfy the asymptotic Varshamov-Gilbert bound are of this type, e.g., alternant codes, (classical) Goppa codes, quasi-cyclic codes, self-dual codes [10], and shortened cyclic codes (cf. [22, Appendix II]). The last member in this list (i.e., the shortened cyclic codes) is seldom mentioned in textbooks but (for error detection) very popular in applications.

## V Conclusions

We have seen that the elementary averaging arguments for linear codes are amazingly versatile. They can be used both in a combinatorial way (which leads to Varshamov-Gilbert-type bounds) and in a probabilistic way (which leads to Shannon-type random coding theorems) and thereby illustrate the discrepancy between these two approaches to

coding. We have also seen that these averaging arguments, despite their simplicity, are able to yield nontrivial insights in a surprising variety of application areas.

## References

- [1] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed., Cambridge: MIT Press, 1972.
- [2] J. L. Massey, *Threshold Decoding*. Cambridge, Mass.: MIT Press, 1963.
- [3] Ph. Delsarte and Ph. Piret, ‘Algebraic constructions of Shannon codes for regular channels’, *IEEE Trans. Inform. Theory*, vol. 28, pp. 593–599, July 1982.
- [4] G. Séguin, ‘Linear ensembles of codes’, *IEEE Trans. Inform. Theory*, vol. 25, pp. 477–480, July 1979.
- [5] R. G. Gallager, *Information Theory and Reliable Communication*, New York: Wiley, 1968.
- [6] H.-A. Loeliger, ‘On the information-theoretic limits of lattices and related codes’, in preparation.
- [7] C. E. Shannon, ‘A mathematical theory of communication’, *Bell Syst. Techn. J.*, vol. 27, pp. 379–423, July 1948, and pp. 623–656, Oct. 1948. Reprinted in *Key Papers in the Development of Information Theory*, New York: IEEE Press, 1974.
- [8] H.-A. Loeliger, ‘An upper bound on the volume of discrete spheres’, submitted to *IEEE Trans. Inform. Theory*.
- [9] E. N. Gilbert, ‘A comparison of signalling alphabets’, *Bell Syst. Techn. J.*, vol. 31, pp. 504–522, May 1952. Reprinted in *Key Papers in the Development of Information Theory*, New York: IEEE Press, 1974.
- [10] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland, 1977.
- [11] J. L. Massey, ‘Coding techniques for digital data networks’, in *Proc. Int. Conf. Inform. Theory and Syst.*, NTG-Fachberichte, vol. 65, Berlin, Germany, Sept. 18–20, 1978.
- [12] J. K. Wolf, A. M. Michelson, and A. H. Levesque, ‘On the probability of undetected error for linear block codes’, *IEEE Trans. Comm.*, vol. 30, pp. 317–324, Feb. 1982.
- [13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [14] J. Justesen, ‘A class of constructive asymptotically good algebraic codes’, *IEEE Trans. Inform. Theory*, vol. 18, pp. 652–656, Sept. 1972. Reprinted in *Key Papers in the Development of Coding Theory*, New York: IEEE Press, 1974.

- [15] A. D. Wyner, ‘Capabilities of bounded discrepancy decoding’, *Bell Syst. Tech. J.*, vol. 54, pp. 1061–1122, 1965.
- [16] T. C. Ancheta, Jr., ‘Syndrome-source-coding and its universal generalization’, *IEEE Trans. Inform. Theory*, vol. 22, pp. 432–436, July 1976.
- [17] J. L. Carter and M. N. Wegmann, ‘Universal classes of hash functions’, *Journal of Computer and System Sciences*, vol. 18, pp. 143–154, 1979.
- [18] A. Patapoutian and P. V. Kumar, ‘The  $(d, k)$  subcode of a linear block code’, *IEEE Trans. Inform. Theory*, vol. 38, pp. 1375–1382, July 1992.
- [19] H.-A. Loeliger, ‘On existence proofs for asymptotically good Euclidean-space group codes’, Proc. of Joint DIMACS/IEEE Workshop on Coding and Quantization, Piscataway, NJ, USA, Oct. 19-21, 1992, to appear.
- [20] M. A. Tsfasman and S. G. Vlăduț, *Algebraic-Geometric Codes*, Kluwer, 1991.
- [21] M. A. Tsfasman, S. G. Vlăduț, and Th. Zink, ‘Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound’, *Math. Nachr.*, vol. 109, pp. 21–28, 1982.
- [22] T. Kasami, ‘An upper bound on  $k/n$  for affine-invariant codes with fixed  $d/n$ ’, *IEEE Trans. Inform. Theory*, vol. 15, pp. 174–176, Jan. 1969.

# Coding and Multiplexing

Hans J. Matt

ALCATEL Corporate Research Centre,  
70499 Stuttgart

Dedicated to James L. Massey on the occasion of his 60th birthday.

## Abstract

In transmission systems, different signals including audio, video and data signals need to be multiplexed over one common channel. Synchronization at the receiver must be obtained before the signal can be corrected and demultiplexed into its original components. We discuss how an integrated architecture of the multiplexer/demultiplexer can use error-correcting codes to combine the functions of error-detection, error-correction, and synchronization.

## I Prologue

My colleagues and I met Prof. J. L. Massey 1969 for the first time in the IEEE IT Transactions studying "Step by step decoding of BCH codes" [1] and "Shift register synthesis and BCH decoding" [2]. It was our introduction on how to find errors in a disturbed codeword. At the 1970 IEEE International Symposium on Information Theory, Noordwijk, our presentation about a two-stage FEC-system using an inner Golay (23,11,8) code for random error correction and an outer interleaved BCH code (15,7,5;  $b = 4$ ;  $\sim 2600$ ) for burst error correction up to and beyond the Rieger bound [3-4] happened to be chaired by J. L. Massey. It was a most exciting event getting acquainted with his warm personality. Early in 1978, J. L. Massey came to Berlin to stay for a few months at the Heinrich Hertz Institut für Nachrichtentechnik. Ideas used in [2-4] lead to "Determining the burst correction limit of cyclic codes" [5] and J. L. Massey gave lectures [6] during an IT symposium at the Technical University of Berlin. He was surprised to find most Berliners very familiar with information theory - throughout Berlin he had found everywhere the demand "Bitte ein Bit".

## II Coding and Multiplexing

In multimedia communication systems, audio, video, and data often need to be multiplexed over one common channel. At the receiver, synchronization must be obtained before the signal can be corrected and demultiplexed into its original components. Since error correcting codes can perform error detection, error-correction, and synchronization, it seems natural to base the architecture of a multiplexer-demultiplexer scheme on such codes.

## Multiplexer Architecture Based on Convolutional Codes

Consider the implementation of a Wyner-Ash  $kB/(k+1)B$  binary convolutional code [7]. A serial datastream is converted into parallel datablocks  $(c_0, c_1, \dots, c_{k-1})$  of  $k$  information bits each. For each such block of  $k$  information bits the encoder generates one parity bit  $c_k$  which is appended to the datablock to yield  $(C) = (c_0, c_1, \dots, c_k)$  with  $k+1$  bits; the parity bit depends on previous information blocks too.

To serialize the data the encoder of a Wyner-Ash convolutional code needs a parallel-to-serial converter which inherently is a multiplex function. Omitting the first serial-to-parallel converter yields a multiplexer architecture (Figure 1a) usable to multiplex data. Figure 1b shows the corresponding decoder-demultiplexer structure. The decoder converts the received datastream into parallel datablocks  $(R) = (r_0, r_1, \dots, r_k)$  of  $k+1$  bits, from which the first  $k$  bits are re-encoded to form a parity bit which being added to the received parity bit  $r_k$  yields one syndrome bit. A sequence of usually  $m = 1 + \lceil ld(k+1) \rceil$  syndrome bits coming out of stage  $s'_{m-1}$  forms a syndrome-vector  $(S)$  of dimension  $m$  to allow single error correction, i.e. to determine the address of a single error among its constraint length  $n = m \cdot (k+1)$ . The first nonzero syndrome bit indicates that an error occurred in the actual received datablock  $(r_0, r_1, \dots, r_k)$  whereas the subsequent  $m-1 = \lceil ld(k+1) \rceil$  syndrome digits then provide the “line-address” where the single error is located.

## Synchronization using Fixed Pattern Superposition

To achieve demultiplexer synchronization, a control circuit in Figure 1b watches the syndrome sequence  $(S)$ . If it contains enough digits zero over a certain interval  $q \cdot n$ , the decoder assumes correct synchronization. The principle of such a synchronization algorithm is depicted in Figure 2. If however, the input datablocks  $(C) = (c_0, c_1, \dots, c_k)$  contain long sequences of zeros causing the received datablocks  $(r_0, r_1, \dots, r_k)$  to equal  $(0)$ , synchronization will fail.

We would like to eliminate this weakness by adding a fixed vector  $(W)$  of length  $k+1$  (componentwise modulo-two) to the outgoing datablock  $(C)$  of the coder and subtracting it in the decoder after the serial-to-parallel converter. As long as the demultiplexer is synchronized this would not affect the received datablocks  $(R) = (r_0, r_1, \dots, r_k) = (0)$  nor the syndrome bits. For any wrong phase  $j = \{1, 2, \dots, k\}$  between multiplexer and demultiplexer however, we would like the syndrome outcome of stage  $s'_{m-1}(t, j = \text{constant} \neq 0)$  to equal one continuously, independent of time; if the demultiplexer stays in any of the  $k$  wrong positions while the input data signals  $(C)$  equal zero. Let  $W(x^j) \bmod (x^k - 1) = (W)_j$  then the output of  $s'_{m-1}$  depends on the received error pattern  $(E)_j$  generated by the difference between the  $j$  times cyclically shifted sum of  $\{(W) + (C)\}$  and  $(W)$ . Thus  $(E)_j = (W)_j + (C)_j - (W)$ . For  $j$  equal to a constant, the pattern  $(E)_j$  periodically repeats at the receiver, producing after at most  $m$  received words a constantly repeating syndrome bit.

To calculate the  $s'_{m-1}(j)$  we assume all registers  $s'_i(\cdot)$  of Figure 1b initially being zero. The values of register  $(S') = (s'_0, s'_1, \dots, s'_{m-1})$  are generated by a set of modulo-two additions taken on the received vector  $(r_0, r_1, \dots, r_k)$ . We describe this set of modulo-two additions for a particular stage  $i$  by a  $(k+1)$ -dimensional vector  $(A)_i = (a_{i,0}, a_{i,1}, \dots, a_{i,k})$

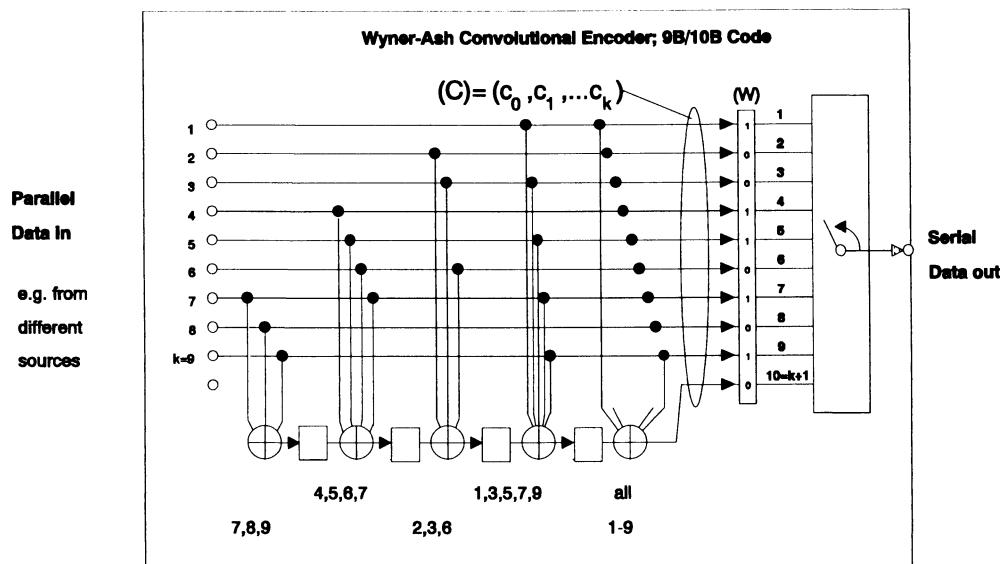


Fig. 1a Multiplexer Architecture based on Convolutional Encoder

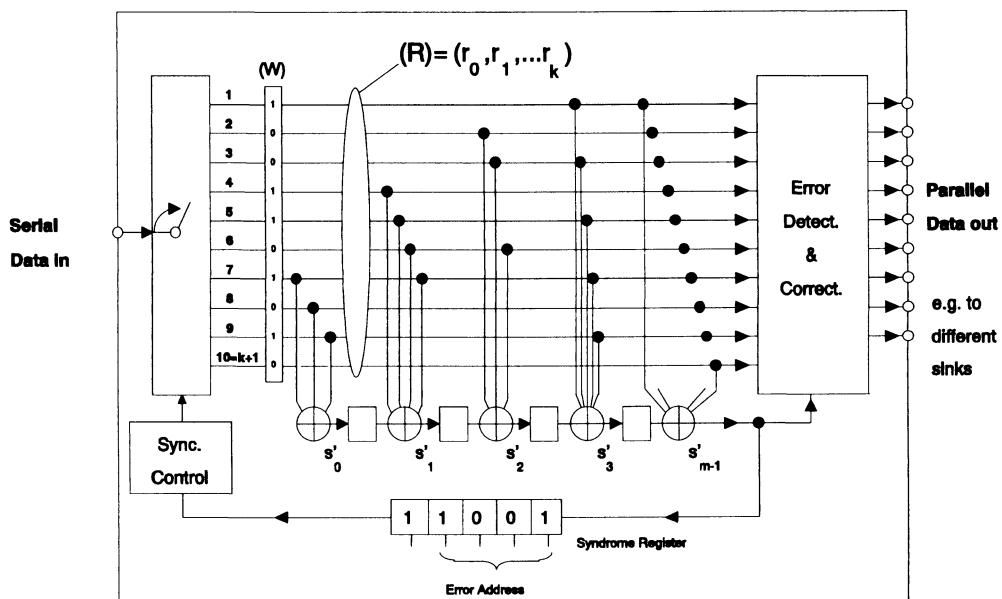


Fig. 1b Demultiplexer Architecture based on Convolutional Decoder

such that

$$\begin{aligned} s'_i(t) &= s'_{i-1}(t-1) + (r_0, r_1, \dots, r_k) \bullet (a_{i,0}, a_{i,1}, \dots, a_{i,k})^T \bmod 2 \\ &\quad \text{with } 0 \leq i \leq m-1; \quad 0 \leq j \leq k; \quad a_{i,j} = \{0, 1\} \end{aligned} \quad (1)$$

where  $a_{i,j} = 1$  if the element  $r_j$  contributes to calculate the  $s'_i$  in Equation (1), and  $a_{i,j} = 0$  otherwise. We then can create a *static check-vector* ( $H$ ) in summing up the  $(A)_i$ :

$$(H) = \sum_{i=0}^{m-1} (A)_i = (h_0, h_1, \dots, h_k) \quad (2)$$

Now the value of  $s'_{m-1}(j)$  can be calculated as

$$s'_{m-1}(j) = (H) \bullet (E)_j^T = (E)_j \bullet (H)^T = [(W)_j + (C)_j - (W)] \bullet (H)^T \quad (3)$$

and remains independent from  $t$  as long as  $(C) = (0)$  and the received vector  $(r_0, r_1, \dots, r_k) = (E)_j$  does not change over time. As the term  $s'_{m-1}(j)$  will equal 1 for all  $j \neq 0$  and  $s'_{m-1}(j=0) = 0$ , Equation (3) can be rewritten as a set of  $k$  linear equations given by

$$\begin{bmatrix} (W)_1 - (W) \\ (W)_2 - (W) \\ \vdots \\ \vdots \\ (W)_k - (W) \end{bmatrix} \bullet (H)^T = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} \quad (4)$$

In order to find solutions for Equation (4) let

$$(W) \bullet (H)^T = 0 \{ \text{ or } 1 \} \text{ and } (W)_j \bullet (H)^T = 1 \{ \text{ or } 0 \} \quad \text{for } 1 \leq j \leq k. \quad (5)$$

observing that  $(W)_j \bullet (H)^T = (W) \bullet (H)_{-j}^T = (H)_{-j} \bullet (W)^T = 1 \{ \text{ or } 0 \}$ , we can rewrite Equation (5) to yield

$$\begin{bmatrix} (H) \\ (H)_{-1} \\ \vdots \\ \vdots \\ (H)_{-k} \end{bmatrix} \bullet (W)^T = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \quad (6)$$

The term  $(W) \bullet (H)^T = 0 \{ \text{ or } 1 \}$  means a repeating vector  $(W)$  must be a codeword whereas the  $(W)_j$  must not, and vice versa. Vector  $(W)$  is of size  $k+1$  but Equation (6) may have lower dimension. Thus the matrix in (6) can easiest be handled via triangularization. If less than  $k+1$  linear independent equations become available, there may be more than one solution for  $(W)$  if there is any. Analyzing Equations (4), (5), and (6) for some practical convolutional codes leads to the following results:

1. There is no solution for  $(W)$  if Equation (6) contains contradictions; i.e. if two identical expressions on its left side demand different values 0 and 1 on the right side.

2. There are pairs of solutions for  $(W)$  if there are any; these contain  $(W)$  and its binary inverse  $(W)'$ . *Proof:* Substituting  $(W)$  by  $(W)'$  and  $(W)_j$  by  $(W)'_j$  in Equation (4) we find it satisfied because  $(W)' = (W) + (1, \dots, 1)$  and  $(W)'_j = (W)_j + (1, \dots, 1)_j$ .
3. For all perfect codes with  $(k+1) = 2^{m-1}$ , there exists no solution for  $(W)$  because the syndrome  $(1, \dots, 1)$  is already assigned to one error pattern.
4. But there is always a way to find a vector  $(W)$  even for the singular cases, e.g. by adding one more stage with just one more tap to the code. This, however, changes the code into a nonperfect one.
5. Given a valid  $(W^{(1)})$  for which  $(W^{(1)}) \bullet (H)^T = (0)$  holds, and an input  $(C) = (1, \dots, 1)$ . The demultiplexer synchronization would not fail if  $(H)$  has even weight; if  $(H)$  has odd weight  $(C) = (1, \dots, 1)$  is not a codeword and hence cannot occur. *Proof:* Assume repeating input sequences  $(C) \neq 0$  being introduced in Equation (3). Assume  $(C) = (1, \dots, 1) = (C)_j$ ; then  $s'_{m-1}(j) = 1$  requires  $(C)_j(H)^T = (1, \dots, 1) \bullet (H)^T = 0$  which means  $(C)$  must be a codeword and vector  $(H)$  must have even weight.
6. There remain some input sequences  $(C^*) \neq 0$  that cause the demultiplexer synchronization to fail. Such  $(C^*)$  may be obtained from Equation (3) assuming  $(W) \bullet (H)^T = 0 \{ \text{ or } 1 \}$ . They are of the form  $(C^*)_j + (W)_j - (W) = (C)$  with  $(C)$  being a codeword. That is, if  $(C^*) = (W) \{ \text{ or } (W)' \}$  then synchronization will fail.

**Example:** For the 9B/10B code the vectors  $(A)_i$  and  $(W) = (1001101010)$  are shown in Figure 2. Here vector  $(H)$  has the form  $(H) = (1100110100)$ ; hence  $(H)$  has odd weight and  $(C) = (1, \dots, 1)$  is not a codeword.

## Synchronization Recovery

The state-diagram given in Figure 2 is suitable to describe a set of synchronization recovery algorithms. It owns at least two main states: a) system being synchronized and b) system being out of synchronization. (Further states are possible.)

- From the state “out of sync” a first algorithm searches to find synchronization. With the previous syndrome decoder in mind it appears sufficient to count the events  $(S) = (0)$  and to check any event  $(S) \neq (0)$ . If (after  $m$  vectors  $(R)$  being received)  $Z_1 = 0$  then phase  $j$  is shifted by 1. If  $Z_1$  reaches the value  $T_1$ , synchronization is achieved with high probability  $P(\cdot)$ ; the algorithm then enters into the state “system in sync”.
- From the state “system in sync.” a second algorithm searches to detect loss of synchronization. Again it suffices to count the events  $(S) = (0)$  in upward and the events  $(S) \neq (0)$  in downward direction. If  $Z_2 = 0$ , loss of synchronization is assumed and the algorithm returns into the state “out of sync.”.

From such a state architecture and known algorithms a number of parameters can be calculated, such as the average time to achieve synchronization, the probability to lose synchronization, the average time to detect loss of synchronization, etc., all parameters as functions of the channel bit error rate.

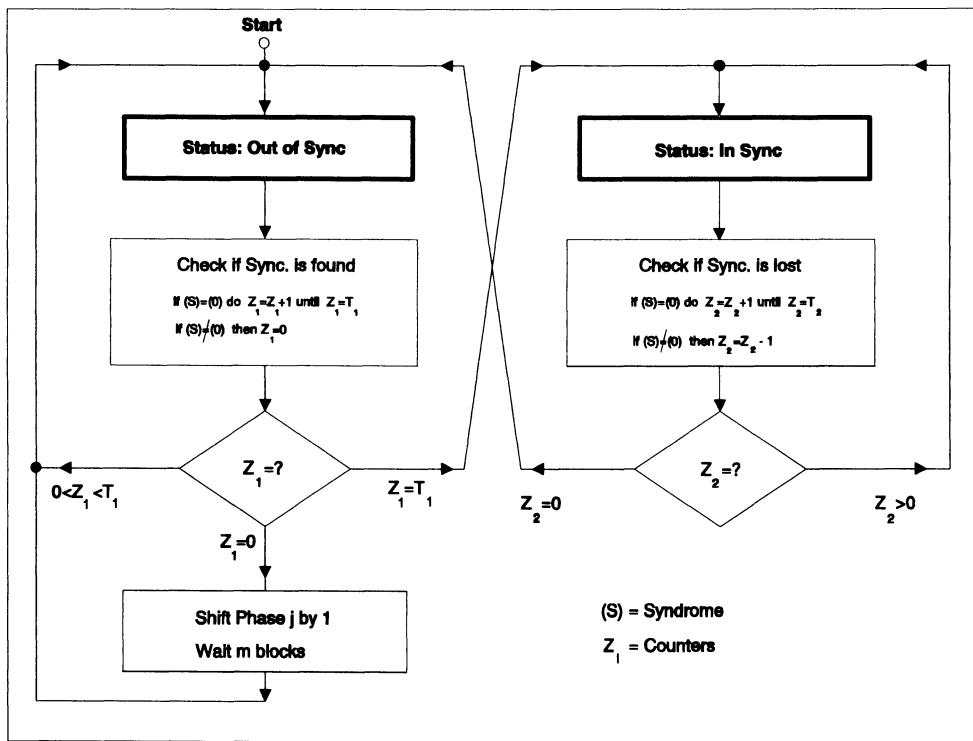


Fig. 2 Synchronization-Algorithm

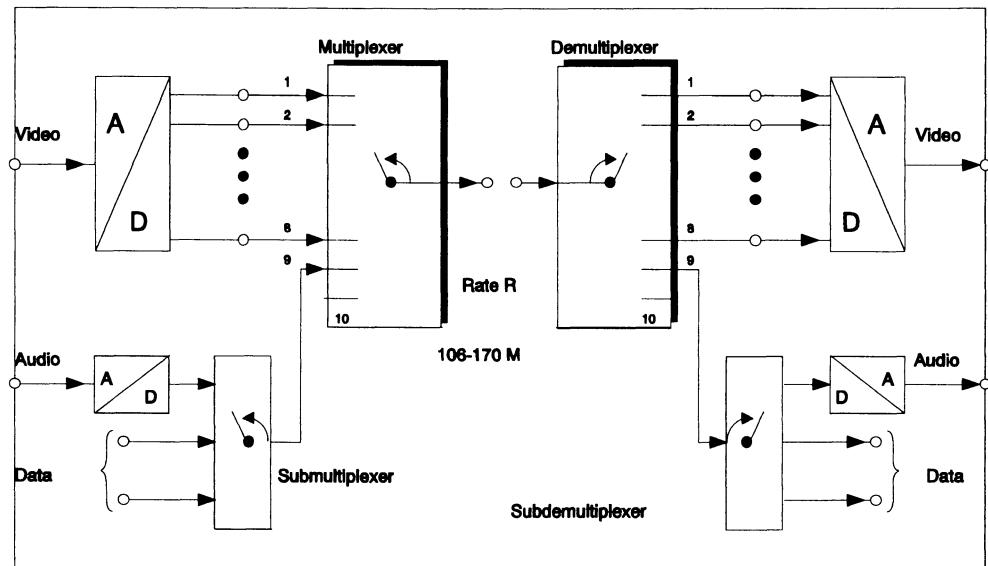


Fig. 3 Integrated Code-Multiplexer Module for Audio-, Video- and Data-Signals

## Multiplexer Architectures based on Block Codes

The architecture of a multiplexer/demultiplexer can also be based on block codes in a similar way to what has been described for convolutional codes. For this purpose the actual blocklength  $n$  of the code may be a multiple of the multiplexer inputs  $q$  with ( $n = pq$ ) preferably odd integers. For block codes a fixed vector ( $W$ ) of size  $n$  can easily be determined to yield  $(S) \neq (0)$  for all wrong phases  $j$  ( $1 \neq j \neq q$ ) of the demultiplexer. We simply demand the vector ( $W$ ) not to be a codeword of the code nor shall the expression  $(W)_j + (C)_j - (W) = (E)_j$  become a codeword. For cyclic block codes this can easily be achieved in writing ( $W$ ) as the sum of any codeword ( $C$ ) and any low weight (error-) pattern ( $ER$ ), such that

$$(W) = (C) + (ER); \text{ with } 0 \leq w(ER) < d/2 \quad (7)$$

and

$$(ER) \neq (ER)_i \quad \text{for all } i \text{ with } (1 \leq i \leq n - 1).$$

If  $0 < w(ER) < d/2$  then  $(ER), (ER)_j\}$  and their sum are nonzero and are not codewords and because all such vectors cannot exceed the weight  $(d - 1)$  and hence are not codewords. If we choose  $w(ER) < d/4$ , then the weight  $w\{(ER) + (ER)_j\} < d/2$  and we could “decode” the actual wrong phase position  $j$  of the demultiplexer from the syndrome of  $(E)_j$ .

## III Multiplexer Architecture for Audio, Video, and Data

In many applications signals with much different rates (e.g. video, audio, telemetry) are multiplexed, which may require a two-stage multiplexing scheme with a main multiplexer and a submultiplexer. The submultiplexer splits one input line of the main multiplexer into several sub-inputs operating at lower speed (Figure 3). One can design the architecture for synchronizing such demultiplexers in different ways: a) allowing each demultiplexer stage to perform its own synchronization independent of the other, or b) by a concatenated scheme that performs synchronization for both stages in a single step. In the latter case it suffices to have generation and evaluation of code redundancy on the submultiplexer/demultiplexer level if the main multiplexer data need no error protection.

An example of an integrated audio, video, and data module including A/D converters, (Figure 3) is based on the architecture of a two-stage multiplexing scheme using a Wyner-Ash convolutional code similar to Figure 1. The main multiplexer uses  $k + 1 = 10$  digits. As the received data ( $R$ ) are split on ten outputs their speed  $R$  is reduced to  $R/10$ ; this rate serves as sampling frequency to an eight-bit video D/A converter. Its eight input lines receive their data from the first eight outputs of the main demultiplexer. The rate  $R$  can be *lower bounded* by the sampling theorem, i.e. by the bandwidth  $B$  of the (composite video) signal that will be digitized and transmitted without aliasing errors:

$$R = 10 \text{ bit} \bullet f_s; \quad f_s > 2B \quad (8)$$

$$R > 20 \bullet B \text{ bit}$$

$R$  is upper bounded by the maximum allowable speed of the circuits such as A/D-converters. Such an audio/video data module can be operated in a wide range of transmission rates from  $\sim 106 \dots 170$  Mbit/s. Input nine of the main multiplexer is connected to a

submultiplexer, for transmission of stereo audio signals (at a rate of  $R/100$ ) and data (at  $R/50$ ). The main characteristics of the module [8-9] are summarized in the table below.

|                                |   |
|--------------------------------|---|
| <b>Channel Rate</b>            | $R \sim 106 \dots 170$ Mbits/s  |
| <b>Video Signal</b>            | Composite CVBS (NTSC, PAL, SECAM)   |
| - video bandwidth              | $\sim 5$ MHZ  |
| - sampling rate                | $R/10$  |
| - bits / sample                | 8   |
| - $(S/N)_{eff.,weighted}$      | $\sim 55$ dB  |
| <b>Audio Signal</b>            | Analog, Stereo (R,L)  |
| - bandwidth                    | $\geq 15$ KHZ; ripple $\leq 1$ dB   |
| - $(S/N)$                      | $\sim 75$ dB  |
| - $a_k$                        | $\leq 0.1\%$  |
| <b>Auxiliary Data</b>          | $5 \bullet R/50$ and $2 \bullet R/100$  |
| <b>Supervision and Control</b> | synchronization recovery<br>error detection<br>error correction: 1 among 100 bits |

## References

- [1] J. L. Massey, "Step by step decoding of the Bose-Chaudhuri-Hocquenghem codes". *IEEE Trans. on Inform. Theory*, vol. IT-11, No.4, pp. 580-585, 1965.
- [2] J. L. Massey, "Shift register synthesis and BCH decoding". *IEEE Trans. on Inform. Theory*, vol. IT-15, pp. 122-127, 1969.
- [3] H. J. Matt and M. Prögler, "A generalization of permutation decoding and its combination with error pattern superposition". *IEEE Intern. Symp. on Inform. Theory*, Noordwijk, 1970.
- [4] U. Haller, H. J. Matt, and M. Prögler "A forward error correction system...." *The Radio and Electronic Engineer*, vol. 42, no. 12, pp. 523-530, 1972.
- [5] H. J. Matt and J. L. Massey, " Determining the burst-correcting limit of cyclic codes". *IEEE Trans. on Inform. Theory*, vol. IT-26, no.3, pp. 289-297, 1980.
- [6] J. L. Massey, "Information Theory and its Applications to Digital Communication Networks". Invited Lectures at Int. Conf. on Inform. and System Theory in Digit. Comm.; Techn. University Berlin, Sept. 18-20, 1978, NTG Band 65, VDE-Verlag.
- [7] W. W. Peterson, and E. J. Weldon, *Error Correcting Codes*. MIT Press 1972.
- [8] H. J. Matt, "Video-Audio Codec". Patent Application P 4302428.9 ; Jan. 29, 1993.
- [9] H. J. Matt, Natürliche Kommunikation durch Breitbandkommunikation: Freisprechen, Bildfern sprechen, Videokonferenz und Multimedia-Kommunikation. Taschenbuch der Telekom Praxis; Fachverlag Schiele and Schn GmbH, Berli, 1993.

# The Strong Secret Key Rate of Discrete Random Triples

Ueli M. Maurer

Institute for Theoretical Computer Science

ETH Zürich

CH-8092 Zürich, Switzerland

Dedicated to James L. Massey on the occasion of his 60th birthday.

## Abstract

Three parties, Alice, Bob, and Eve, know the sequences of random variables  $X^N = [X_1, X_2, \dots, X_N]$ ,  $Y^N = [Y_1, Y_2, \dots, Y_N]$  and  $Z^N = [Z_1, Z_2, \dots, Z_N]$ , respectively, where the triples  $(X_i, Y_i, Z_i)$ , for  $1 \leq i \leq N$ , are generated by a discrete memoryless source according to some probability distribution  $P_{XYZ}$ . Motivated by Wyner's and Csiszár and Körner's pioneering definition of, and work on, the secrecy capacity of a broadcast channel, the secret key rate of  $P_{XYZ}$  was defined by Maurer as the maximal rate  $M/N$  at which Alice and Bob can generate secret shared random key bits  $S_1, \dots, S_M$  by exchanging messages over an insecure public channel accessible to Eve, such that the rate at which Eve obtains information about the key is arbitrarily small, i.e., such that  $\lim_{N \rightarrow \infty} I(S_1, \dots, S_M; Z^N, C^t)/N = 0$ , where  $C^t$  is the collection of messages exchanged between Alice and Bob over the public channel. However, this definition is not completely satisfactory because only the rate, but not the total amount of information about the key obtained by Eve is bounded. This paper introduces and investigates the *strong* secret key rate: it is required that the total amount of information about the key obtained by Eve be negligible, i.e.  $\lim_{N \rightarrow \infty} I(S_1, \dots, S_M; Z^N, C^t) = 0$ , and that  $[S_1, \dots, S_M]$  be arbitrarily close to uniformly distributed, i.e.  $\lim_{N \rightarrow \infty} M - H([S_1, \dots, S_M]) = 0$ . Using novel results on privacy amplification by Bennett, Brassard, Crépeau and Maurer we demonstrate that the known results for the secret key rate also hold for the stronger definition.

## I Introduction

Unlike a communications engineer who can prove the quality of a designed communication system for a given noisy channel simply by demonstrating the error-free transmission of information at a specified rate, a cryptographer is usually in a much less comfortable position. He (or she) can usually only affirm that the state-of-the-art in cryptography and in cryptanalysis has been taken into account in the design of a system, but is not able to

*prove* the security of the system. It is conceivable that a cipher gets broken shortly after it was designed and, to make things even worse, it is possible that such a failure will not even become known to the designer or users of a system. No presently-used ciphers (except the one-time pad that is used in rare applications where security is paramount), including public-key cryptosystems, can be proven secure.

In his research, Jim Massey has always attacked the fundamental question behind a given problem and refrained from going for the more promising, but less exciting goal of making minor contributions along a path other researchers had previously taken. On our trip to Eurocrypt '86 held in Linköping (shortly after my entrance into Sweden had caused severe complications because my passport was expired, and it was only due to Lis Massey's diplomatic intervention that finally an exception was made), Jim explained to me the strong need for rigorous proofs in cryptography and asked me whether I would like to accept the challenge of working towards provable security in cryptography as the topic of my doctoral research. This challenge struck me immediately and has never since ceased to drive my research. As a doctoral student I had the invaluable opportunity to work with Jim on various aspects of provable security (cf. [12], [15], [16]). I am deeply grateful for his careful guidance and for demonstrating, as an outstanding example, how rewarding and enjoyable it can be to work in an academic environment.

This paper is concerned with *provable* security in cryptography. More precisely, we try to beat Shannon's bound [18] for perfect secrecy which states that a cipher can only be perfect, i.e., plaintext and ciphertext can only be statistically independent, if the entropy of the plaintext is at most equal to the entropy of the secret key. Shannon's bound applies only when an opponent can, except for the secret key, see precisely the same information as the legitimate receiver. This assumption is overly pessimistic in many situations and we therefore consider a scenario in which two parties, called Alice and Bob, exploit knowledge of some correlated random variables about which an opponent Eve also has partial information. Alice and Bob, who share no secret key initially, can generate a secret key by communicating only over an insecure channel, even when Eve has more information than Bob about Alice's random variable and also more information than Alice about Bob's random variable. Eve has essentially no information about the finally shared secret key which can thus be used as the key in a one-time pad system [19] to transmit messages in perfect secrecy.

## II Secret Key Agreement by Public Discussion

In this section we describe the general scenario investigated in this paper, which was first suggested in [13] and independently in [1]. The purpose of this paper is to derive more powerful results for the same scenario.

Consider the following general key agreement problem. Assume that Alice, Bob and Eve know random variables  $X$ ,  $Y$ , and  $Z$ , respectively<sup>1</sup>, with joint probability distribution  $P_{XYZ}$ , and that Eve has no information about  $X$  and  $Y$  other than through her knowledge of  $Z$ . More precisely,  $I(XY; U|Z) = 0$  where  $U$  summarizes Eve's complete information about the universe.  $X$ ,  $Y$ , and  $Z$  take on values in some finite alphabets  $\mathcal{X}$ ,  $\mathcal{Y}$ , and  $\mathcal{Z}$ , respectively.

---

<sup>1</sup>Sequences of random variables as described in the abstract will be considered later.

Alice and Bob share no secret key initially, but are assumed to know  $P_{XYZ}$  or at least an upper bound on the correlation between  $Z$  and  $X$  and  $Y$ . Eve is assumed to know everything about the protocol used by Alice and Bob. Every message communicated between Alice and Bob over an insecure channel can be intercepted by Eve, but it is assumed that Eve cannot insert fraudulent messages nor modify messages on this public channel without being detected. In a scenario where Eve is not restricted to passive eavesdropping, an unconditionally-secure authentication scheme with a short initially shared secret key [20] can be used to detect active tampering with messages with very high probability. In this case, our protocols can be viewed as a method for expanding a short secret key rather than generating a key from scratch. If only a computationally-secure authentication scheme were used, the unconditional security would only be retained against passive, but not against active wire-tapping.

A realistic scenario for the generation of random variables  $X$ ,  $Y$ , and  $Z$  is by using a satellite broadcasting random bits to the earth at a very low signal power. Alice, Bob, and Eve can receive the bits over partially independent channels with certain bit error probabilities  $\epsilon_A$ ,  $\epsilon_B$  and  $\epsilon_E$ , respectively. Eve's channel must be assumed to be imperfect (i.e.,  $\epsilon_E > 0$ ) for Alice and Bob to be able to generate a secret key, but as demonstrated in Section V of [13], it is neither required that  $\epsilon_E > \epsilon_A$  nor that  $\epsilon_E > \epsilon_B$ . In fact, the capacity of Eve's channel can be allowed to be significantly (e.g., a thousand times) larger than the capacities of Alice's and Bob's channel. In this paper we are not concerned with particular such situations, but we rather investigate the rate at which Alice and Bob can generate secret key in a scenario where either  $I(X; Z) > I(X; Y)$  or  $I(Y; Z) > I(Y; X)$ . In fact, we demonstrate that any such difference can be fully exploited.

Alice and Bob use a protocol in which at each step either Alice sends a message to Bob depending on  $X$  and all the messages previously received from Bob, or vice versa (with  $X$  replaced by  $Y$ ). Without loss of generality, we consider only protocols in which Alice sends messages at odd steps ( $C_1, C_3, \dots$ ) and Bob sends messages at even steps ( $C_2, C_4, \dots$ ). Moreover, we can restrict the analysis to deterministic protocols since a possible randomizer which Alice's and Bob's strategy and messages might depend on can be considered as part of  $X$  and  $Y$ , respectively. In other words, Alice and Bob can, without loss of generality, extend their known random variables  $X$  and  $Y$ , respectively, by random bits that are statistically independent of  $X$ ,  $Y$ , and  $Z$ . At the end of the  $t$ -step protocol, Alice computes a key  $S$  as a function of  $X$  and  $C^t \triangleq [C_1, \dots, C_t]$  and Bob computes a key  $S'$  as a function of  $Y$  and  $C^t$ . Their goal is to maximize  $H(S)$  under the conditions that  $S$  and  $S'$  agree with very high probability and that Eve has very little information about either  $S$  or  $S'$ . More formally we have

$$H(C_i | C^{i-1} X) = 0 \quad (1)$$

for odd  $i$ ,

$$H(C_i | C^{i-1} Y) = 0 \quad (2)$$

for even  $i$ ,

$$H(S | C^t X) = 0 \quad (3)$$

and

$$H(S' | C^t Y) = 0, \quad (4)$$

and it is required that

$$P[S \neq S'] \leq \epsilon \quad (5)$$

and

$$I(S; C^t Z) \leq \delta \quad (6)$$

for some specified (small)  $\delta$  and  $\epsilon$ .

If one requires that  $P[S \neq S'] = 0$  and  $I(S; C^t) = 0$  (i.e., that  $\epsilon = 0$  in (5) and  $\delta = 0$  in (6)) it appears intuitive but not obvious that  $I(X; Y)$  is an upper bound on  $H(S)$ . It appears to be similarly intuitive that  $H(S) \leq I(X; Y|Z) = I(XZ; YZ) - H(Z)$  because even under the assumption that Alice and Bob could learn  $Z$ , the remaining information shared by Alice and Bob is an upper bound on the information they can share in secrecy. It was proved in [13] that for every key agreement protocol satisfying (1)-(4),

$$H(S) \leq \min[I(X; Y), I(X; Y|Z)] + H(S|S') + I(S; C^t Z), \quad (7)$$

and hence, by Fano's lemma (cf. [4], p. 156) and conditions (5) and (6), that

$$H(S) \leq \min[I(X; Y), I(X; Y|Z)] + \delta + h(\epsilon) + \epsilon \log_2(|\mathcal{S}| - 1),$$

where  $|\mathcal{S}|$  denotes the number of distinct values that  $S$  takes on with nonzero probability. It is worth pointing out that  $I(X; Y) < I(X; Y|Z)$  is possible.

It is more interesting to derive lower rather than upper bounds on  $H(S)$ . In order to be able to prove lower bounds on the achievable size of a key  $S$  shared by Alice and Bob in secrecy, we need to make more specific assumptions about the distribution  $P_{XYZ}$ . One natural model is that of a discrete memoryless source generating triples  $(X_i Y_i Z_i)$  independently for  $i = 1, 2, \dots, N$  according to some distribution  $P_{XYZ}$ . In other words, Alice, Bob and Eve receive  $X^N = [X_1, \dots, X_N]$ ,  $Y^N = [Y_1, \dots, Y_N]$  and  $Z^N = [Z_1, \dots, Z_N]$ , respectively, where

$$P_{X^N Y^N Z^N} = \prod_{i=1}^N P_{X_i Y_i Z_i}$$

and where  $P_{X_i Y_i Z_i} = P_{XYZ}$  for  $1 \leq i \leq N$ .

It is common practice in information theory to state results about a scenario of independent repetitions of a random experiment in terms of information *rates*. The definition of secrecy capacity of a broadcast channel introduced by Wyner [21]<sup>2</sup>, and later generalized by Csiszár and Körner [7], is natural in this sense. The secrecy capacity of a broadcast channel specified by the conditional distribution  $P_{YZ|X}$  is defined as the maximal rate at which Alice (controlling the  $X$ -input of the channel) can send information to Bob (receiving the  $Y$ -output) such that the rate at which Eve (receiving the  $Z$ -output) obtains this secret information is arbitrarily small.

In cryptography it is usually assumed that the availability of secure channels such as a trusted courier is restricted but that insecure channels are freely available. Therefore the following generalized definition of secrecy capacity introduced in [13], which allows arbitrary communication between Alice and Bob over an insecure channel, appears to be natural.

---

<sup>2</sup>We refer to [10] for a simplified treatment of the wire-tap channel.

**Definition 1.** The *secret key rate of  $X$  and  $Y$  with respect to  $Z$* , denoted  $S(X; Y||Z)$ , is the maximum rate at which Alice and Bob can agree on a secret key  $S$  while keeping the rate at which Eve obtains information arbitrarily small, i.e., it is the maximal  $R$  such that for every  $\epsilon > 0$  there exists a protocol for sufficiently large  $N$  satisfying (1)-(5) with  $X$  and  $Y$  replaced by  $X^N$  and  $Y^N$ , respectively, further satisfying

$$\frac{1}{N} I(S; C^t Z^N) \leq \epsilon, \quad (8)$$

and achieving

$$\frac{1}{N} H(S) \geq R - \epsilon.$$

*Remark:* If for some protocol the secret key generated by Alice and Bob were not uniformly distributed, an almost uniformly distributed key could be generated by applying the protocol a sufficient number of times and using an ideal data compression scheme. Hence the condition

$$\frac{1}{N} H(S) > \frac{1}{N} \log_2 |\mathcal{S}| - \epsilon \quad (9)$$

could be included in the above definition without loss of generality.

Like Wyner's and Csiszár and Körner's definition, this definition is not completely satisfactory both from a theoretical and a practical viewpoint. Since the results are asymptotic, it is possible for Eve to obtain a nonnegligible amount of information about the secret key  $S$ , even if the rate at which she receives information is arbitrarily small. In fact, according to the definition, her information is allowed to grow without bound as  $N$  goes to infinity, as long as the growth is less than linear in  $N$ . The confidentiality of a small part of a plaintext message could be of paramount importance and it is not guaranteed that this particular part is protected in a one-time pad that uses a generated secret key.

The purpose of this paper is to show that privacy amplification [3], [2] allows Alice and Bob to generate a secret key  $S$ , even when it is required that Eve's *total* information about  $S$  be negligibly small. Furthermore we require a uniformity condition on  $S$  that is much stricter than (9). We therefore introduce the following definition, where  $|\mathcal{S}|$  denotes the cardinality of the set  $\mathcal{S}$  of keys.

**Definition 2.** The *strong secret key rate of  $X$  and  $Y$  with respect to  $Z$* , denoted  $\overline{S}(X; Y||Z)$ , is defined in the same way as the secret key rate in Definition 1, with the two modifications that condition (8) is replaced by

$$I(S; C^t Z^N) \leq \epsilon$$

and that

$$H(S) \geq \log_2 |\mathcal{S}| - \epsilon.$$

We obviously have

$$\overline{S}(X; Y||Z) \leq S(X; Y||Z) \leq \min[I(X; Y), I(X; Y|Z)],$$

where the second inequality is an immediate consequence of (7). One of the results of [13] states that if either Eve has less information about  $Y$  than Alice or, by symmetry, Eve has

less information about  $X$  than Bob, then such a difference of information can be exploited:

$$\begin{aligned} S(X;Y||Z) &\geq \max[I(Y;X) - I(Z;X), I(X;Y) - I(Z;Y)] \\ &= I(X;Y) - \min[I(Z;X), I(Z;Y)]. \end{aligned} \quad (10)$$

The main result of this paper is a proof that the same lower bound holds also for the strong secret key rate.

### III Reconciliation and Privacy Amplification

One particular protocol that allows Alice and Bob to generate a secret key consists of the following two phases. It should be pointed out that this type of protocol only allows to prove our main result, namely that the lower bound (10) also holds for the strong secret key rate, but that in situations where the right side of (10) vanishes, more complicated protocols must be used to generate a secret key.

In a first phase, Alice sends  $h(X^N)$  to Bob, where  $h : \mathcal{X}^N \rightarrow \{0,1\}^L$  is a function designed to provide Bob (who knows  $Y^N$ ) with a sufficient amount of redundant information about  $X^N$  to allow him to reconstruct  $X^N$  with high probability. The existence of such a function for  $L$  on the order of  $N \cdot H(X|Y)$  is stated in the following theorem which implies that Bob can be informed by Alice about her string by sending bits (over a perfect channel) at a rate arbitrarily close to  $H(X|Y)$ . The proof of the theorem is omitted but will be given in a subsequent paper [14] which will provide a more general treatment of strong secret key rate and secrecy capacity.

**Theorem 1:** *Let the sequence  $[(X_1, Y_1), \dots, (X_N, Y_N)]$  be generated as described above. For every  $\epsilon > 0$  and  $\epsilon' > 0$ , for sufficiently large  $N$  and for every  $L$  satisfying  $L/N > (1 + \epsilon)H(X|Y)$ , there exists a function  $h : \mathcal{X}^N \rightarrow \{0,1\}^L$  such that  $[X_1, \dots, X_N]$  can be decoded from  $[Y_1, \dots, Y_N]$  and  $h(X^N)$  with error probability at most  $\epsilon'$ .*

In a second phase, called privacy amplification, Alice and Bob compress the now shared string  $X^N$ , in a manner known to Eve, to result in a shorter binary string  $S = [S_1, \dots, S_M]$  with virtually uniform distribution about which Eve has essentially no information. Of course, this privacy amplification step must take into account Eve's total information about  $X^N$  consisting of  $Z^N$  and  $h(X^N)$ .

Privacy amplification was introduced in [3] and generalized in [2] and can be described as follows. Assume Alice and Bob share an  $N$ -bit string  $w$  about which an eavesdropper Eve has incomplete information characterized by a probability distribution  $P$  over the  $N$ -bit strings. For instance, Eve might have received some bits or parities of bits of  $w$ , she might have eavesdropped on some of the bits of  $w$  through a binary symmetric channel, or have some more complicated type of information about  $w$ . Alice and Bob have some knowledge of this distribution  $P$ , but they do not know exactly what is compromised about the secrecy of their string. Using a public channel, which is totally susceptible to eavesdropping, they wish to agree on a function  $g : \{0,1\}^N \rightarrow \{0,1\}^M$  such that Eve, despite her partial knowledge about  $w$  and complete knowledge of  $g$ , almost certainly knows nearly nothing about  $g(w)$ . This process transforms a partially secret  $N$ -bit string  $w$  into a highly secret but shorter  $M$ -bit string  $g(w)$ .

Bennett, Brassard, and Robert [3] solved the problem for the case where Eve is allowed to specify (secretly) an arbitrary eavesdropping function  $e : \{0, 1\}^N \rightarrow \{0, 1\}^T$  from  $N$  bits to  $T$  bits such that only  $T$ , but not the function  $e$  is known to Alice and Bob, and where Eve obtains the result  $e(w)$  of applying the eavesdropping function to  $w$ . Equivalently, Eve could be allowed to perform an arbitrary computation with  $w$  as input, as long as she keeps only  $T$  bits of the result and discards the input and all the intermediate results. The solution of [3] consists of Alice randomly selecting a function from a universal class of hash functions (see definition below) mapping  $N$ -bit strings to  $T$ -bit strings for an appropriate choice of  $T$ , and sending the description (or index) of the selected function to Bob (and hence also to Eve) over the insecure channel.

**Definition 3** [6]: A class  $G$  of functions  $\mathcal{A} \rightarrow \mathcal{B}$  is *universal*<sup>2</sup> (“universal” for short) if, for any distinct  $x_1$  and  $x_2$  in  $\mathcal{A}$ , the probability that  $g(x_1) = g(x_2)$  is at most  $1/|\mathcal{B}|$  when  $g$  is chosen at random from  $G$  according to the uniform distribution.

*Example:* Let  $a$  be an element of  $GF(2^N)$  and also interpret  $x$  as an element of  $GF(2^N)$ . Consider the function  $\{0, 1\}^N \rightarrow \{0, 1\}^M$  assigning to an argument  $x$  the first  $M$  bits of the element  $ax$  of  $GF(2^N)$ . The class of such functions for  $a \in GF(2^N)$  with  $a \neq 0$  is a universal class of functions for  $1 \leq M \leq N$ .

The results of [3] were generalized by Bennett, Brassard, Crépeau, and Maurer [2] to include scenarios in which Eve’s information about  $w$  is specified by some general probability distribution satisfying a certain constraint in terms of collision entropy defined below.

**Definition 4** [2]: Let  $P_W$  be a probability distribution over some sample space  $\mathcal{W}$ . (Equivalently, we can consider the random variable  $W$  distributed according to  $P_W$ .) The *collision probability* of  $W$ , denoted  $P_c(W)$ , is the probability of drawing the same element if one samples twice in  $\mathcal{W}$ , with replacement, according to probability distribution  $P_W$ :

$$P_c(W) = \sum_{w \in \mathcal{W}} (P_W(w))^2.$$

The *collision entropy*<sup>3</sup> of  $W$ , denoted  $H_c(W)$  is the negative logarithm of the collision probability, i.e.,

$$H_c(W) = -\log P_c(W). \quad ^4$$

It follows immediately from Jensen’s inequality that

$$H(W) \geq H_c(W), \quad (11)$$

with equality if and only if  $P_W$  is the uniform distribution over  $\mathcal{W}$  or a subset of  $\mathcal{W}$ , and where  $H(W)$  is the (Shannon) entropy of a random variable  $W$  distributed according to  $P_W$ .

In analogy to Shannon entropy, one can also define conditional collision entropy. For an event  $\mathcal{E}$ ,  $H_c(W|\mathcal{E})$  is naturally defined as the collision entropy of the conditional distribution  $P_{W|\mathcal{E}}$ , for instance

$$H_c(W|V = v) = -\log \sum_{w \in \mathcal{W}} (P_{W|V}(w, v))^2,$$

---

<sup>3</sup>also known as Renyi entropy of order 2

<sup>4</sup>All logarithms in this paper are to the base 2.

and the collision entropy conditioned on a random variable can be defined as the expected value of the conditional collision entropy:

$$H_c(W|V) = \sum_v P_V(v) H_c(W|V = v).$$

One can also define collision information in analogy to Shannon information.

Like Shannon entropy, collision entropy conditioned on a random variable is “well-behaved”: it is proved in [5] that

$$H_c(W) - H_c(W|V) \leq H(V).$$

It should be pointed out, however, that the more intuitive inequality  $H_c(W) - H_c(W|V) \leq H_c(V)$  is false in general [5]. However, an important problem we will have to deal with is that, like for Shannon entropy, the condition  $V = v$  can induce an arbitrarily large decrease of collision entropy: If  $V$  can take on  $2^L$  values,  $H_c(W) - H_c(W|V = v) \gg L$  is possible for certain values  $v$ .

We will make use in a crucial manner of an interesting and counterintuitive property of collision entropy pointed out and used in [2]. As opposed to Shannon entropy, collision entropy can *increase* when extra information is revealed, i.e.,  $H_c(W|V) > H_c(W)$  is possible. (Of course, this property rules out collision entropy as a measure of information that could be useful in investigating source and channel coding.)

We now return to the discussion of privacy amplification. One of the main results of [2] can be restated as follows.

**Theorem 2** [2]: *Let  $P_W$  be a probability distribution over  $\mathcal{W}$  with collision entropy  $H_c(W)$ , and let  $G$  be the random variable corresponding to a universal class of hash functions from  $\mathcal{W}$  to  $\{0, 1\}^M$  with uniform distribution over the class. Then*

$$H(G(W)|G) \geq H_c(G(W)|G) \geq M - \frac{2^{M-H_c(W)}}{\ln 2}.$$

*Remark:* While this theorem applies of course also to conditional probability distributions, i.e.,

$$H_c(G(W)|G, V = v) \geq M - 2^{M-H_c(W|V=v)} / \ln 2,$$

it should be pointed out that it cannot be generalized to collision entropy conditioned on a random variable:  $H_c(G(W)|GV) \geq M - 2^{M-H_c(W|V)} / \ln 2$  is false in general.

Theorem 2 states that if Alice and Bob share a particular string  $w$  and Eve’s information about  $w$  can be modeled by the distribution  $P_{W|V=v}$  (where  $v$  denotes the particular value of her information vector) about which Alice and Bob know nothing except a lower bound  $T$  on the collision entropy, i.e.  $H_c(W|V = v) \geq T$ , then Alice and Bob can generate a secret key of roughly  $T$  bits. More precisely, if Alice and Bob compress  $w$  slightly more to an  $M$ -bit key with  $M < T$ , then Eve’s total information about this key decreases exponentially in the excess compression  $T - M$ .

## IV A Lower Bound on Strong Secret Key Rate

Our goal is to apply privacy amplification to the string  $X^N$  shared by Alice and Bob after the error-correction phase in which  $h(X^N)$  is sent from Alice to Bob, taking into account Eve's knowledge consisting of  $Z^N$  and  $h(X^N)$ . However, several major problems arise:

- First, Eve's initial collision entropy  $H_c(X^N|Z^N = z^N)$  depends on the particular string  $z^N$  that she has received. Unfortunately, as pointed out above, privacy amplification does not apply when only a bound on the *average* collision entropy  $H_c(X^N|Z^N)$  is known.
- Second, the reduction of Eve's collision entropy about  $X_1, \dots, X_N$  due to receiving a particular value of the error-correction information,  $h(X^N) = a$ , sent from Alice to Bob over the public channel, must be analyzed. Knowing that  $H_c(X^N|Z^N) - H_c(X^N|Z^N, h(X^N)) \leq H(h(X^N)) \leq L$  is not sufficient for the same reason as mentioned above. Because  $H_c(X^N|Z^N = z^N, h(X^N) = a)$  could potentially be much smaller than  $H_c(X^N|Z^N = z^N)$  one has to consider all possible values of  $h(X^N)$ .
- Third, Theorem 2 suggests that  $H_c(X^N|Z^N = z^N, h(X^N) = a)$  is an upper bound on the size of the secret key that can be generated by privacy amplification. Unfortunately, the collision entropy is generally smaller than the Shannon entropy. In particular,  $H_c(X^N|Z^N)$  will generally be substantially smaller than  $H(X^N|Z^N)$ ; hence it appears impossible to exploit Eve's full Shannon entropy  $H(X^N|Z^N)$ , reduced by the amount of extra information (on the order of  $H(X|Y)$ ) provided by  $h(X^N)$ , as would be necessary in order to prove that the lower bound (10) also holds for the strong secret key rate.
- Fourth, one needs to guarantee that the finally shared string  $S = [S_1, \dots, S_M]$  has virtually maximal entropy.

We solve these problems by exploiting the fact described earlier that collision entropy can increase when extra information is revealed. It is therefore conceivable to consider an oracle who gives Eve some side information (called spoiling knowledge in [2]) about  $X^N$  for free. Revealing extra information can certainly not harm Eve since she could always discard it. However, if chosen carefully, this extra information may increase Eve's collision entropy. This demonstrates that a longer key than suggested by considering Eve's collision entropy about  $X^N$  (without the oracle's "help") can safely be distilled by application of Theorem 2. Clearly, Eve's Shannon entropy will be reduced by receiving the oracle's side information, but in our case this reduction will be negligible in terms of rate, i.e., when divided by  $N$ .

In the following we will make use of typical sequence arguments. There exist several definitions of typical sequences, and we use that of [4] for strongly typical sequences. Consider a probability distribution  $P_U$  over some finite set  $\mathcal{U}$ , which we assume without loss of generality to be  $\mathcal{U} = \{1, \dots, t\}$  for some  $t$ . We further assume that  $P_U(i) > 0$  for  $1 \leq i \leq t$ . Consider a sequence  $u^N$  of  $N$  digits of  $\mathcal{U}$  and define  $n_i(u^N)$ , for  $i = 1, \dots, t$ , to be the number of occurrences of the digit  $i$  in  $u^N$ . A sequence  $u^N$  is called a  $\delta$ -typical sequence if and only if

$$(1 - \delta)P_U(i) \leq \frac{n_i(u^N)}{N} \leq (1 + \delta)P_U(i)$$

for  $1 \leq i \leq t$ . Consider now a sequence  $U^N = [U_1, \dots, U_N]$  of  $N$  independent and identically distributed random variables, each distributed according to  $P_U$ . Using the Chernoff bound (cf. [4]) one can prove that the total probability of all  $\delta$ -typical sequences approaches 1 as  $N$  goes to infinity. More precisely, the total probability of the non- $\delta$ -typical sequences goes to zero faster than  $1/N$ : For every  $\delta > 0$  and  $\epsilon > 0$ , we have

$$N \cdot P[U^N \text{ is not } \delta\text{-typical}] < \epsilon \quad (12)$$

for sufficiently large  $N$ .

We now return to the discussion of our secret key agreement scenario with independent random triples  $(X_i Y_i Z_i)$ , for  $i = 1, \dots, N$ , being generated according to  $P_{XYZ}$ . We first focus on the sequence of pairs  $(X_i Z_i)$ . Without loss of generality we let the alphabets for  $X$  and  $Z$  be  $\mathcal{X} = \{1, \dots, t_1\}$  and  $\mathcal{Z} = \{1, \dots, t_2\}$ , respectively. Let  $m_j$  for  $1 \leq j \leq t_2$  denote the number of occurrences of digit  $j$  in the sequence  $Z_1, \dots, Z_N$ , and let  $n_{ij}$  for  $1 \leq i \leq t_1$  and  $1 \leq j \leq t_2$  denote the number of occurrences of the pair  $(i, j)$  in the sequence  $[(X_1, Z_1), \dots, (X_N, Z_N)]$ . We have

$$\sum_{i=1}^{t_1} n_{ij} = m_j \quad (13)$$

for  $1 \leq i \leq t_1$ , and

$$\sum_{j=1}^{t_2} m_j = N. \quad (14)$$

Let  $\mathcal{E}$  be the event that the sequence  $[(X_1, Z_1), \dots, (X_N, Z_N)]$  is  $\delta$ -typical for the alphabet  $\mathcal{X} \times \mathcal{Z}$  and the distribution  $P_{XZ}$ . According to (12),  $P[\bar{\mathcal{E}}]$  can be made arbitrarily small for any fixed  $\delta > 0$  by choosing a sufficiently large blocklength  $N$ . In the following we will consider probability distributions and entropies conditioned on the event  $\mathcal{E}$ . By definition, this condition implies that

$$(1 - \delta)P_{XZ}(i, j) \leq \frac{n_{ij}}{N} \leq (1 + \delta)P_{XZ}(i, j) \quad (15)$$

and, as a consequence, that

$$(1 - \delta)P_Z(j) \leq \frac{m_j}{N} \leq (1 + \delta)P_Z(j). \quad (16)$$

Eve knows a particular sequence  $z^N$  with corresponding values  $m_1, \dots, m_{t_2}$ . Assume now that the oracle mentioned above tells Eve the numbers  $n_{ij}$  for free. This extra information, denoted as  $O$ , decreases Eve's Shannon entropy somewhat, i.e.,

$$H(X^N | Z^N = z^N, O) < H(X^N | Z^N = z^N)$$

and

$$H(X^N | Z^N = z^N, O, \mathcal{E}) < H(X^N | Z^N = z^N, \mathcal{E})$$

but increases her collision entropy significantly, i.e.,

$$H_c(X^N | Z^N = z^N, O, \mathcal{E}) > H_c(X^N | Z^N = z^N, \mathcal{E}).$$

In fact, for a particular value  $O = o$  provided by the oracle, Eve's distribution of  $X^N$ , i.e.  $P_{X^N|Z^N=z^N, O=o, \mathcal{E}}$ , is such that all sequences  $[x_1, \dots, x_N]$  that are consistent with her information are equally probable. It is easy to see that the number of such sequences is

$$Q = \prod_{j=1}^{t_2} \frac{m_j!}{\prod_{i=1}^{t_1} n_{ij}!}.$$

Therefore both the Shannon and the collision entropy of this distribution are equal to  $\log Q$ .

As pointed out before, privacy amplification applies to conditional distributions only when a bound on the collision entropy of a random variable, given the particular value of the conditioning random variable, is known. In order to be able to apply privacy amplification to the distribution  $P_{X^N|Z^N=z^N, O=o, \mathcal{E}}$ , we state the following result for specific values  $Z^N = z^N$  and  $O = o$ . Of course, it also holds when averaged over all values of  $Z^N$  and  $O$ .

**Lemma 3.** *For  $0 < \delta \leq 1/2$  and for all values  $z^N$  and  $o$ ,*

$$H_c(X^N|Z^N = z^N, O = o, \mathcal{E}) > N [H(X|Z) - \delta(H(X) + H(XZ) + 4)] - t_1 t_2 \log N.$$

This lemma implies that for sufficiently small  $\delta$  and for sufficiently large  $N$ , Eve's per-digit Shannon and collision entropy are both arbitrarily close to  $H(X|Z)$ . Note again that  $H_c(X^N|Z^N O, \mathcal{E})$  is significantly larger than  $H_c(X^N|Z^N, \mathcal{E})$ .

*Proof:* Stirling's formula for  $n!$  (cf. [9], p. 467) implies that

$$n(\log n - \alpha) < \log n! < n(\log n - \alpha) + \log n$$

for all sufficiently large  $n$ , where  $\alpha = 1/\ln 2$  and  $\ln 2$  denotes the logarithm of 2 to the base  $e$ . Using (13), (14), (15) and (16) we get

$$\begin{aligned} \log Q &= \sum_{j=1}^{t_2} \left( \log(m_j!) - \sum_{i=1}^{t_1} \log(n_{ij}!) \right) \\ &> \sum_{j=1}^{t_2} m_j (\log m_j - \alpha) - \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} [n_{ij} (\log n_{ij} - \alpha) + \log n_{ij}] \\ &> \sum_{j=1}^{t_2} m_j (\log m_j - \log N) - \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} [n_{ij} (\log n_{ij} - \log N) + \log N] \\ &= N \left[ \sum_{j=1}^{t_2} \frac{m_j}{N} \log \frac{m_j}{N} - \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N} \right] - t_1 t_2 \log N \\ &\geq N \left[ \sum_{j=1}^{t_2} (1 + \delta) P_Z(j) \log((1 - \delta) P_Z(j)) \right. \\ &\quad \left. - \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} (1 - \delta) P_{XZ}(i, j) \log((1 + \delta) P_{XZ}(i, j)) \right] - t_1 t_2 \log N \end{aligned}$$

$$\begin{aligned} &\geq N \left[ (1 + \delta) \sum_{j=1}^{t_2} P_Z(j) \log P_Z(j) - (1 - \delta) \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} P_{XZ}(i, j) \log P_{XZ}(i, j) \right] \\ &\quad + N \left[ (1 + \delta) \log(1 - \delta) - (1 - \delta) \log(1 + \delta) \right] - t_1 t_2 \log N \end{aligned}$$

Note that because  $\sum_{j=1}^{t_2} m_j = \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} n_{ij}$ , the two occurrences of  $\alpha$  on the second line can both be deleted or replaced by any other expression (like  $\log N$  in our case). We have also made use of the trivial fact that  $\log n_{ij} \leq \log N$ . It is easy to check that  $(1 + \delta) \log(1 - \delta) - (1 - \delta) \log(1 + \delta) > -4\delta$  for all  $\delta \leq 1/2$ . Hence

$$\begin{aligned} H_c(X^N | Z^N = z^N, O = o, \mathcal{E}) &= \log Q \\ &> N[-(1 + \delta)H(Z) + (1 - \delta)H(XZ) - 4\delta] - t_1 t_2 \log N \\ &= N[H(X|Z) - \delta(H(X) + H(XZ) + 4)] - t_1 t_2 \log N \end{aligned}$$

for all  $\delta \leq 1/2$ , as was to be shown.  $\square$

In order to apply privacy amplification according to Theorem 2 to compress the string  $X^N$  now shared by Alice and Bob to a shorter string  $S$  about which Eve has essentially no information, it remains to investigate the reduction of Eve's collision entropy about  $X^N$  due to seeing  $h(X^N)$  sent from Alice to Bob over the public channel. As pointed out before, for any particular value  $a$  taken on by  $h(X^N)$ , the reduction of collision entropy induced by obtaining sideinformation  $h(X^N) = a$ , i.e.  $H_c(X^N) - H_c(X^N|h(X^N) = a)$ , could generally be arbitrarily large. (The fact that  $H_c(X^N) - H_c(X^N|h(X^N)) \leq H(h(X^N))$  is of little use here.) However, it is easy to prove (cf. [2],[5]) that for a uniform distribution, i.e., one for which all nonzero probabilities are identical, revealing  $L$  bits of information can reduce the collision entropy by at most  $L$ . Thus

$$H_c(X^N | Z^N = z^N, O = o, h(X^N) = a, \mathcal{E}) \geq H_c(X^N | Z^N = z^N, O = o, \mathcal{E}) - L. \quad (17)$$

The main result of this paper can be summarized in the following theorem. Only a sketch of the proof is given and we refer to [14] for a complete proof.

**Theorem 4:**  $\bar{S}(X; Y || Z) \geq \max[I(Y; X) - I(Z; X), I(X; Y) - I(Z; Y)]$ .

*Proof sketch:* We only prove that  $I(Y; X) - I(Z; X)$  is an achievable rate; the proof for  $I(X; Y) - I(Z; Y)$  follows by symmetry. Alice and Bob choose a suitable error-correction function  $h : \mathcal{X}^N \rightarrow \{0, 1\}^L$  and, after having sent and received  $h(X^N)$ , choose a compression function  $G$  at random from a universal class of hash functions  $\mathcal{X}^N \rightarrow \{0, 1\}^M$ , for appropriate parameters  $L$  and  $M$ . Then they compute  $S = G(X^N)$ . The quantity to be bounded is  $I(S; GZ^N h(X^N))$ . It can be shown to be arbitrarily small by proving that  $H(S|GZ^N h(X^N))$  is arbitrarily close to  $M$ .

$$\begin{aligned} H(S|GZ^N h(X^N)) &\geq (1 - P[\mathcal{E}])H(S|GZ^N h(X^N), \mathcal{E}) \\ &\geq H(S|GZ^N h(X^N), \mathcal{E}) - P[\mathcal{E}] \cdot H(S) \\ &\geq H(S|GZ^N h(X^N)O, \mathcal{E}) - P[\mathcal{E}] \cdot N \cdot H(X). \end{aligned}$$

Theorem 2 implies that if  $\mathcal{E}$  occurs, then for every value  $[z^N, o, a]$  which the random triple  $[Z^N, O, h(X^N)]$  can take on,

$$H(S|G, Z^N = z^N, O = o, h(X^N) = a, \mathcal{E}) \geq M - 2^{M - H_c(X^N | Z^N = z^N, O = o, h(X^N) = a, \mathcal{E}) / \ln 2}.$$

Let  $\epsilon > 0$  be an arbitrary but fixed parameter. For an appropriate choice of  $L/N$   $\epsilon$ -close to  $H(X|Y)$  and of  $M/N$   $\epsilon$ -close to  $H(X|Z) - H(X|Y)$ , and by using (17) and Lemma 3, one can show that the above exponent goes to minus infinity as  $N$  goes to infinity. Hence  $H(S|G, Z^N = z^N, O = o, h(X^N) = a, \mathcal{E})$  and thus also  $H(S|GZ^N h(X^N)O, \mathcal{E})$  can be made arbitrarily close to  $M$  for sufficiently large  $N$ . Furthermore, (12) implies that  $N \cdot P[\mathcal{E}] \cdot H(X)$  vanishes when  $N$  goes to infinity, and Theorem 1 implies that Bob can decode  $X^N$  from  $Y^N$  and  $h(X^N)$  with probability arbitrarily close to 1.  $\square$

## V Conclusions

We have pointed out that previous definitions of secrecy capacity of broadcast channels and secret key rate of random triples are not satisfactory because the total amount of information an opponent can obtain is not bounded, let alone arbitrarily small. For a correspondingly stronger definition of secret key rate it was proved that the results previously obtained for a weak definition of secret key rate also hold for the new stronger definition. The techniques of [2] used in the proof appear to be novel and we believe that they will have other applications in information theory. Results for a strengthened definition (in analogy to Definition 2) of secrecy capacity in the broadcast channel models of Wyner [21] and Csiszár and Körner [7] will be described in [14].

The strong secret key rate is an asymptotic definition. However, concrete protocols based on techniques described in Section V of [13] and in [8] and on efficiently decodable error-correcting codes can be constructed and analyzed using the techniques of [5]. Perfectly-secure secret-key agreement is possible even when Eve initially has more information about Alice's string than Bob and also more information about Bob's string than Alice [13]. In this case, however, several rounds of interaction between Alice and Bob are required.

## VI Acknowledgment

It is a pleasure to thank Jim Massey for providing the initial motivation for this research, and Charles H. Bennett, Gilles Brassard, Christian Cachin, Claude Crépeau, and Martin Gander for interesting discussions.

## References

- [1] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography – Part I: secret sharing", *IEEE Transactions on Information Theory*, Vol. 39, No. 4, pp. 1121-1132, 1993.
- [2] C.H. Bennett, G. Brassard, C. Crépeau, and U.M. Maurer, "Privacy amplification against probabilistic information, preprint", 1993.
- [3] C.H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion", *SIAM Journal on Computing*, Vol. 17, No. 2, pp. 210-229, 1988.

- [4] R.E. Blahut, *Principles and Practice of Information Theory*, Reading, MA: Addison-Wesley, 1987.
- [5] C. Cachin and U.M. Maurer, “Linking information reconciliation and privacy amplification”, preprint, 1994.
- [6] J.L. Carter and M. N. Wegman, “Universal classes of hash functions”, *Journal of Computer and System Sciences*, Vol. 18, 1979, pp. 143–154.
- [7] I. Csiszár and J. Körner, “Broadcast channels with confidential messages”, *IEEE Transactions on Information Theory*, Vol. 24, No. 3, pp. 339-348, 1978.
- [8] M. Gander and U.M. Maurer, “On the secret-key rate of binary random variables”, (extended abstract), to be presented at the 1994 Int. Symp. on Information Theory.
- [9] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete mathematics*, Reading, MA: Addison-Wesley, 1990.
- [10] J.L. Massey, “A simplified treatment of Wyner’s wire-tap channel”, *Proc. 21st Annual Allerton Conf. on Comm., Control, and Computing*, Monticello, IL, Oct. 5-7, 1983, pp. 268-276.
- [11] J.L. Massey, “Contemporary cryptology – an Introduction”, in *Contemporary cryptology – the science of information integrity*, G.J. Simmons (Ed.), IEEE Press, 1992.
- [12] U.M. Maurer, *Provable security in cryptography*, Ph. D. dissertation, No. 9260, Swiss Federal Institute of Technology (ETH), Zurich, 1990.
- [13] U.M. Maurer, “Secret key agreement by public discussion from common information”, *IEEE Transactions on Information Theory*, Vol. 39, No. 3, pp. 733-742, May 1993.
- [14] U.M. Maurer, “Strengthening the definition of secret key rate and secrecy capacity”, in preparation.
- [15] U.M. Maurer and J.L. Massey, “Local randomness in pseudo-random sequences”, *Journal of Cryptology*, Vol. 4, No. 2, 1991, pp. 135-149.
- [16] U.M. Maurer and J.L. Massey, “Cascade ciphers: the importance of being first”, *Journal of Cryptology*, Vol. 6, No. 1, pp. 55-61, 1993.
- [17] R.L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, 1978.
- [18] C.E. Shannon, “Communication theory of secrecy systems”, *Bell System Technical Journal*, Vol. 28, pp. 656-715, Oct. 1949.
- [19] G.S. Vernam, “Cipher printing telegraph systems for secret wire and radio telegraphic communications”, *J. Amer. Inst. Elec. Eng.*, Vol. 55, pp. 109-115, 1926.

- [20] M.N. Wegman and J.L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, Vol. 22, pp. 265-279, 1981.
- [21] A.D. Wyner, "The wire-tap channel", *Bell System Technical Journal*, Vol. 54, No. 8, pp. 1355-1387, 1975.

# The Self-Shrinking Generator

Willi Meier  
HTL Brugg-Windisch  
CH-5200 Windisch, Switzerland

Othmar Staffelbach  
Gretag Data Systems AG  
CH-8105 Regensdorf, Switzerland

Dedicated to James L. Massey on the occasion of his 60th birthday.

## Abstract

A construction of a pseudo random generator based on a single linear feedback shift register is investigated. The construction is related to the so-called shrinking generator and is attractive by its conceptual simplicity. The lower bounds that are provided for period, linear complexity, and known cryptanalytic attacks allow for efficient practical implementations at a reasonable scale.

## I Introduction

In [1] a new pseudo random sequence generator, the so-called *shrinking generator*, has been suggested by Coppersmith, Krawcyk, and Mansour for potential use in stream cipher applications. The shrinking generator is attractive by its conceptual simplicity as it combines only two linear feedback shift registers in a simple way. One is tempted to conjecture that such a simple construction might be insecure. However no successful cryptanalytic attack has been publicly reported so far.

In this paper we present an apparently simpler structure using only one LFSR whose output sequence is shrunken in a similar way as is done for the shrinking generator. As the shrinking of the LFSR-sequence is achieved under the control of the LFSR itself, the generator will be called *self-shrinking generator*.

Recall that the shrinking generator [1] uses two binary LFSRs, say LFSR 1 and LFSR 2, as basic components. The pseudo random bits are produced by shrinking the output sequence of LFSR 1 under the control of LFSR 2 as follows: The output bit of LFSR 1 is taken if the current output of LFSR 2 is 1, otherwise it is discarded. For the self-shrinking generator we suggest to use only one LFSR. Instead of output bits, pairs of output bits are considered. If a pair happens to take the value 10 or 11, this pair is taken to produce the pseudo random bit 0 or 1, depending on the second bit of the pair. On the other hand if a pair happens to be 01 or 00, it will be discarded. The key consists of the initial state of the LFSR and preferably also of the LFSR feedback logic. For practical applications it is assumed that the feedback connection is to produce maximal length LFSR-sequences.

We mention here that Rueppel et al. in unpublished work [5] have proposed a generator which is also based on the principle of discarding certain output bits of an LFSR. It turns out however

that the construction in [5] is different from ours, and that the analysis of [5] does not apply to the self-shrinking generator. Furthermore the self-shrinking mechanism of an LFSR might be compared with the self-decimation of an LFSR as introduced in [4]. As quoted in [4], the self-decimated sequence cannot be directly applied for stream enciphering. As the decimation intervals are revealed by the output sequence, one can derive the original LFSR-sequence at fixed positions from the self-decimated sequence. Thus the original LFSR-sequence can be computed by solving linear equations. For a shrunken or self-shrunken sequence one also sees certain output bits of the original LFSR-sequence, but one does not know the size of the gaps between the known bits.

It turns out that the self-shrinking generator and the shrinking generator are closely related to each other. In fact, it will be shown in Section II that the self-shrinking generator can be implemented as a shrinking generator, and conversely, that the shrinking generator can be implemented as a self-shrinking generator. The latter implementation however cannot be accomplished with a maximum length LFSR. Thus the self-shrinking generator has its main interest in implementing the shrinking principle at lower hardware costs. According to [1], the effective key size of the shrinking generator, measured in terms of the complexity of known cryptanalytic attacks, is roughly half of the maximum possible key size. In view of the presently known cryptanalytic attacks (see Section V) the effective key size of the self-shrinking generator can be estimated to be more than 80% of the maximum possible value.

It is difficult to give a general and reliable measure for the cryptographic quality of pseudo random sequences being applied in stream cipher systems. Certainly well known are the classical measures, period and linear complexity. For a secure design one should have proofs or at least strong practical evidence that these parameters are large enough to withstand the more generic attacks like the celebrated *Berlekamp-Massey LFSR synthesis algorithm* [3]. For a self-shrinking generator implemented with a maximum length LFSR of length  $N$ , it is proved in Section III that the period and the linear complexity are lower bounded by  $2^{\lfloor N/2 \rfloor}$  and  $2^{\lfloor N/2 \rfloor - 1}$ , respectively. Furthermore in Section IV strong evidence is provided that the period is in fact  $2^{N-1}$  for  $N > 3$ , and that the linear complexity is very close to that value. Therefore it is easy to implement the self-shrinking generator to satisfy sufficiently large proved lower bounds for period and linear complexity.

The experimental results in Section IV reveal another interesting fact, namely that the linear complexity does not exceed the value  $2^{N-1} - N + 2$ . This can be considered as an algebraic property of the shrunken LFSR-sequence. The original LFSR-sequence has a rich algebraic structure. For being applicable as pseudo randomizer for cryptographic purposes it is necessary to destroy most of the algebraic structure—in particular the property of satisfying a short linear recursion.

For the self-shrinking generator, the fact that it is unknown at which positions the LFSR-sequence is shrunken and that the shrinking is controlled by the LFSR itself suggest that most of the algebraic structure of the original LFSR-sequence has been destroyed. Thus the above mentioned upper bound on the linear complexity appears to be surprising. Proving this fact as well as the conjecture that  $2^{N-1}$  is the minimum period remain as open problems. These problems deal with elementary manipulations on LFSR-sequences, for which a thorough mathematical theory is available.

## II Shrinking and Self-Shrinking

Self-shrinking can be applied to arbitrary binary sequences. The original sequence  $\mathbf{a} = (a_0, a_1, a_2, \dots)$  is considered as a sequence of pairs of bits  $((a_0, a_1), (a_2, a_3), \dots)$ . If a pair  $(a_{2i}, a_{2i+1})$  equals the value  $(1, 0)$  or  $(1, 1)$ , it is taken to produce the pseudo random bit 0 or 1, respectively. On the other hand, if the pair is equal to  $(0, 0)$  or  $(0, 1)$ , it will be discarded, which means that it will not contribute an output bit to the new sequence  $\mathbf{s} = (s_0, s_1, s_2, \dots)$ .

Self-shrinking is in particular intended to be applied to pseudo random sequences in order to produce new pseudo random sequences of (potentially) better cryptographic quality. We especially analyze the situation where the original sequence  $\mathbf{a}$  is generated by an LFSR. For a cryptographic application the key consists of the initial state of the LFSR. Preferably the feedback connection is variable and also part of the key. The self-shrunken sequence  $\mathbf{s}$  can be considered as being obtained from the original sequence  $\mathbf{a}$  by discarding certain bits. In the average  $3/4$  of the bits are expected to be omitted. Hence the data rate of the original sequence is reduced by the factor 4.

It appears to be natural to ask the question whether the self-shrinking generator can be implemented as a special case of the shrinking generator. To show that this is in fact the case, let  $\mathbf{a} = (a_0, a_1, a_2, \dots)$  be the sequence produced by an LFSR of length  $N$  defining a self-shrinking generator. According to the self-shrinking rule, the sequence  $(a_0, a_2, a_4, \dots)$  effects the output control, and  $(a_1, a_3, a_5, \dots)$  defines the sequence being controlled. Both sequences can be produced by the original LFSR when loaded with the initial states  $(a_0, a_2, \dots, a_{2N-2})$ , or  $(a_1, a_3, \dots, a_{2N-1})$  respectively. This implies that the self-shrinking generator can be implemented as a shrinking generator with two LFSRs having identical feedback connections.

Conversely, we will show that the shrinking generator can be implemented as a special case of the self-shrinking generator. To this end, consider an arbitrary shrinking generator defined by two linear shift registers LFSR 1 and LFSR 2 with feedback polynomials  $f(x)$ , and  $g(x)$ , respectively. Furthermore, let  $\mathbf{b} = (b_0, b_1, b_2, \dots)$  and  $\mathbf{c} = (c_0, c_1, c_2, \dots)$  denote the corresponding LFSR output sequences. Then, by applying the self-shrinking rule to the interleaved sequence  $\mathbf{a} = (c_0, b_0, c_1, b_1, \dots)$ , the original output sequence of the shrinking generator is reproduced. On the other hand, it can be shown that the sequence  $\mathbf{a}$  can be produced by an LFSR with feedback polynomial  $f(x^2)g(x^2) = f(x)^2g(x)^2$ . This implies that the shrinking generator has an equivalent implementation as a self-shrinking generator.

The investigations on the shrinking generator in [1] assume that the two LFSRs involved are independent, e.g., that their periods are coprime. Therefore the results in [1] on period and linear complexity do not apply to the self-shrinking generator. For obtaining corresponding results for the self-shrinking generator, a different approach will be required.

## III Period and Linear Complexity of Self-Shrunken Maximum Length LFSR-sequences

We now establish lower and upper bounds on period and linear complexity of self-shrunken sequences generated by maximum length LFSRs ( $m$ -LFSRs).

## Period

Let  $\mathbf{a} = (a_0, a_1, a_2, \dots)$  be the output sequence of a nontrivially initialized  $m$ -LFSR of length  $N$ . Hence  $\mathbf{a}$  is a sequence with period  $2^N - 1$ . The self-shrunken sequence will also be periodic. In fact, after  $2(2^N - 1)$  bits of the original sequence, the sequence of pairs  $(a_0, a_1), (a_2, a_3), \dots, (a_{2^N-2}, a_0), (a_1, a_2), \dots, (a_{2^N-3}, a_{2^N-2})$  has been processed, and the next pair will be  $(a_0, a_1)$  again. Hence the shrunken sequence is repeating. Within this period each possible output pair  $(a_i, a_{i+1})$ ,  $0 \leq i < 2^N - 1$ , of the original LFSR-sequence has occurred exactly once. As is well-known, within the period of a  $m$ -LFSR-sequence each of the pairs 01, 10, and 11 appears exactly  $2^{N-2}$  times, and the pair 00 appears  $2^{N-2} - 1$  times. By the definition of the shrinking rule, it follows that  $2^{N-1}$  is a period of the shrunken sequence. Moreover, as the pairs 10 and 11 occur equally often, the shrunken sequence must be balanced. As the shrunken sequence is repeating after  $2^{N-1}$  bits, it must be purely periodic with period  $p = 2^{N-1}$ , i.e.,  $s_n = s_{n+p}$  for all  $n > 0$ . This implies that the smallest period  $P$  of  $\mathbf{s}$  must divide  $2^{N-1}$ . Summarizing we obtain

**Proposition 1** *Let  $\mathbf{a}$  be an  $m$ -LFSR-sequence generated by an LFSR of length  $N$  and let  $\mathbf{s}$  be the self-shrunken sequence obtained from  $\mathbf{a}$ . Then  $\mathbf{s}$  is a balanced sequence whose period divides  $2^{N-1}$ .*

A lower bound on the period of a shrunken  $m$ -LFSR-sequence is given in the following theorem.

**Theorem 2** *The period  $P$  of a self-shrunken maximum length LFSR-sequence produced by an LFSR of length  $N$  satisfies*

$$P \geq 2^{\lfloor N/2 \rfloor}. \quad (1)$$

**Proof.** Let us first consider the case when  $N$  is even, and let  $n = N/2$ . Since the feedback connection of the LFSR is chosen to produce maximum length sequences, every nonzero  $N$ -bit word appears exactly once when scanning the LFSR-sequence with a window of length  $N$  over the full period. In view of the self-shrinking, we consider the sequence  $\mathbf{a}$  being scanned over the double period with increments by two bits. As the period is odd, the same  $N$ -bit patterns occur (possibly in different order) as if the sequence were scanned over one period with one bit increments. By the maximum length property, the  $N$ -bit pattern  $(1, x_1, 1, x_2, \dots, 1, x_n)$  appears in the original sequence for every choice of  $(x_1, x_2, \dots, x_n)$ . It follows that every  $n$ -bit pattern appears in the shrunken sequence when scanning it with window size  $n$ .

If a sequence of period  $P$  is scanned over an interval of arbitrary length, at most  $P$  different patterns can occur (independent of the window size). As the shrunken sequence contains all  $2^n$  patterns of length  $n$ , it follows that the inequality  $P \geq 2^n$  must hold. This proves the theorem for the case when  $N$  is even. For odd  $N$  let  $n = (N - 1)/2$ . Then the  $(N - 1)$ -bit pattern  $(1, x_1, 1, x_2, \dots, 1, x_n)$  appears (twice) when scanning the original sequence. The rest of the proof is similar as in the case when  $N$  is even.  $\square$

## Linear Complexity

For purely periodic sequences the linear complexity  $L$  is equal to the degree of the minimal polynomial  $f(x)$ . Recall that  $f(x)$  is defined as the characteristic polynomial of the shortest linear recursion satisfied by the sequence (see [2]). Furthermore, the minimum period of the sequence is the smallest positive integer  $P$  such that  $f(x)$  divides  $x^P - 1$ . For a self-shrunken  $m$ -LFSR-sequence the linear complexity satisfies a lower bound as given in Theorem 3.

**Theorem 3** *The linear complexity  $L$  of a self-shrunken maximum length LFSR-sequence produced by an LFSR of length  $N$  satisfies*

$$L > 2^{\lfloor N/2 \rfloor - 1}. \quad (2)$$

**Proof.** By Proposition 1 and Theorem 2 the period  $P$  of a self-shrunken  $m$ -LFSR-sequence  $\mathbf{s}$  divides  $2^{N-1}$ , i.e., is of the form  $P = 2^a$  for some integer  $a \geq \lfloor N/2 \rfloor$ . Hence over  $GF(2)$ ,  $x^P - 1$  can be written as  $x^P - 1 = (x - 1)^{2^a}$ . Thus the condition  $f(x) \mid (x^P - 1)$  implies that  $f(x)$  is of the form  $f(x) = (x - 1)^L$  where  $L$  is the linear complexity of the sequence  $\mathbf{s}$ . We claim that  $L > 2^{a-1}$ .

Suppose to the contrary that  $L \leq 2^{a-1}$ . Then  $f(x) = (x - 1)^L$  would divide  $(x - 1)^{2^{a-1}} = x^{2^{a-1}} - 1$ . Thus  $x^{2^{a-1}} - 1$  would be the characteristic polynomial of a recursion satisfied by  $\mathbf{s}$ . This recursion would be  $s_n = s_{n-2^{a-1}}$  which contradicts to the fact that the minimum period is  $2^a$ .  $\square$

It is a common assumption in the analysis of the shrinking generator [1] or clock-controlled generators that the two LFSRs involved are independent. This allows for example to decimate the process of generating the output sequence with the period of the controlling LFSR. The output sequence obtained in this way can be considered as a decimated sequence of the controlled LFSR. This allows to apply the theory of LFSR-sequences to derive results on the period and linear complexity of the generated output sequence. This approach cannot be applied to the self-shrinking generator as the controlling and the controlled part cannot be separated from one another. For this reason the exact computation of the period and the linear complexity of a self-shrunken  $m$ -LFSR-sequence appears to be difficult. The bounds given in Theorems 2 and 3 are rough estimates. Experimental results as given in Section IV support the conjecture that the period  $P$  is maximal for LFSR-length  $N > 3$ , i.e.,  $P = 2^{N-1}$ . For the linear complexity  $L$  this would imply that  $L$  is bounded by  $2^{N-2} < L \leq 2^{N-1}$ . Nevertheless the bounds as given in Theorems 2 and 3 are far sufficient for practical applications. For example for  $N = 200$ , period and linear complexity are proved to be at least  $10^{30}$ .

## IV Examples and Experimental Results

By the analysis in Section III the period of a shrunken  $m$ -LFSR-sequence generated by an LFSR of length  $N$  is at most  $2^{N-1}$ . So far we have found only one example where the period does not reach this maximum value.

**Example.** Consider the  $m$ -LFSR of length  $N = 3$  defined by the recursion  $a_n = a_{n-2} + a_{n-3}$ . The output sequence of this LFSR with initial state 111 is the sequence 11100101110010... with period 7. The sequence of pairs for self-shrinking is 11, 10, 01, 01, 11, 00, 10, ..., with period 8, yielding the shrunken output sequence 1,0,1,0, ..., with period only 2 instead of the maximum possible value 4.

The other  $m$ -LFSR of length  $N = 3$  has the recursion  $a_n = a_{n-1} + a_{n-3}$ . Initialized with 111 the self-shrunken sequence 1,1,0,0, ..., of period 4 is obtained. Note that the self-shrunken output sequence of an  $m$ -LFSR of length  $N = 3$  must be balanced and 4-periodic. There are only two such sequences, namely 1010 ... and 1100 ... with minimum period 2 and 4, respectively. As for  $N = 3$ , the two self-shrunken  $m$ -LFSR-sequences are different, one of them must have period 2.

Experiments have shown that there are no other  $m$ -LFSRs of length  $N < 20$  for which the self-shrunken sequence does not attain maximum period  $2^{N-1}$ . This has been confirmed by exhausting all  $m$ -LFSRs of length  $N < 20$ . Table 1 shows the minimum and the maximum value of the linear complexity taken over all self-shrunken  $m$ -LFSRs of given LFSR-length  $N$  for  $N \leq 15$ .

| <i>LFSR-length N</i> | <i># of m-LFSR</i> | <i>Minimum LC</i> | <i>Maximum LC</i> | $\delta$ |
|----------------------|--------------------|-------------------|-------------------|----------|
| 2                    | 1                  | 2                 | 2                 | 0        |
| 3                    | 2                  | 2                 | 3                 | 1        |
| 4                    | 2                  | 5                 | 5                 | 3        |
| 5                    | 6                  | 10                | 13                | 3        |
| 6                    | 6                  | 25                | 28                | 4        |
| 7                    | 18                 | 54                | 59                | 5        |
| 8                    | 16                 | 118               | 122               | 6        |
| 9                    | 48                 | 243               | 249               | 7        |
| 10                   | 60                 | 498               | 504               | 8        |
| 11                   | 176                | 1009              | 1015              | 9        |
| 12                   | 144                | 2031              | 2038              | 10       |
| 13                   | 630                | 4072              | 4085              | 11       |
| 14                   | 756                | 8170              | 8180              | 12       |
| 15                   | 1800               | 16362             | 16371             | 13       |

Table 1: Minimum and maximum linear complexity of self-shrunken  $m$ -LFSRs

Commenting Table 1, we first note that for a sequence with an even number of 1's within the period  $P$ , the maximum possible linear complexity is  $P - 1$ , as  $\sum_{i=0}^{P-1} s_{n-i} = 0$ . For self-shrunken  $m$ -LFSR-sequences, maximum and minimum value of the linear complexity appear to be close to each other and very close to the maximum possible value  $2^{N-1} - 1$ .

Furthermore Table 1 shows a remarkable property: Except for  $N = 4$ , the upper bound attained for the linear complexity is  $2^{N-1} - \delta$ , where  $\delta = N - 2$ . This upper bound also holds for the exceptional case  $N = 4$ . Hence, for  $2 \leq N \leq 15$ ,  $(x^{2^{N-1}} - 1)/(x - 1)^{N-2}$  is a characteristic polynomial of any self-shrunken  $m$ -LFSR-sequence produced by an LFSR of length  $N$ . This fact can be viewed as an algebraic property of the self-shrunken LFSR-sequence that persists although most of the algebraic structure of the original sequence  $m$ -LFSR-sequence has been destroyed.

## V Cryptanalysis

In this section we discuss some approaches for possible cryptanalytic attacks and their complexities. We start with a general method for reconstructing the original sequence from a known portion of the self-shrunken sequence. This method is not restricted to the case where the original sequence is produced by an LFSR.

Assume that  $(s_0, s_1, \dots)$  is the known portion of the self-shrunken sequence. The bit  $s_0$  is produced by a bit pair  $(a_j, a_{j+1})$  of the original sequence where the index  $j$  is unknown. Our aim is to reconstruct the original sequence in forward direction beginning with position  $j$ . As we

know  $s_0$  we conclude that  $a_j = 1$  and  $a_{j+1} = s_0$ . For the next bit pair  $(a_{j+2}, a_{j+3})$  there remain three possibilities, namely  $a_{j+2} = 1, a_{j+3} = s_1$  if the bit pair was used to produce  $s_1$ , or the two alternatives  $a_{j+2} = 0, a_{j+3} = 0$  and  $a_{j+2} = 0, a_{j+3} = 1$  if the bit pair was discarded. For each of the three possibilities there are again three alternatives for the next bit pair. Therefore, for reconstructing  $n$  bit pairs, i.e.,  $N = 2n$  bits, we obtain a total of

$$S = 3^{n-1} \approx 3^{N/2} = 2^{((\log_2 3)/2)N} = 2^{0.79 \cdot N} \quad (3)$$

possible solutions. However the solutions have different probabilities. We explain this fact by considering the above bit pair  $(a_{j+2}, a_{j+3})$ . Assuming that the original sequence is purely random,  $a_{j+2} = 1$  with probability  $1/2$ . Hence the first alternative has probability  $1/2$  and the other two cases have probability  $1/4$ . In terms of information theory the uncertainty about the bit pair is

$$H = -(1/2)\log_2(1/2) - (1/4)\log_2(1/4) - (1/4)\log_2(1/4) = 3/2.$$

As for the reconstruction the individual bit pairs are supposed to be independent from each other, the total entropy for  $n$  bit pairs is  $3n/2$ . Therefore the optimum strategy for reconstructing  $N$  bits of the original sequence has average complexity  $2^{3N/4}$ . For example, for  $N = 200$ , this complexity is equivalent to an exhaustive search over a key of size 150 bit.

So far we did not take into account that the original sequence is produced by an LFSR. For cryptographic applications the key consists of the initial state and preferably also of the LFSR feedback connection. In order to assess the security we assume that the feedback connection is known. With this assumption we estimate the difficulty of finding the initial state (or the key) of the LFSR. For the above method of finding the key the average complexity is upper bounded by  $2^{3N/4}$ , where  $N$  is the length of the LFSR. If there are only few feedback taps or if they are concentrated around few locations, there are cases where faster attacks are possible, as will be shown below. On the other hand, if we exclude such special situations we know of no better method than reconstructing the initial state of the LFSR as described above.

Suppose for example that the LFSR only has two feedback taps (which is the smallest number of feedback taps for a  $m$ -LFSR). Then the feedback relation can be written as  $a_k + a_{k+t} + a_{k+t+s} = 0$ , for all  $k \in \mathbb{N}$ . Let  $a_j$  be the bit of the original sequence which determines the first known bit, say  $s_0$ , of the shrunken sequence. Our aim is to do an exhaustive search over the two  $m$ -bit blocks

$$\begin{aligned} \mathcal{B}_1 &= (a_j, a_{j+1}, \dots, a_{j+m-1}) \\ \mathcal{B}_2 &= (a_{j+t}, a_{j+t+1}, \dots, a_{j+t+m-1}) \end{aligned}$$

of suitably chosen size  $m$ . For every choice of the two blocks the third block

$$\mathcal{B}_3 = (a_{j+t+s}, a_{j+t+s+1}, \dots, a_{j+t+s+m-1})$$

is determined by the feedback relation. By self-shrinking there result three bit strings. The known segment of the self-shrunken sequence is scanned for the occurrence of these strings. For the correct choice of the  $m$ -bit blocks the three strings are expected to be about  $s/2$  or  $t/2$  bits apart from each other.

We call a block pair a solution if the three strings can be found at suitable positions. We investigate the problem regarding the number of solutions that are to be expected. According to (3) there are about  $3^{m/2}$  solutions for  $\mathcal{B}_1$ . If one knows the position of the substring in the

shrunken sequence which is produced by the second block  $\mathcal{B}_2$ , one again has about  $3^{m/2}$  solutions for  $\mathcal{B}_2$ . As this position is not exactly known, the number of solutions for  $\mathcal{B}_2$  is slightly larger. Thus we conclude that there are at least about  $3^{m/2} \cdot 3^{m/2} = 3^m$  solutions for the pair  $(\mathcal{B}_1, \mathcal{B}_2)$ . By the same argument we conclude that there are at least about  $3^{m/2}$  solutions for  $\mathcal{B}_3$ . It follows that with probability about  $p = 3^{m/2}/2^m$ , a random block  $\mathcal{B}_3$  is compatible with the shrunken sequence. Thus the number of solutions for the pair  $(\mathcal{B}_1, \mathcal{B}_2)$  is reduced by the factor  $p$  due to the recurrence relation. Therefore there remain about

$$T = 3^m \frac{3^{m/2}}{2^m} = \frac{3^{3m/2}}{2^m} = 2^{[3(\log_2 3)/2-1]m} = 2^{1.38 \cdot m} \quad (4)$$

solutions. For finding these solutions a search over  $2^{2m}$  block pairs is necessary.

In a similar way, with complexity  $2^{2m}$ , we do an exhaustive search over  $m$ -bit blocks

$$\begin{aligned} \mathcal{B}'_1 &= (a_{j-1}, a_{j-2}, \dots, a_{j-m}) \\ \mathcal{B}'_2 &= (a_{j+t-1}, a_{j+t-2}, \dots, a_{j+t-m}) \end{aligned}$$

in reverse direction from position  $j$ , or  $j + t$ , respectively. As for  $(\mathcal{B}_1, \mathcal{B}_2)$  there remain  $T = 2^{1.38 \cdot m}$  solutions for  $(\mathcal{B}'_1, \mathcal{B}'_2)$ . Every solution for  $(\mathcal{B}_1, \mathcal{B}_2)$  and  $(\mathcal{B}'_1, \mathcal{B}'_2)$  defines  $4m$  bits of the LFSR-sequence. Since  $N$  bits are required for reconstructing the original LFSR-sequence, we choose  $m = N/4$ . Thus the complexity of the search is  $2 \cdot 2^{N/2}$  with a possibility of  $T^2 = 2^{[3(\log_2 3)/2-1]N/2} = 2^{0.69 \cdot N}$  remaining solutions. The correct solution is singled out by trying all these possible solutions. This second exhaustive search obviously has complexity  $2^{0.69 \cdot N}$  which dominates the overall complexity of the attack. Thus the fact that the LFSR has only two feedback taps allows an attack that is slightly faster than the general method whose complexity is  $2^{0.75 \cdot N}$ .

The described method is a divide and conquer attack. The key is divided into two block pairs  $(\mathcal{B}_1, \mathcal{B}_2)$  and  $(\mathcal{B}'_1, \mathcal{B}'_2)$ , and the search for each block pair is done individually. It seems straightforward to extend the attack by searching for  $k$  rather than for two different  $m$ -bit block pairs. The complexity then would be  $k2^{2m}$  with  $(2^{1.38 \cdot m})^k$  possible solutions remaining. Each solution would determine  $2km$  bits of the LFSR-sequence. In order to obtain  $N$  bits we would choose  $k = N/(2m)$ . For  $k > 2$  the initial search has lower complexity. However the overall complexity is still dominated by the number of solutions which is  $(2^{1.38 \cdot m})^{N/(2m)} = 2^{0.69 \cdot N}$  as for  $k = 2$ .

It turns out that the attack is less effective if the number  $f$  of feedback taps increases. Corresponding to the feedback tap positions at the LFSR we would search for tuples  $(\mathcal{B}_1, \dots, \mathcal{B}_f)$  of  $m$ -bit blocks. Instead of (4), a number

$$T = (3^{m/2})^f \frac{3^{m/2}}{2^m} = 3^{((f+1)/2)m} 2^{-m} = 2^{[(\log_2 3)(f+1)/2-1]m} \quad (5)$$

of candidate solutions would remain after the search. Following the idea of divide and conquer we would search for at least  $k = 2$  such tuples. For  $k = 2$  these would determine  $2fm$  bits of the original LFSR-sequence. This suggests to choose  $m = N/(2f)$ . Thus the complexity of the search is again  $2 \cdot 2^{N/2}$  but with a possibility of

$$T^2 = 2^{[(\log_2 3)(f+1)/2-1]N/f} = 2^{[(\log_2 3)(1/2-1/(2f))]N} \quad (6)$$

solutions. For  $f = 4$  this quantity is  $2^{0.74 \cdot N}$ , and the asymptotic value, as  $f$  increases, is  $2^{((\log_2 3)/2)N} = 2^{0.79 \cdot N}$ . This coincides with the number of solutions (3) obtained for the general method.

The feasibility of the attack is further limited as the blocks become shorter. For shorter blocks the corresponding shrunken strings are more likely to appear accidentally in the shrunken sequence. This has the effect that it is more difficult to link the blocks with the corresponding positions in the shrunken sequence. Hence more incorrect solutions are likely to be accepted in the initial search.

## Acknowledgement

We are grateful to Christoph Günther for helpful comments.

## References

- [1] D. Coppersmith, H. Krawcyk, and Y. Mansour, *The Shrinking Generator*, Crypto'93, to appear.
- [2] S.W. Golomb, *Shift Register Sequences*, Aegean Park Press, 1982.
- [3] J.L. Massey, “Shift Register Synthesis and BCH Decoding”, *IEEE Transactions on Information Theory*, Vol. IT-15, pp. 122–127, 1969.
- [4] R.A. Rueppel, “When Shift Registers Clock Themselves”, *Advances in Cryptology—Eurocrypt'87, Proceedings*, pp. 53–64, Springer-Verlag, 1988.
- [5] R.A. Rueppel et al., “Verfahren und Schaltanordnung zum Erzeugen einer Pseudozufallsfolge sowie deren Verwendung”, *Deutsche Patentanmeldung P 43 01 279.5*, 1993.

# Construction and Decoding of Optimal Group Codes from Finite Reflection Groups

Thomas Mittelholzer

Center for Magnetic Recording Research \*

University of California, San Diego

La Jolla, CA 92093-0401, USA

Dedicated to James L. Massey on the occasion of his 60<sup>th</sup> birthday.

## Abstract

In this paper, the theory of finite Coxeter groups is applied to group codes. The class of group codes generated by finite Coxeter groups is a generalization of the well-known permutation modulation codes of Slepian. As a main result, a simple set of linear equations is given to characterize the optimal solution to a restricted initial point problem for all these codes. In particular, it is found that Ingemarsson's solution to the initial point problem for permutation modulation without sign changes is not always optimal. Moreover, a list of new good group codes in dimension 8 is presented. Finally, a new maximum-likelihood decoding algorithm is presented that has a reasonably low complexity and that applies to all codes generated by finite Coxeter groups.

## I Introduction

Group codes have been introduced by Slepian (cf. [1], [2]) more than 25 years ago and, since then, this topic has been intensely investigated (cf. [3] for a survey in this field). A group code  $C$  is an orbit of some initial point  $\mathbf{x} \in \mathbb{R}^n$  under the action of an isometry group  $\mathcal{G}$  of the n-dimensional Euclidean space, i.e.,

$$C = \{T \cdot \mathbf{x} \in \mathbb{R}^n : T \in \mathcal{G}\}.$$

The quality of a code  $C$  is determined by its cardinality  $M = |C|$  and its minimum Euclidean distance  $d_{min}$  between different codewords. For some given dimension  $n$ , the parameters  $M$  and  $d_{min}$  should be made as large as possible by appropriate choice of the isometry group  $\mathcal{G}$  and some initial point  $\mathbf{x} \in \mathbb{R}^n$ . This problem seems to be difficult because little is known about a good choice of the isometry group and the task of finding the best initial point is only partially solved for the subclass of permutation modulation codes (cf. [4]).

In this paper, we confine our attention to finite reflection groups, which are also called finite Coxeter groups (cf. [5], [6]). This class of groups is a quite limited class of isometry

groups but it subsumes the groups used for permutation modulation. The particular reason to consider finite reflection groups is the existing detailed theory on these groups that gives among many other results a relatively simple description and classification of these groups. This allows one to give a new approach to the construction of group codes using the description of finite reflection groups by their root systems. In particular, it is shown that a restricted initial point problem has a canonical solution. Furthermore, a well-known theorem of Iwahori (cf. Chap. 6 of [5]) allows one to devise an efficient maximum-likelihood decoding algorithm for the AWGN channel valid for all finite reflection groups.

## II Finite Reflection Groups

A *finite reflection group*  $\mathcal{G}$  is a finite group generated by reflections in the  $N$ -dimensional Euclidean space  $\mathbb{R}^N$  (here we use  $N$  to denote the dimension of the space on which  $\mathcal{G}$  operates and we reserve  $n$  for the rank of the reflection group, which is defined below). Every reflection  $S \in \mathcal{G}$  leaves some hyperplane  $H$  invariant. If  $\mathbf{r} \in \mathbb{R}^N$  denotes a normal vector of  $H$ , then the reflection  $S$  acts on an element  $\mathbf{x} \in \mathbb{R}^N$  by the formula

$$S\mathbf{x} = \mathbf{x} - 2 \frac{\mathbf{x} \cdot \mathbf{r}}{\mathbf{r} \cdot \mathbf{r}} \mathbf{r},$$

where  $\mathbf{x} \cdot \mathbf{r}$  denotes the dot product between  $\mathbf{x}$  and  $\mathbf{r}$ . We may write  $S = S_{\mathbf{r}}$  and  $H = H_{\mathbf{r}}$  to denote the reflection  $S$  with hyperplane  $H$  determined by  $\mathbf{r}$ . Note that  $S_{\mathbf{r}}$  and  $H_{\mathbf{r}}$  do not depend on the length of the normal vector  $\mathbf{r}$ . In the sequel, we will choose for each reflection  $S$ , and hence, for each reflecting hyperplane  $H$ , one single normal vector of some fixed length. Such a normal vector  $\mathbf{r}$  and its negative  $-\mathbf{r}$  are called *roots* of  $\mathcal{G}$ . The set of all roots of  $\mathcal{G}$  is called the *root system*  $\Delta$  of the finite reflection group  $\mathcal{G}$ , i.e.,

$$\Delta = \{\pm \mathbf{r} \in \mathbb{R}^N : S_{\mathbf{r}} \in \mathcal{G}\}.$$

Conjugation of a reflection  $S_{\mathbf{r}}$  by some orthogonal  $N \times N$ -matrix  $T$  yields again a reflection, which is given by the formula (cf. Prop. 1.2 in [6])

$$TS_{\mathbf{r}}T^{-1} = S_{T\mathbf{r}}. \quad (1)$$

**Example 1** Consider the dihedral group  $\mathcal{H}_2^3$  of order 6, which is the symmetry group of the regular triangle. It is generated by the two reflections along the roots  $\mathbf{r}_1 = [0, 1]$  and  $\mathbf{r}_2 = [\sqrt{3}/2, -1/2]$ . The group  $\mathcal{H}_2^3$  contains one further reflection with root  $\mathbf{r}_3 = [\sqrt{3}/2, 1/2]$  and, hence, its root system is  $\Delta = \{\pm \mathbf{r}_1, \pm \mathbf{r}_2, \pm \mathbf{r}_3\}$ . The roots  $\mathbf{r}_i$  and the reflecting hyperplanes  $H_{\mathbf{r}_i}$  are shown in Figure 1. The remaining three elements of  $\mathcal{H}_2^3$  are the identity and the two rotations given by  $S_{\mathbf{r}_1}S_{\mathbf{r}_2}$  and  $(S_{\mathbf{r}_1}S_{\mathbf{r}_2})^2$ .

**Example 2**  $\mathcal{B}_n$ ,  $n \geq 2$ , denotes the orthogonal group corresponding to permutation modulation with sign changes. A element  $T \in \mathcal{B}_n$  acts on a point  $\mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^n$  by

$$T\mathbf{x} = [\epsilon_1 x_{i_1}, \epsilon_2 x_{i_2}, \dots, \epsilon_n x_{i_n}],$$

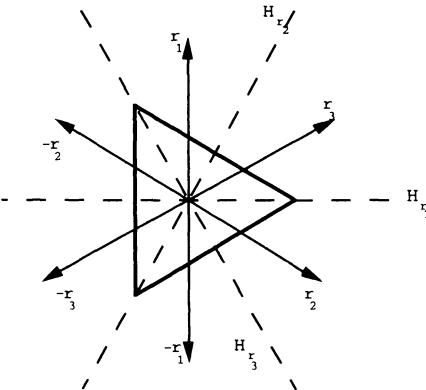


Figure 1: Roots and reflecting hyperplanes of the symmetry group of the regular triangle.

where  $i_1, i_2, \dots, i_n$  determines a permutation and the  $\epsilon_i$  denote the sign changes. It is clear that  $\mathcal{B}_n$  is a group and its order is  $2^n \cdot n!$ . The permutation group of  $n$  letters is generated by the  $n - 1$  transpositions  $(i \ i - 1)$ ,  $i = 2, \dots, n$ . These transpositions correspond to the reflections  $S_i = S_{\mathbf{e}_i - \mathbf{e}_{i-1}}$  with roots  $\mathbf{r}_i = \mathbf{e}_i - \mathbf{e}_{i-1}$ , where the  $\mathbf{e}_i$  denote the standard basis vectors in  $\mathbb{R}^n$ . The sign changes are generated by the  $n$  reflections  $S_{\mathbf{e}_1}, \dots, S_{\mathbf{e}_n}$  along the basis vectors. Since, by (1),  $S_{\mathbf{e}_i} = S_i S_{\mathbf{e}_{i-1}} S_i^{-1} = S_{S_i(\mathbf{e}_{i-1})}$ , one can obtain the reflections  $S_{\mathbf{e}_i}$ ,  $i \geq 2$ , from  $S_{\mathbf{e}_1}$  and  $S_2, S_3, \dots, S_n$ . Hence,  $\mathcal{B}_n$  is a reflection group that is generated by the  $n$  reflections  $S_1 = S_{\mathbf{e}_1}, S_2, \dots, S_n$ .

It is clear that a finite reflection group  $\mathcal{G}$  is generated by all the reflections  $S_{\mathbf{r}}, \mathbf{r} \in \Delta$ . But not all roots  $\mathbf{r}$  are required to generate  $\mathcal{G}$ , because  $\mathbf{r}$  and  $-\mathbf{r}$  determine the same reflection. This leads one to partition  $\Delta$  into a set of ‘positive’ and ‘negative’ roots, and to search for a minimal set of generators  $S_{\mathbf{r}_1}, S_{\mathbf{r}_2}, \dots, S_{\mathbf{r}_m}$  determined by positive roots  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m$ .

To this end, we choose a vector  $\mathbf{t} \in \mathbb{R}^N$  such that  $\mathbf{t} \cdot \mathbf{r} \neq 0$  for every  $\mathbf{r} \in \Delta$ . We partition the root system  $\Delta$  into the subset of *positive roots* and *negative roots* (with respect to  $\mathbf{t}$ ), viz.

$$\Delta_{\mathbf{t}}^+ = \{\mathbf{r} \in \Delta : \mathbf{t} \cdot \mathbf{r} > 0\}$$

and

$$\Delta_{\mathbf{t}}^- = \{\mathbf{r} \in \Delta : \mathbf{t} \cdot \mathbf{r} < 0\}.$$

We will single out a minimal subset  $\Pi$  of  $\Delta_{\mathbf{t}}^+$  by the following requirement: Every positive root  $\mathbf{r} \in \Delta_{\mathbf{t}}^+$  is a linear combination of elements of  $\Pi$  with all coefficients nonnegative. Such a subset  $\Pi$  will be called a *base* (or  $\mathbf{t}$ -*base*) for  $\Delta$  and the elements of  $\Pi$  are called *simple roots*. It is not obvious, but one can show that every positive system  $\Delta_{\mathbf{t}}^+$  contains a unique  $\mathbf{t}$ -base  $\Pi$  (cf. Chap. 1.3 in [6] or Chap. 4.1 in [5]).

**Example 1 (continued)** It is obvious from Figure 1 that the roots  $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$  are positive with respect to the vector  $\mathbf{t} = \mathbf{r}_3$  and the other three roots are negative. Since  $\mathbf{r}_3$  is a linear

combination of  $\mathbf{r}_1$  and  $\mathbf{r}_2$  with *nonnegative* coefficients, it follows that the unique t-base is given by  $\Pi = \{\mathbf{r}_1, \mathbf{r}_2\}$ .

In the sequel, we shall consider only irreducible finite reflection groups, i.e., those that are not direct products of smaller reflection groups. Every reflection group is irreducible or decomposes into a direct product of irreducible reflection groups and, moreover, there are only ten different types of irreducible finite reflection groups (cf. [6], [5]). All the irreducible reflection groups can be characterized by their t-base  $\Pi$  because  $\Pi$  is unique for each group. In Table 1 (cf. Table 5.1 in [5]), one particular choice of a t-base is given for each irreducible Coxeter group. The index (= the subscript)  $n$  of each group denotes the rank of the corresponding group, which is defined to be the dimension of the linear span of the root system.

E.g., the graph  $\mathcal{A}_n$  has a t-base with coordinates in  $\mathbb{R}^{n+1}$ , but all roots are orthogonal to the vector  $[1, 1, \dots, 1]$ , i.e., they span an  $n$ -dimensional hyperplane in  $\mathbb{R}^{n+1}$ . Thus, this reflection group has rank  $n$ . Table 1 also contains the size  $|\Delta|$  of the root system of each irreducible finite reflection group, which is a decisive parameter for the complexity of the decoding algorithm given in the last section of this paper. This table summarizes all basic information about finite reflection groups that is needed for our coding and decoding applications.

| Group             | Base  | $ \Delta $ |
|-------------------|---|------------|
| $\mathcal{A}_n$   | $\mathbf{r}_i = \mathbf{e}_{i+1} - \mathbf{e}_i, 1 \leq i \leq n.$  | $n^2 + n$  |
| $\mathcal{B}_n$   | $\mathbf{r}_1 = \mathbf{e}_1, \mathbf{r}_i = \mathbf{e}_i - \mathbf{e}_{i-1}, 2 \leq i \leq n.$   | $2n^2$     |
| $\mathcal{D}_n$   | $\mathbf{r}_1 = \mathbf{e}_1 + \mathbf{e}_2, \mathbf{r}_i = \mathbf{e}_i - \mathbf{e}_{i-1}, 2 \leq i \leq n.$  | $2n(n-1)$  |
| $\mathcal{E}_6$   | $\mathbf{r}_1 = \frac{1}{2}(\sum_1^3 \mathbf{e}_i - \sum_4^8 \mathbf{e}_i), \mathbf{r}_i = \mathbf{e}_i - \mathbf{e}_{i-1}, 2 \leq i \leq 6.$   | 72         |
| $\mathcal{E}_7$   | $\mathbf{r}_1 = \frac{1}{2}(\sum_1^3 \mathbf{e}_i - \sum_4^8 \mathbf{e}_i), \mathbf{r}_i = \mathbf{e}_i - \mathbf{e}_{i-1}, 2 \leq i \leq 7.$   | 126        |
| $\mathcal{E}_8$   | $\mathbf{r}_1 = \frac{1}{2}(\sum_1^3 \mathbf{e}_i - \sum_4^8 \mathbf{e}_i), \mathbf{r}_i = \mathbf{e}_i - \mathbf{e}_{i-1}, 2 \leq i \leq 8.$   | 240        |
| $\mathcal{F}_4$   | $\mathbf{r}_1 = -\frac{1}{2}(\sum_1^4 \mathbf{e}_i), \mathbf{r}_2 = \mathbf{e}_1, \mathbf{r}_i = \mathbf{e}_{i-1} - \mathbf{e}_{i-2}, 3 \leq i \leq 4.$   | 48         |
| $\mathcal{I}_3$   | $\mathbf{r}_1 = \beta[2\alpha + 1, 1, -2\alpha], \mathbf{r}_2 = \beta[-2\alpha - 1, 1, 2\alpha],$<br>$\mathbf{r}_3 = \beta[2\alpha, -2\alpha - 1, 1], \text{ where } \alpha = \cos(\pi/5), \beta = \cos(2\pi/5).$   | 30         |
| $\mathcal{I}_4$   | $\mathbf{r}_1 = \beta[2\alpha + 1, 1, -2\alpha, 0], \mathbf{r}_2 = \beta[-2\alpha - 1, 1, 2\alpha, 0],$<br>$\mathbf{r}_3 = \beta[2\alpha, -2\alpha - 1, 1, 0], \mathbf{r}_4 = \beta[-2\alpha, 0, -2\alpha - 1, 1].$ | 120        |
| $\mathcal{H}_2^m$ | $\mathbf{r}_1 = [1, 0], \mathbf{r}_2 = [-\cos(\pi/m), \sin(\pi/m)].$  | $2m$       |

Table 1: Root bases of irreducible finite reflection groups

### III Construction of Optimal Group Codes

The optimality criterion that we use deviates from the usual criterion. Usually, one fixes the dimension  $n$  and the size  $M = |\mathcal{C}|$  of the code and seeks to maximize the minimum Euclidean distance  $d_{min}$ . In this paper, we do not only fix the size of the code but, additionally, we fix the stabilizer (= isotropy group) that leaves the initial point fixed. This simpler problem has a canonical solution for all finite reflection groups. The usual standard initial point

problem is harder and has only been completely solved for permutation modulation with sign changes (cf. [4]). The solution of Ingemarsson in [4] for permutation modulation without sign changes is not complete and is not always optimal, which will be illustrated by two examples.

We start with a given dimension  $n$  and a finite reflection group  $\mathcal{G}$  of rank  $n$ . The parameters  $M = |\mathcal{C}|$  and  $d_{min}$  of the code  $\mathcal{C}$  are clearly determined by the choice of some isotropy group  $\mathcal{H} = \text{stab } \mathbf{x}$  of the initial point  $\mathbf{x}$ . The simplicity of the proposed optimal code construction relies on the fact that the isotropy group  $\text{stab } \mathbf{x}$  has a simple description in terms of a particular set of simple roots as given below in (2). A well-known theorem (cf. Theorem 1.12 in [6]) implies that  $\text{stab } \mathbf{x}$  is itself a finite reflection group that is generated by those reflections that it contains. By making a suitable choice of the  $\mathbf{t}$ -base  $\Pi$  (or equivalently, by taking an equivalent initial point  $\mathbf{x}' = T\mathbf{x}, T \in \mathcal{G}$ ), one can write

$$\text{stab } \mathbf{x} = \langle S_{\mathbf{r}} : S_{\mathbf{r}}\mathbf{x} = \mathbf{x}, \mathbf{r} \in \Pi \rangle.$$

The unique simple roots that are singled out by the condition  $S_{\mathbf{r}}\mathbf{x} = \mathbf{x}$  or, equivalently, by  $\mathbf{r} \cdot \mathbf{x} = 0$ , will be called *passive roots* and the set of all passive roots will be denoted by

$$\Pi_p = \{\mathbf{r} \in \Pi : \mathbf{r} \cdot \mathbf{x} = 0\}. \quad (2)$$

The set of passive roots  $\Pi_p$  determines the stabilizer and, hence, its knowledge allows one to compute the order  $|\text{stab } \mathbf{x}|$  (cf. [8]) and the size  $M = \frac{|\mathcal{G}|}{|\text{stab } \mathbf{x}|}$  of the code.

The set of passive roots  $\Pi_p$  is also the key for the computation of the optimal initial point  $\mathbf{x}_0$ , which is characterized in the following theorem.

**Theorem 1** Let  $\mathcal{G}$  be a finite reflection group of rank  $n$  with normalized simple roots  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n$ , i.e.,  $\|\mathbf{r}_i\| = 1$ . Let  $\Pi_p \subset \Pi = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n\}$  be a set of passive roots, which determine a subgroup  $\mathcal{H} = \langle S_{\mathbf{r}} : \mathbf{r} \in \Pi_p \rangle$ . The optimal initial point  $\mathbf{x}_0$  of a group code  $\mathcal{C} = \{T\mathbf{x}_0 : T \in \mathcal{G}\}$  with stabilizer  $\text{stab } \mathbf{x}_0 = \mathcal{H}$  is given by the following  $n + 1$  equations

$$\mathbf{r} \cdot \mathbf{x}_0 = 0 \quad \text{for all } \mathbf{r} \in \Pi_p, \quad (3)$$

$$\mathbf{r} \cdot \mathbf{x}_0 = \lambda \quad \text{for all } \mathbf{r} \in \Pi \setminus \Pi_p \text{ and some } \lambda > 0, \quad (4)$$

$$\mathbf{x}_0 \cdot \mathbf{x}_0 = 1 \quad (\text{normalization}). \quad (5)$$

Moreover, the minimum Euclidean distance of  $\mathcal{C}$  is  $d_{min} = 2\lambda$ .

It is intuitively clear that the conditions (3) and (4) must hold. Indeed, (3) implies that  $\mathbf{x}_0$  has the prescribed isotropy group  $\mathcal{H}$  and (4) guarantees that  $\mathbf{x}_0$  has the same distance from all reflecting hyperplanes that correspond to nonpassive roots. A formal proof of this theorem is given in [8].

**Example 3** Consider the reducible reflection group  $\mathcal{G}$  acting on  $\mathbb{R}^2$  with simple roots  $\mathbf{r}_1 = [1, 0]$  and  $\mathbf{r}_2 = [0, 1]$ . This group has order 4 and is isomorphic to  $\mathbb{Z}_2 \times \mathbb{Z}_2$ . If one chooses the stabilizer to be trivial, then Theorem 1 yields  $\mathbf{x}_0 = \frac{1}{\sqrt{2}}[1, 1]$  for the optimal initial point of the 4-element group code  $\mathcal{C}$ . This group code corresponds to QPSK modulation.

**Example 4** (The two-dimensional simplex code). Consider permutation modulation of length-3 vectors without sign changes. The underlying reflection group is  $\mathcal{A}_2$  (which is isomorphic to the dihedral group  $\mathcal{H}_2^3$ ). Using Table 1, we find that the root base  $\Pi$  consists of the two vectors  $\mathbf{r}_1 = \mathbf{e}_2 - \mathbf{e}_1$  and  $\mathbf{r}_2 = \mathbf{e}_3 - \mathbf{e}_2$ , which are both orthogonal to  $[1, 1, 1]$ . The group  $\mathcal{A}_2$  operates effectively on the two-dimensional subspace  $V = \{\mathbf{x} : \mathbf{x} \cdot [1, 1, 1] = 0\}$ . We select an isotropy group  $\mathcal{H}$  by choosing the set of passive roots to be  $\Pi_p = \{\mathbf{r}_2\}$ , i.e.,  $\mathcal{H} = \langle S_{\mathbf{r}_2} \rangle = \{1, S_{\mathbf{r}_2}\}$ . It follows that the group code  $C$  has  $M = \frac{|\mathcal{A}_2|}{|\mathcal{H}|} = 3$  codewords. One readily checks that the initial point  $\mathbf{x}_0 = \frac{1}{\sqrt{6}}[-2, 1, 1]$  satisfies the equations of Theorem 1 and it also satisfies  $\mathbf{x}_0 \cdot [1, 1, 1] = 0$ , which ensures that  $\mathbf{x}_0$  lies in  $V$ . The minimum Euclidean distance is  $d_{min} = 2 \frac{\|\mathbf{x}_0 \cdot \mathbf{r}_1\|}{\|\mathbf{r}_1\|} = \sqrt{3}$ .

This code  $C$  is the only permutation modulation code of length 3 with rate  $R = \frac{1}{3} \log_2 3$ . Hence,  $C$  is an example of an optimal permutation modulation code, which has a nonsymmetrical initial point, i.e., the initial point is not of the form  $[-\mu, 0, \mu]$ . This shows that Ingemarsson's proposed optimal solution in [4] is not complete (some optimal codes are missing), since all his initial points are required to be symmetrical.

**Example 5** Consider again permutation modulation of vectors of length  $n + 1 = 22$  with underlying reflection group  $\mathcal{A}_{21}$ . Within the base  $\Pi = \{\mathbf{e}_{i+1} - \mathbf{e}_i : 1 \leq i \leq n\}$ , we choose the following set of passive roots

$$\begin{aligned} \Pi_p = & \{\mathbf{e}_3 - \mathbf{e}_2\} \cup \{\mathbf{e}_5 - \mathbf{e}_4, \mathbf{e}_6 - \mathbf{e}_5\} \cup \{\mathbf{e}_8 - \mathbf{e}_7, \mathbf{e}_9 - \mathbf{e}_8, \mathbf{e}_{10} - \mathbf{e}_9\} \\ & \cup \{\mathbf{e}_{12} - \mathbf{e}_{11}, \mathbf{e}_{13} - \mathbf{e}_{12}, \mathbf{e}_{14} - \mathbf{e}_{13}\} \cup \{\mathbf{e}_{16} - \mathbf{e}_{15}, \mathbf{e}_{17} - \mathbf{e}_{16}\} \\ & \cup \{\mathbf{e}_{19} - \mathbf{e}_{18}, \mathbf{e}_{20} - \mathbf{e}_{19}\} \cup \{\mathbf{e}_{22} - \mathbf{e}_{21}\} \end{aligned}$$

The set of passive roots is already decomposed into seven subsets, which correspond to mutually orthogonal roots. The two 1-element subsets generate Coxeter groups of type  $\mathcal{A}_1$ , the three 2-element subsets generate Coxeter groups of type  $\mathcal{A}_2$  and the two 3-element subsets generate Coxeter groups of type  $\mathcal{A}_3$ . It follows that the isotropy group  $\mathcal{H}$  has order  $|\mathcal{H}| = |\mathcal{A}_1|^2 \cdot |\mathcal{A}_2|^3 \cdot |\mathcal{A}_3|^2 = (2!)^2 \cdot (3!)^3 \cdot (4!)^2$ , and hence, the code  $C$  has cardinality  $M = \frac{22!}{(2!)^2 \cdot (3!)^3 \cdot (4!)^2}$ . The optimal initial point according to Theorem 1 is

$$\begin{aligned} \mathbf{x}_0 = & \frac{1}{\sqrt{39622}}[-83, -61, -61, -39, -39, -39, -17, -17, -17, -17, \\ & 5, 5, 5, 27, 27, 27, 49, 49, 49, 71, 71]. \end{aligned}$$

The minimum squared Euclidean distance of this code is  $d_{min}^2 = \frac{968}{39622} \approx 0.02443$ .

If one requires – as for Ingemarsson's solution (cf. [4]) – that the initial point is symmetrical and that the resulting code has the same cardinality  $M = \frac{22!}{2! \cdot 2! \cdot 3! \cdot 3! \cdot 4! \cdot 4!} = \frac{22!}{3! \cdot 3! \cdot 4! \cdot 4! \cdot 4!}$  as the previously constructed code  $C$ , then one obtains the following initial point

$$\mathbf{x}'_0 = \frac{1}{\sqrt{82}}[-4, -3, -2, -2, -2, -1, -1, -1, -1, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 3, 4].$$

This second code has a slightly smaller squared minimum Euclidean distance given by  $d_{min}^2 = \frac{1}{41} \approx 0.02439$ . This example shows that for fixed length and a fixed number  $M$  of codewords, the symmetrical initial point – as proposed in [4] – is not always optimal.

| $\mathcal{G}$   | $\Pi_p$  | $M$    | $R$  | $d_{min}^2$ | $\mathbf{x}_0$  |
|-----------------|--|--------|------|-------------|---|
| $\mathcal{E}_8$ | $\mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \mathbf{r}_7, \mathbf{r}_8$ | 17280  | 1.76 | 0.250       | $\frac{-1}{\sqrt{8}}[1, 1, 1, 1, 1, 1, 1, 1]$                               |
| $\mathcal{E}_8$ | $\mathbf{r}_1, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \mathbf{r}_7$               | 30240  | 1.86 | 0.200       | $\frac{-1}{\sqrt{10}}[2, 1, 1, 1, 1, 1, 1, 0]$                              |
| $\mathcal{E}_8$ | $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_7, \mathbf{r}_8$ | 60480  | 1.99 | 0.167       | $\frac{-\sqrt{3}}{4}[1, 1, 1, 1, 1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ |
| $\mathcal{E}_8$ | $\mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \mathbf{r}_7$               | 138240 | 2.13 | 0.125       | $\frac{-1}{8}[3, 3, 3, 3, 3, 3, 3, 1]$                                      |

Table 2: Some group codes from exceptional group  $\mathcal{E}_8$ .

| Length $n$ | $R$  | $d_{min}^2$ |
|------------|------|-------------|
| 8          | 1.64 | 0.250       |
| 8          | 1.77 | 0.181       |
| 8          | 1.84 | 0.166       |
| 8          | 1.92 | 0.128       |
| 8          | 2.08 | 0.100       |
| 8          | 2.14 | 0.083       |

Table 3: Best permutation modulation codes from  $\mathcal{B}_8$ .

A number of remarkably good group codes from the exceptional finite reflection groups have been determined in [7]. The best codes with underlying group  $\mathcal{E}_8$  are listed in Table 2. Each code is uniquely characterized by the set  $\Pi_p$  of passive roots, which determines a unique optimal initial point  $\mathbf{x}_0$ . For each code the following parameters are given: the cardinality  $M = |C|$ , the normalized rate  $R = \frac{1}{n} \log_2(M)$  and the squared minimum Euclidean distance  $d_{min}^2$ .

Table 3 contains the best permutation modulation codes for  $\mathcal{B}_8$  – as found in [7] – with rates that are comparable to the ones of Table 2. It is evident from these tables that the group codes from the exceptional reflection group  $\mathcal{E}_8$  outperform permutation modulation codes considerably.

Some of the codes relying on the group  $\mathcal{E}_8$  are well-known: The codes with 30 240 and 60 480 codewords are shells of the  $\mathcal{E}_8$ -lattice (cf. Chap. 4.8 in [9]). The other two codes with 17 280 and 138 240 codewords are only subsets but not full shells of the  $\mathcal{E}_8$ -lattice. The full shells contain 17 520 and 140 400 points and they decompose into two orbits under  $\mathcal{E}_8$ . One can show – using Table 4.10 in [9] – that these two shells do not have a codeword-independent distance spectrum, hence, they cannot be obtained as group codes.

## IV Decoding by Length Reduction

In this section, we present a general decoding algorithm that can be applied to all group codes generated by finite reflection groups. This algorithm applies to a broader class of codes than Slepian’s decoding algorithm for permutation modulation (cf. [2]); however, for the class of permutation modulation codes, Slepian’s algorithm is considerably faster than

the proposed algorithm. Thus, the proposed algorithm should rather be used for group codes generated by exceptional reflection groups, where Slepian's algorithm does not apply.

Henceforth, we will assume – as in the previous section – that the finite reflection group  $\mathcal{G}$  operates effectively on  $\mathbb{R}^n$ , i.e., we assume  $\mathbb{R}^n = V = \text{linear span of all roots of } \mathcal{G}$ . Let  $C = \{T\mathbf{x}_0 : T \in \mathcal{G}\} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{M-1}\}$  be a group code. Suppose the codewords of  $C$  are sent over an AWGN channel, i.e., the received signal is of the form  $\mathbf{y} = \mathbf{x}_i + \mathbf{z}$ , where  $\mathbf{x}_i = T\mathbf{x}_0$  is the transmitted codeword and  $\mathbf{z}$  denotes a zero-mean Gaussian noise sample. The task of a maximum-likelihood decoding algorithm is to determine (one of) the nearest codeword(s)  $\hat{\mathbf{x}}_i$  given  $\mathbf{y}$ . Equivalently, such a decoder will determine some element  $\hat{T} \in \mathcal{G}$  with  $\hat{T}\mathbf{x}_0 = \hat{\mathbf{x}}_i$ . In general,  $\hat{T}$  is not unique because every element in the left coset  $\hat{T}(\text{stab } \mathbf{x}_0)$  determines the same codeword  $\hat{\mathbf{x}}_i$ .

The length-reduction algorithm is based on the notion of length of an element  $T \in \mathcal{G}$ . The *length*  $l(T)$  of  $T$  is defined to be the smallest nonnegative integer  $r$  for which an expression  $S_1 S_2 \dots S_r$  exists with  $T = S_1 S_2 \dots S_r$  and  $S_i = S_{\mathbf{r}_i}$  for some  $\mathbf{r}_i \in \Pi$ . By convention,  $l(1) = 0$ . Using a theorem due to Iwahori (cf. Prop. 6.11 in [5]) and an explicit description of the decoding regions as given in [8], one obtains the following ML-decoding algorithm. It proceeds by reducing the length of  $\hat{T}$  by 1 each time the main loop (Steps 2 and 3) is executed.

### Length-Reduction Algorithm

- Input** The base  $\Pi = \{\mathbf{r}_1, \dots, \mathbf{r}_n\}$  of  $\mathcal{G}$  and some initial point  $\mathbf{x}_0 \in \mathbb{R}^n$ . The received signal point  $\mathbf{y} \in \mathbb{R}^n$ .
- Output**  $\hat{\mathbf{x}}_i$  = (one of) the nearest codeword(s) to  $\mathbf{y}$ .
- Step 1** (Initialization).  $L = \{\}$  empty list.
- Step 2** Determine  $N = \{\mathbf{r}_j \in \Pi : \mathbf{r}_j \cdot \mathbf{y} < 0\}$ .
- Step 3** (Length reduction). If  $N = \emptyset$  go to Step 4, else select some  $\mathbf{r}_j \in N$  and append  $j$  to the list  $L$ . Set  $\mathbf{y} \leftarrow S_{\mathbf{r}_j} \mathbf{y} = \mathbf{y} - 2 \frac{\mathbf{y} \cdot \mathbf{r}_j}{\|\mathbf{r}_j\|^2} \mathbf{r}_j$  and go to Step 2.
- Step 4** (Presentation of output). Apply successive reflections to  $\mathbf{x}_0$  according to the list  $L = \{j_1, j_2, \dots, j_l\}$ , i.e.,

$$\hat{\mathbf{x}}_i = S_{\mathbf{r}_{j_1}} S_{\mathbf{r}_{j_2}} \dots S_{\mathbf{r}_{j_l}} \mathbf{x}_0$$

and terminate.

A worst-case analysis of this algorithm is easily obtained by counting the number of additions and multiplications of real numbers in each step. In Step 2, one needs  $n$  comparisons and no more than  $n^2$  additions/multiplications are required. Step 3 requires no more than  $2n$  additions/multiplications (here we assume that  $\|\mathbf{r}\| = 1$ ). The main loop, namely Steps 2 and 3, will be repeated at most  $\frac{|\Delta|}{2}$  times because  $\frac{|\Delta|}{2}$  is the maximum length of an element  $T \in \mathcal{G}$  (cf. Chap. 1.6 and 1.8 in [6]). Hence, the algorithm requires no more than  $\frac{|\Delta|}{2}(n^2 + 2n)$  additions/multiplications and  $\frac{|\Delta|}{2}n$  comparisons to build up the list  $L$ . In the last step (Presentation of output), the number of additions/multiplications is bounded above by  $\frac{|\Delta|}{2} \cdot 2n$ . Thus, the overall complexity on the number of additions/multiplications is bounded by

$$\frac{|\Delta|}{2}(n^2 + 4n). \quad (6)$$

For irreducible  $\mathcal{G}$ , the value of Expression (6) is immediately obtained from Table 1. Evaluation of (6) yields that the complexity of this algorithm is roughly  $n^4$ , which shows that the algorithm is efficient compared to a full search through all codewords. However, for the subclass of permutation modulation codes, one should apply Slepian's decoding algorithm, which is considerably faster than the Length-Reduction algorithm. Note that when  $\mathcal{G}$  is reducible, say  $\mathcal{G} \cong \mathcal{G}_1 \times \mathcal{G}_2$ , one can decode for  $\mathcal{G}_1$  and  $\mathcal{G}_2$  separately, which yields an important reduction in complexity.

**Example 6** Let  $C$  be the code from  $\mathcal{E}_8$  with 138 240 codewords and initial point  $\mathbf{x}_0 = \frac{-1}{8}[3, 3, 3, 3, 3, 3, 3, 1]$  (cf. Table 2 above). Suppose the received signal point is  $\mathbf{y} = [-2.15, 1.65, 1.85, 2.05, 1.45, 1, 35, -0.25, 1.25]$ . We apply the Length-Reduction algorithm to  $\mathbf{y}$  and we denote by  $i$  the number of times the computation passes through the main loop (Steps 2 and 3) of the algorithm. The results of these computations are summarized in a table as follows.

| $i$ | $N$  | $L$                | $\mathbf{y}$  |
|-----|--|--------------------|---|
| 0   | -  | {}                 | $[-2.15, 1.65, 1.85, 2.05, 1.45, 1, 35, -0.25, 1.25]$ |
| 1   | $\mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_8$ | {3}                | $[-2.15, 1.85, 1.65, 2.05, 1.45, 1.35, -0.25, 1.25]$  |
| 2   | $\mathbf{r}_4, \mathbf{r}_8$               | {3, 4}             | $[-2.15, 1.85, 2.05, 1.65, 1.45, 1.35, -0.25, 1.25]$  |
| 3   | $\mathbf{r}_1, \mathbf{r}_3, \mathbf{r}_8$ | {3, 4, 1}          | $[-2.0, 1.7, 1.9, 1.8, 1.6, 1.5, -0.1, 1.4]$          |
| 4   | $\mathbf{r}_3, \mathbf{r}_8$               | {3, 4, 1, 3}       | $[-2.0, 1.9, 1.7, 1.8, 1.6, 1.5, -0.1, 1.4]$          |
| 5   | $\mathbf{r}_4, \mathbf{r}_8$               | {3, 4, 1, 3, 4}    | $[-2.0, 1.9, 1.8, 1.7, 1.6, 1.5, -0.1, 1.4]$          |
| 6   | $\mathbf{r}_8$                             | {3, 4, 1, 3, 4, 8} | $[-2.0, 1.9, 1.8, 1.7, 1.6, 1.5, 1.4, -0.1]$          |
| 7   | $\emptyset$                                | {3, 4, 1, 3, 4, 8} | $[-2.0, 1.9, 1.8, 1.7, 1.6, 1.5, 1.4, -0.1]$          |

Using  $L = \{3, 4, 1, 3, 4, 8\}$ , we compute  $\hat{\mathbf{x}}_i = S_{\mathbf{r}_3}S_{\mathbf{r}_4}S_{\mathbf{r}_1}S_{\mathbf{r}_3}S_{\mathbf{r}_4}S_{\mathbf{r}_8}\mathbf{x}_0 = \frac{-1}{8}[4, 2, 4, 4, 2, 2, 0, 2]$ , which is the nearest codeword to  $\mathbf{y}$ .

## Acknowledgment

The author gratefully acknowledges the many extensive discussions with Jim Massey who introduced him to information theory and coding. This very personal introduction to coding was an excellent apprenticeship how to apply mathematics in these fields, while keeping the respect for both the theory as well as for the applied problems. In particular, this work was encouraged by Jim Massey's support to apply the theory of Coxeter groups to group codes.

## References

- [1] D. Slepian, 'Permutation modulation,' *Proc. of the IEEE*, vol. 53, no. 3, pp. 228-236, 1965.
- [2] D. Slepian, 'Group codes for the Gaussian channel,' *Bell Syst. Tech. J.*, vol. 47, pp. 575-602, 1968.

- [3] I. Ingemarsson, ‘Group codes for the Gaussian channel,’ in *Topics in Coding Theory*, LN in Contr. and Inform. Sciences, vol. 128, pp.73-108, Springer 1989.
- [4] I. Ingemarsson, ‘Optimized permutation modulation,’ *IEEE Trans. Inform. Theory*, Vol. 36, No. 5, pp. 1098 – 1100, 1990.
- [5] L.C. Grove and C.T. Benson, *Finite Reflection Groups*, GTM 99, Springer, New York, 1985.
- [6] J.E. Humphreys, *Reflection Groups and Coxeter Groups*, Cambridge studies in advanced mathematics, Vol. 29, Cambridge Univ. Press, 1990.
- [7] J.L. Camenisch, ‘Group Codes from Crystallographic Reflection Groups,’ MSEE thesis, Lab. Signal- and Information Processing, ETH-Zurich, Winter 1992/93.
- [8] T. Mittelholzer and J. Lahtonen, ‘On group codes generated by finite reflection groups,’ (manuscript in preparation).
- [9] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, Grundlehren der math. Wissenschaften Vol. 290, Springer, 1988.

# Duality of Linear Input-Output Maps

Sanjoy K. Mitter \*

Department of Electrical Engineering and Computer Science  
and  
Laboratory for Information and Decision Systems  
Massachusetts Institute of Technology  
Cambridge, MA 02139 USA

Dedicated to James L. Massey on the occasion of his 60th birthday.

## Abstract

This paper is concerned with the duality of linear input-output maps and makes precise in what sense the dual of a linear input-output map gives rise to a dual linear state-space system and how reachability and observability of the original system corresponds to observability and reachability of the dual system.

## I Introduction

The objective of this technical note is to close a gap in the module theory of stationary finite-dimensional linear systems as developed by R.E. Kalman. This is concerned with the duality of linear input-output maps and makes precise in what sense the dual of a linear input-output map gives rise to a dual linear state-space system and how reachability and observability of the original system corresponds to observability and reachability of the dual system.

It is appropriate that this note be dedicated to Jim Massey. He more than anybody else has investigated the deep connections that exist between systems theory and coding (decoding) theory. We see ample demonstration of this in his early work on the Berlekamp-Massey algorithm and its connections to Partial Realization Theory, his joint work with Sain on inverses of linear sequential systems and in his recent joint work with his students Loeliger and Mittelholzer on the relationship between the behavioural theory of linear systems and coding theory.

On a more personal note, I have admired and continue to admire Jim's devotion to the science of engineering, his constant search for clarity of thought and exposition and his intellectual integrity. In a scientific and technological world over-crowded with opportunists and charlatans the values that Jim stands for have provided strength and sustenance to his friends and undoubtedly to his students.

---

\*This research has been supported by the US Army Research Office under grant DAAL03-92-G-0115 through the Center for Intelligent Control Systems

## II Problem Formulation and Preliminaries

We shall follow the notation and terminology of R.E. Kalman (see KALMAN-FALB-ARBIB [2]).

Consider the linear stationary finite-dimensional system:

$$(\Sigma) \begin{cases} x(t+1) = Fx(t) + Gu(t); & u(-\infty) = 0 \\ y(t) = Hx(t) \end{cases}$$

where  $U$ ,  $X$  and  $Y$  are finite-dimensional topological  $K$ -vector spaces and  $F : X \rightarrow X$ ,  $G : U \rightarrow X$  and  $H : X \rightarrow Y$  are  $K$ -linear continuous maps.

Let

$$U[z] = \{u = u_{-k}z^k + u_{-k+1}z^{k-1} + \cdots + u_0 \mid u_i \in U, i = -k, -k+1, \dots, 0\}$$

and

$$Y[[z^{-1}]] = \{y = y_1z^{-1} + y_2z^{-2} + \cdots \mid y_i \in Y, i = 1, 2, \dots\}$$

The input-output map corresponding to  $\Sigma$  is defined by solving the equation defining  $(\Sigma)$  recursively:

$$\begin{aligned} f_\Sigma &: U[z] \rightarrow Y[[z^{-1}]] \\ &: u \mapsto f_\Sigma(u) \end{aligned} \tag{2.1}$$

The map  $f_\Sigma$  is a Hankel map which is a module homomorphism from the  $K[z]$ -module  $U[z]$  to the  $K[z]$ -module  $Y[[z^{-1}]]$ . Now suppose that  $U[z]$  and  $Y[[z^{-1}]]$  can be made into topological  $K$ -vector spaces. Let  $(U[z])^*$  and  $(Y[[z^{-1}]]^*)$  be their topological duals.

Define the dual map

$$f_\Sigma^* : (Y[[z^{-1}]]^*)^* \longrightarrow (U[z])^*$$

as follows:

Let  $Q : Y[[z^{-1}]] \rightarrow K$  be  $K$ -linear and continuous. Then

$$f_\Sigma^* \circ Q = Q \circ f_\Sigma$$

Now it is not a-priori clear that  $f_\Sigma^*$  is an input-output map, that is, maps polynomials into formal power series. We wish to show in this note that appropriate topologies can be put on  $Y[[z^{-1}]]$  and  $U[z]$  such that there exists  $K[z^{-1}]$ -module isomorphisms  $\varphi$  and  $\psi$  such that the following diagram commutes:

$$\begin{array}{ccc} (Y[[z^{-1}]]^*)^* & \xrightarrow{f_\Sigma^*} & (U[z])^* \\ \varphi \downarrow & & \downarrow \psi \\ Y^*[z^{-1}] & \xrightarrow{f_\Sigma} & U^*[z]. \end{array}$$

with  $f_\Sigma^*$  and  $f_\Sigma$  being  $K[z^{-1}]$ -module homomorphisms. In the above  $Y^*$  and  $U^*$  are dual spaces of  $Y$  and  $U$  and  $f_\Sigma$  is the input-output map of the dual system:

$$(\bar{\Sigma}) \begin{cases} \xi(t-1) = F^*\xi(t) + H^*\eta(t); \xi(+\infty) = 0 \\ \chi(t) = G^*\xi(t), \text{ where} \end{cases}$$

$F^*$ ,  $H^*$ ,  $G^*$  are the dual linear maps corresponding to  $F$ ,  $G$ ,  $H$  defined in the natural way. We shall treat the following two cases:

- (i)  $K$  is either  $R$  or  $C$  and  $U$  and  $Y$  are finite-dimensional  $K$ -vector spaces with the euclidean topology.
- (ii)  $K$  is a finite field with the discrete topology and  $U$  and  $Y$  are finite-dimensional  $K$ -vector spaces with discrete topologies.

### III Main Results

We first look at Case (i). The ideas that we use are well-known in Functional Analysis (cf. TREVES [3], pp. 227–231). For the sake of completeness we describe this here.

Let  $\mathcal{P}$  denote the vector space of all polynomials in one indeterminate with complex coefficients (the real case is similar). Let  $\mathcal{P}_k$  denote the vector subspace of polynomials with degree  $\leq k$ ,  $k = 0, 1, 2, \dots$ . Each  $\mathcal{P}_k$  is finite-dimensional. We provide  $\mathcal{P}$  with the locally convex topology which is the inductive limit of the topologies of the Hausdorff finite-dimensional space (we can choose this to be the euclidian topology)  $\mathcal{P}_k$ ,  $k = 0, 1, \dots$  (cf. TREVES [3], p. 130).  $\mathcal{P}$  with this topology is a so-called *LF*-space.

On the other hand, let  $\mathcal{F}$  denote the vector space of formal power series in one-indeterminate. We put on  $\mathcal{F}$  the topology of convergence of each coefficient.

This topology can be defined by the sequence of seminorms:

$$u = \sum_{p \in \mathbb{N}} u_p z^p \rightsquigarrow \sup_{p \leq k} |u_p|, k = 0, 1, 2, \dots$$

This topology converts  $\mathcal{F}$  into a Fréchet space (cf. TREVES [3], p. 91).

There is a natural duality between polynomials and formal power series which can be expressed by the bracket

$$\langle P, u \rangle = \sum_{p \in \mathbb{N}} P_p u_p, \text{ where}$$

$$P = \sum_p P_p z^p \text{ and } u = \sum_p u_p z^p.$$

This is well-defined since all coefficients  $P_p$ , except possibly a finite number of them are equal to zero. We then have:

**Theorem 3.1** (TREVES [3], p. 228, Th. 22.1).

- (a) The map.  $u \rightsquigarrow (P \rightsquigarrow \langle P, u \rangle)$  is an isomorphism for the structures of topological vector spaces of the Fréchet space of formal power series  $\mathcal{F}$  onto the strong dual of the *LF*-space  $\mathcal{P}$  of polynomials.
- (b) The map.  $P \rightsquigarrow (u \rightsquigarrow \langle P, u \rangle)$  is an isomorphism of  $\mathcal{P}$  onto the strong dual of  $\mathcal{F}$ .

**Remark 3.1** If we disregard the multiplicative structure of  $\mathcal{P}$  and  $\mathcal{F}$  and we write  $u = (u_p)_{p \in \mathbb{N}}$ , then  $\mathcal{F}$  is the space of complex functions on  $\mathbb{N}$  and  $\mathcal{P}$  is the space of functions on  $\mathbb{N}$  which vanish outside a finite set.

Then

$$\mathcal{F} = \prod_{p \in \mathbb{N}} C_p \quad C_p \simeq C$$

and  $\mathcal{P}$  can be regarded as a direct sum of the  $C_p$ 's.

In fact, the topology of simple convergence of the coefficients of  $u \in \mathcal{F}$  is precisely the product topology of the  $C_p$ 's. Furthermore  $\mathcal{P}$  is continuously embedded in  $\mathcal{F}$  and is dense in  $\mathcal{F}$ .

We can apply the above ideas to our problem. Firstly, we can make  $Y[[z^{-1}]]$  into a topological vector space by assigning to each  $\underline{y} \in Y[[z^{-1}]]$  a basis of a filter.

Let  $\underline{y} = \sum_{i=1}^{\infty} y_i z^i$ , and define

$$Q_{m,n} = \{y \in Y[[z^{-1}]] \mid \forall i \leq n, |y_i| \leq \frac{1}{m}\}, n = 1, 2, \dots, m = 1, 2, \dots$$

It is easily checked that  $(Q_{m,n})$  is a basis for a filter on  $\underline{0}$ . Define now  $\{\underline{y} + Q_{m,n}\}$  to be a basis of a filter for any  $\underline{y} \in Y[[z^{-1}]]$ . We can now easily show that this filter assignment defines a topology on  $Y[[z^{-1}]]$ . We now claim that with this topology  $Y[[z^{-1}]]$  is a topological vector space. For this purpose we need to check the following:

- (i)  $f_{y_0} : Y[[z^{-1}]] \rightarrow Y[[z^{-1}]] : y \mapsto y_0 + y$  is a homeomorphism, i.e., the topology is translation invariant.
- (ii) If  $V \in (Q_{m,n})$ ,  $\exists U \in (Q_{m,n})$  such that  $U + U \subset V$ .
- (iii)  $V \in (Q_{m,n})$  is absorbing.
- (iv)  $\exists \lambda \in \mathbf{C}, 0 < \lambda < 1$ , such that  $\lambda Q_{m,n} \in (Q_{m,n})$ .

The above steps are easily carried out.

We now topologize  $U[z]$ . Let  $U_m[z] = \{u \in U[z] \mid \deg(u) \leq m\}$ . Clearly  $U_m[z]$  is a finite-dimensional vector space. Endow  $U^m[z]$  with the topology given by the norm

$$\|u\|_m = (|u_0|^2 + |u_{-1}|^2 + \dots + |u_{-m}|^2)^{1/2}.$$

Now  $U[z] = \bigcup_m U_m[z]$ . Endow  $U[z]$  with the inductive limit of the Hausdorff topology on the finite-dimensional spaces  $U_m[z]$ . This makes  $U[z]$  into a topological vector space.

There is a natural  $K[z^{-1}]$ -module structure on  $Y^*[z^{-1}]$  and  $U^*[[z]]$ . Define multiplication of elements of  $Y^*[z^{-1}]$  by a polynomial as follows:

For

$$a(z^{-1}) = \sum_{i=0}^n a_i z^{-i}, a_i \in \mathbf{C}, a_i = 0, i > n, f = \sum_{j=0}^m f_j z^{-j}, f_j \in Y^*$$

$$a(z^{-1}) \cdot f = \sum_{\ell=0}^{n+m} g_\ell z^{-\ell} = g$$

$$\text{where } g_\ell = \sum_{k=0}^{\ell} a_{\ell-k} f_k.$$

This multiplication is well-defined and  $g \in Y^*[z^{-1}]$ . The module axioms are easily checked. As far as  $U^*[[z]]$  is concerned, define multiplication of elements of  $U^*[[z]]$  by a polynomial as follows:

For  $a(z^{-1}) = \sum_{i=0}^n a_i z^{-i}$ ,  $a_i \in \mathbf{C}$ , and  $\underline{f} = \sum_{j=1}^{\infty} f_{-j} z^j$  with  $a_i = 0$  for all other  $i \in Z$  and  $f_j = 0$ , for all other  $j \in Z$ ,

$$a(z^{-1}) \cdot \underline{f} = \sum_{\ell=1}^{\infty} g_{-\ell} z^{\ell} = g$$

where  $g_{-\ell} = \sum_{k=0}^{\ell+n} a_{-\ell+k} f_{-k}$ ,  $\ell \geq 1$  and  $g_{-\ell} = 0$ , for  $\ell < 1$ . Again the module axioms are easily verified. Let  $(Y[[z^{-1}]])^*$  and  $(U[z])^*$  denote the strong duals of  $Y[[z^{-1}]]$  and  $U[z]$  respectively. For the definition of strong dual, see TREVES [3], p. 198.

Define the pairings

(i)

$$\begin{aligned} <\cdot, \cdot>_1 : Y^*[z^{-1}] \times Y[[z^{-1}]] &\longrightarrow K \\ &: (\underline{f}, \underline{y}) \longmapsto \sum_{i=0}^n f_i y_{i+1}, \end{aligned}$$

where

$$f = \sum_{i=0}^n f_i z^{-i} \text{ and } \underline{y} = \sum_{i=1}^{\infty} y_i z^{-i}$$

(ii)

$$\begin{aligned} <\cdot, \cdot>_2 : U^*[z] \times U[z] &\longrightarrow K \\ &: (\underline{f}, u) \longmapsto \sum_{i=1}^{\infty} f_{-i} u_{-i+1} \end{aligned}$$

where

$$\underline{f} = \sum_{i=1}^{\infty} f_{-i} z^i, u = \sum_{i=0}^n u_{-i} z^i.$$

**Theorem 3.2** (a) *The map*

$$\begin{aligned} \phi : Y^*[z^{-1}] &\longrightarrow (Y[[z^{-1}])^* \\ \underline{f} &\rightsquigarrow (\underline{y} \rightsquigarrow <\underline{f}, \underline{y}>_1) \end{aligned}$$

is a  $K[z^{-1}]$ -module isomorphism from  $Y^*[z^{-1}]$  to the strong dual of  $Y[[z^{-1}]]$ .

The map  $\phi$  is an isomorphism for the structures of topological vector spaces on  $Y^*[z^{-1}]$  and  $Y[[z^{-1}]]^*$

(b) *The map*

$$\begin{aligned} \psi : U^*[z] &\longrightarrow (U[z])^* \\ f &\rightsquigarrow (u \rightsquigarrow <\underline{f}, u>_2) \end{aligned}$$

is a  $K[z^{-1}]$ -module isomorphism. It is also an isomorphism for the structures of topological vector spaces on  $U^*[z]$  and  $(U[z])^*$ .

**Proof.** We make  $(Y[[z^{-1}]])^*$  into a  $K[z^{-1}]$  module in the following way:

For  $f \in Y^*[z^{-1}]$ , let  $\varphi(f)$  be denoted as  $Q_f$  which belongs to  $Y[[z^{-1}]]^*$ . Define multiplication of  $Q_f$  by  $a(z^{-1}) \in K[z^{-1}]$  as

$$a(z^{-1}) \circ Q_f = Q_g \text{ where } g = a(z^{-1}) \circ f$$

Now we can easily check that  $\varphi$  is  $K$ -linear and the module axioms are satisfied. Using the  $K$ -linearity and the above definition of multiplication it is easily checked that  $\varphi$  is  $K[z^{-1}]$ -linear.

In a similar way we make  $(U[z])^*$  into a  $K[z^{-1}]$ -module. The proof is now carried out by proving  $\varphi$  and  $\psi$  are continuous, injective, and surjective. This follows the proof of Theorem 3.1 as given by Treves.

As far as  $\varphi$  and  $\psi$  being isomorphisms of the topological vector space structures, the proof is carried by proving that  $\varphi^{-1}$  and  $\psi^{-1}$  are continuous as in Treves' proof of Theorem 3.1. ■

**Theorem 3.3** *There exists module isomorphism  $\varphi$  and  $\psi$  as in the previous theorem such that the following diagram commutes:*

$$\begin{array}{ccc} (Y[[z^{-1}]])^* & \xrightarrow{f_\Sigma^*} & (U[z])^* \\ \uparrow \varphi & & \uparrow \psi \\ Y^*[z^{-1}] & \xrightarrow{f_\Sigma} & U^*[z] \end{array}$$

where  $f_\Sigma^*$  is defined as,  $f_\Sigma^* \circ Q = Q \circ f_\Sigma$  and  $Q : Y[[z^{-1}]] \rightarrow K$  is  $K$ -linear,  $f_\Sigma^*$  is  $K[z^{-1}]$ -linear, and

$$f_\Sigma : f_n z^{-n} + f_{n-1} z^{-n+1} + \cdots + f_0 \mapsto g_{-1} z^1 + g_{-2} z^2 + \cdots$$

with  $f_i \in Y^*$  and  $g_{-i} = G^* F^{*i-1} H^* f_0 + \cdots + G^* F^{*n+i-1} H^* f_n$ .

**Proof.** The proof is constructed by showing that

$$f_\Sigma^*[\varphi(f)](u) = \psi[f_\Sigma(f)](u), u \in U[z] \quad (3.1)$$

where  $f = f_n z^n + f_{n-1} z^{-n+1} + \cdots + f_0, f_i \in Y^*, i = 0, \dots, n$ .

Now  $f_\Sigma^*[\varphi(f)](u) = \phi(f)[f_\Sigma(u)]$ .

By solving the recurrence relation corresponding to  $(\Sigma)$ , we get

$$\begin{aligned} \varphi(f)[f_\Sigma(u)] &= f_0(HG u_0) + f_1(HFG u_0) + \cdots + f_n(HF^n G u_0) \\ &\quad + f_0(HFG u_{-1}) + f_1(HF^2 G u_{-1}) + \cdots + f_n(HF^{n+1} G u_{-1}) \\ &\quad + \cdots \\ &\quad + f_0(HF^K G u_{-k}) + f_1(HF^{k+1} G u_{-k}) + \cdots + f_n(HF^{n+k} G u_{-k}) \\ &= (G^* H^* f_0)(u_0) + (G^* F^* H^* f_1)(u_0) + \cdots + (G^* F^{*n} H^* f_n)(u_0) \\ &\quad + (G^* F^* H^* f_0)(u_{-1}) + (G^* F^{*2} H^* f_1)(u_{-1}) \\ &\quad + \cdots + (G^* F^{*n+1} H^* f_n)(u_1) + \cdots \\ &\quad + (G^* F^{*k} H^* f_0)(u_{-k}) + (G^* F^{k+1} H^* f_1)(u_{-1}) + \cdots + \\ &\quad \cdots (G^* F^{*n+k} H^* f_n)(u_{-k}) \end{aligned} \quad (3.2)$$

On the other hand if,

$$\begin{aligned}\underline{g} &= g_{-1}z^1 + g_{-2}z^2 + \cdots, \quad g_{-i} \in U^* \\ \psi(\underline{g})(u) &= \sum_{i=1}^{\infty} g_{-i}(u_{-i+1})\end{aligned}\tag{3.3}$$

By solving the recurrence relation corresponding to  $(\bar{\Sigma})$ , we get for  $\underline{g} = f_{\bar{\Sigma}}(f)$ , that

$$\begin{aligned}g_{-1} &= G^* H^* f_0 + \cdots + (G^* F^{*n} H f_n) \\ &\vdots \\ g_{-i} &= G^* F^{*i-1} H^* f_0 + \cdots + (G^* F^{*n+i-1} H^* f_n) \\ &\vdots\end{aligned}$$

Therefore  $\phi[f_{\bar{\Sigma}}(f)](u)$  is precisely the right side of (3.1). ■

The Case where  $K$  is a Finite Field. Let  $K$  be a finite field with the discrete topology and  $U$  and  $Y$  finite-dimensional  $K$ -vector spaces with the discrete topology. Let the topologies on  $K$  and  $U, Y$  be generated by a norm  $|\cdot|_K, |\cdot|_U, |\cdot|_Y$  where  $|v| = 0 \Leftrightarrow v = 0$  and  $|v| = 1$  if and only if  $v \neq 0$ . Here  $v \in K, U$  or  $Y$ .

Let us define

$$M^n = \{\underline{y} \in Y[[z^{-1}]] | y_1 = y_2 = \cdots = y_{n-1} = 0\}, n = 1, 2, \dots$$

The family of sets  $(M^n)$  is the same as the family of set  $Q_{m,n}$  defined earlier.

In the same way as before  $(\underline{y} + M^n)$  is the assignment of a filter based on  $\underline{y}$  and generates a topology. We can now check that addition and multiplication are continuous and hence  $Y[[z^{-1}]]$  becomes a topological  $K$ -vector space.

On  $U[z]$  we put the discrete topology. With these choices, we can proceed in the same way as before and prove Theorem 3.3 in the case where  $K$  is a finite field.

## IV Final Remarks

We may proceed using realization theory instead of starting with an explicit state-space realization. Thus given an input-output map  $f_{\Sigma} : U[z] \rightarrow Y[[z^{-1}]]$  obtain a minimal (reachable and observable) realization via the canonical factorization

$$\begin{array}{ccc} U[z] & \xrightarrow{f_{\Sigma}} & Y[[z^{-1}]] \\ \mathcal{R} \searrow & \nearrow 0 & \\ U[z]/\ker f_{\Sigma}! & = & X \end{array}$$

where the reachability operator  $\mathcal{R}$  and the observability operator  $0$  are defined by

$$\begin{aligned}\mathcal{R} &: U[z] \rightarrow X : u \mapsto [u] \quad ([\cdot] \text{ denotes equivalence}) \\ O &: X \rightarrow Y[[z^{-1}]] : [u] \mapsto f_{\Sigma}(u).\end{aligned}$$

Let  $F : X \rightarrow X$ ,  $G : U \rightarrow X$  and  $H : X \rightarrow Y$  be  $K$ -linear maps defining the corresponding minimal state space realization. Now define

$$\begin{aligned} f_{\bar{\Sigma}} &: Y^*[z^{-1}] \longrightarrow U[[z]] \text{ by} \\ f_{\bar{\Sigma}}^* \circ \varphi &= \phi \circ f_{\bar{\Sigma}}. \end{aligned}$$

Then as in Theorem 3.3, we can check that the state-space system defined by

$$\begin{aligned} F^* &: X^* \rightarrow X^* \\ H^* &: Y^* \rightarrow X^* \\ G^* &: X^* \rightarrow U^* \end{aligned}$$

realizes the map  $f_{\bar{\Sigma}}$  (note the time-reversal). We can explicitly compute the reachability operator  $\bar{\mathcal{R}}$  and the observability operator  $\bar{\mathcal{O}}$  corresponding to  $f_{\bar{\Sigma}}$ . We have that  $(\Sigma)$  is reachable if and only if  $(\bar{\Sigma})$  is observable and  $(\Sigma)$  is observable if and only if  $(\bar{\Sigma})$  is reachable.

These ideas have relevance towards the development of a theory of duality for linear systems defined in a behavioural framework (cf. WILLEMS [4]). To make connection with the input-output setting we need to work with controllable systems in the sense of Willems. These ideas also have relevance on the duality theory of Abelian Group Codes and Systems. See FORNEY-TROTT[1].

## Acknowledgement

I would like to acknowledge discussions I had about the subject of this paper with T. E. Djaferis of the University of Massachusetts, Amherst, MA when he was my doctoral student at M.I.T. in the mid-seventies.

## References

- [1] G.D. Forney Jr. and M. Trott , Private Communication. See also G.D. Forney, Duals of Abelian Group Codes and Systems, Manuscript, 1993.
- [2] R.E. Kalman, P.L. Falb, and M.A. Arbib , *Topics in Mathematical System Theory*, McGraw Hill, New York, 1969.
- [3] F. Treves , *Topological Vector Spaces, Distributions and Kernels*, Academic Press, New York, 1967.
- [4] J.C. WILLEMS, “Models for Dynamics” in *Dynamics Reported*, Vol. 2, U. Kirchgraber and H.O. Walther, ed., Wiley, New York, 1989.

# Cut-off Rate Channel Design

Prakash Narayan  
University of Maryland,  
College Park, MD 20742

Donald L. Snyder  
Washington University,  
St. Louis, MO 63130

## Abstract

The eloquence with which Massey advocated the use of the cut-off rate parameter for the coordinated design of modulation and coding in communication systems caused many to redirect their thinking about how communication systems should be designed. Underlying his recommendation is the view that modulation and demodulation should be designed to realize a good discrete channel for encoding and decoding, rather than the prevailing view at the time, and still the view of many, that bit error-probability should be optimized. In this short paper, some of the research influenced by Massey's insightful suggestions on this subject is reviewed.

## I Introduction

The use of the cut-off rate parameter  $R_0$  in the study of single-user coded communication systems was first advocated by Wozencraft and Kennedy [11] in 1966. Unfortunately, their proposal to use  $R_0$  as a criterion for the design of the modulation system remained largely unheeded. Then, in a remarkable paper in 1974, Massey [4] gave an eloquent argument in favor of the cut-off rate parameter as a criterion for the coordinated design of modulation and coding in a communication system. Calling to discard the heretofore popular "error probability" criterion on the grounds that it was apposite only for uncoded systems, he resurrected the earlier proposal of Wozencraft and Kennedy and showed that the  $R_0$  criterion led to a rich "communication theory" of its own for coded communications. In particular, by interpreting  $R_0$  as a function of the modulator and demodulator, Massey [4] demonstrated how it could be used to design the best discrete channel as seen by the encoder and decoder. He crowned his arguments by establishing that a simplex signal set maximized the cut-off rate of an infinite-bandwidth, additive, white Gaussian-noise channel for infinitely soft decisions. The "optimality" of such a signal set with respect to the error probability criterion has remained a conjecture for many years.

Massey's paper [4] opened the floodgates for a plethora of publications employing the cut-off rate criterion to assess the performance of coding and modulation schemes for a variety of applications ranging from optical communications to spread-spectrum systems to an extension to multiaccess channels. A fair citation of this field is beyond the scope of this paper; good sources of relevant publications are the *IEEE Transactions on Information Theory and Communications*.

A mention of the widespread use of the cut-off rate as a performance criterion must be qualified by the observation that it has not been accorded universal acceptance. A key reason is its seeming lack of fundamental significance, unlike that of the *capacity* of a communication channel. Another reason is the existence of channels with memory for which the cut-off rate, unlike capacity, exhibits anomalous behavior (see, e.g., [5]).

This paper focuses on the role of the cut-off rate criterion in the coordinated design of coding and modulation formats for the single-user, additive white Gaussian-noise channel [4]. We briefly review the work of Massey [4] when the modulated signals possess unlimited bandwidths. This problem is then considered in the presence of constraints on signal bandwidth; a complete solution remains elusive. Partial results by Narayan and Snyder [6] are then presented.

## II The Cut-Off Rate and Its Properties

Let  $\{W : \mathcal{X} \rightarrow \mathcal{Y}\}$  be a discrete memoryless channel (DMC) with finite input and output alphabets,  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively. Consider a (random) code of rate  $R > 0$  with codewords of blocklength  $n$ , and a maximum-likelihood decoder. It is assumed that the codewords are independent and identically distributed (i.i.d.) and that the symbols in each codeword are i.i.d. with (an arbitrary) probability mass function  $P$  on  $\mathcal{X}$ . The average probability of a decoding error when this code is used on the DMC  $\{W\}$  is bounded above [3, pp. 142-143] according to

$$P_e(P, W) \leq \exp[-n(E_0(\rho, P, W) - \rho R)], \quad 0 \leq \rho \leq 1 \quad (1)$$

where

$$E_0(\rho, P, W) = -\log \left[ \sum_{y \in \mathcal{Y}} \left( \sum_{x \in \mathcal{X}} P(x) W^{1/(1+\rho)}(y|x) \right)^{1+\rho} \right]. \quad (2)$$

(All logarithms and exponentiations are with respect to the base 2.)

The upper bound in (2) is improved by choosing  $P$  and  $\rho$  so as to maximize  $E_0(\rho, P, W) - \rho R$ . To this end, consider

$$\max_P \max_{0 \leq \rho \leq 1} E_0(\rho, P, W) - \rho R \quad (3)$$

and note from [3] that for

$$R \leq R_{crit}(P, W) = \frac{\partial E_0}{\partial \rho}(\rho, P, W) |_{\rho=1}, \quad (4)$$

$E_0(\rho, P, W)$  is maximized by  $\rho = 1$ .

The quantity  $\max_P E_0(1, P, W)$  is called the *cut-off rate* of the DMC  $\{W\}$ , denoted  $R_0(W)$ , and is given by

$$R_0(W) = \max_P \left[ -\log \sum_{y \in \mathcal{Y}} \left( \sum_{x \in \mathcal{X}} P(x) W^{1/2}(y|x) \right)^2 \right]. \quad (5)$$

Observe that  $R_0(W)$  depends on  $\{W\}$  but not on  $P$ . If  $P^*$  achieves the maximum in (5), we obtain from (2) that

$$P_e(P^*, W) \leq \exp[-n(R_0(W) - R)]. \quad (6)$$

This simpler bound on average error probability is useful if  $R < R_0(W)$ , and is quite accurate for  $R \cong R_{crit}(P^*, W)$ . Massey [4] concluded from (6) that for block codes with maximum-likelihood decoding,  $R_0(W)$  determines a range of code rates at which reliable communication can be assured, as well as the coding complexity, reflected by  $n$ , to achieve a specified level of reliability. (For a different interpretation of cut-off rate for channels with memory, see [5].)

The cut-off rate of a DMC  $\{W\}$  also affords other interpretations. For instance, as pointed out by Csiszár [2],  $R_0(W)$  equals Rényi capacity [7] or “information radius” of order  $\alpha = 1/2$ , and the  $\beta$ -cut-off rate [2] for  $\beta = -1$ .

The cut-off rate parameter plays an important role in assessing the performance of codes different from that considered above. For instance, Viterbi [10] has shown for convolutional coding with maximum-likelihood decoding that the average probability of decoding error on a DMC  $\{W\}$  is bounded above according to

$$P_e(P^*, W) \leq k_R \exp[-nR_0(W)] \quad (7)$$

where  $n$  is the constraint length, and  $k_R$  varies slowly with  $R$ . Also, the cut-off rate is a key parameter in sequential decoding, wherein the receiver decodes a code with a tree structure by computing the metrics of, and making tentative hypotheses on, successive branches of the tree and by changing these hypotheses when subsequent choices indicate an earlier incorrect hypothesis. The cut-off rate  $R_0(W)$  of a DMC  $\{W\}$  is the  $R_{comp}(W)$  for sequential decoding (cf. Arikan [1]), which is the rate above which the average computational complexity of the decoding algorithm becomes unbounded.

### III The Additive White Gaussian Noise Channel

Let  $\mathcal{X} = \{1, \dots, a\}$  be the (finite) encoder alphabet. The corresponding (modulated) signal set is  $\mathcal{S} = \{s_i(t), 0 \leq t \leq T; i = 1, \dots, a\}$ , where  $s_i(t)$  is the signal transmitted by the sender over the waveform channel when the encoder produces the symbol  $i$ . Signals are transmitted and received on the interval  $[0, T]$ . The (random) signal  $Z(t)$  received at the output of the additive, white, Gaussian-noise (AWGN) waveform-channel is

$$Z(t) = s_i(t) + N(t), \quad (8)$$

where  $N(t)$  is white Gaussian-noise with power-spectral density  $N_0/2$  W/Hz. The demodulator then produces an output from the alphabet  $\mathcal{Y} = \{1, \dots, b\}$ .

From a coding viewpoint, the modulator, waveform channel, and demodulator together constitute a discrete channel with input alphabet  $\mathcal{X}$  and output alphabet  $\mathcal{Y}$ . By virtue of the stationarity and independent increments property of  $\{N(t), 0 \leq t \leq T\}$ , this channel is also memoryless, and hence a DMC; we denote this DMC by  $\{W\}$ . It is important to note that  $\{W\}$  depends on the choice of the signal set  $\mathcal{S}$ , although this dependence will not be displayed for notational convenience.

The cut-off rate of the DMC  $\{W\}$  is not decreased by using a finer output quantization at the demodulator. In this treatment, we shall restrict ourselves to the limiting situation when the output quantization is arbitrarily fine, i.e.,  $b = \infty$ . For the effects of finite quantization, see [4]. It then follows from [4] and (2) that

$$E_0(1, P, W) = -\log \sum_{i,j=1}^a P(i)P(j) \exp[-s_{ij}/4N_0] \quad (9)$$

where

$$s_{ij} = \int_0^T [s_i(t) - s_j(t)]^2 dt. \quad (10)$$

The problem of coordinated design of the encoder and modulator, using the cut-off rate criterion, can now be stated as follows.

*Problem 1:* Determine

$$\max_S R_0(W)$$

or, equivalently,

$$\max_S \max_P E_0(1, P, W) \quad (11)$$

subject to

$$P(i) \geq 0, \quad i = 1, \dots, a; \quad \sum_{i=1}^a P(i) = 1; \quad (12)$$

and the “average energy” constraint

$$\sum_{i=1}^a P(i) \left( \int_0^T s_i^2(t) dt \right) = E. \quad (13)$$

**Theorem 1 (Massey [4]):** The maximum in Problem 1 is attained by the *simplex* signal set  $S^*$  characterized by

$$\frac{1}{a} \sum_{i=1}^a s_i^*(t) = 0, \quad 0 \leq t \leq T; \quad (14)$$

$$s_{ij}^* = s, \quad i \neq j; \quad (15)$$

$$\int_0^T s_i^{*2}(t) dt = E, \quad i = 1, \dots, a; \quad (16)$$

where  $s$  is a constant for distinct signals and with the code symbols being chosen according to the uniform probability distribution

$$P^*(i) = 1/a, \quad i = 1, \dots, a. \quad (17)$$

The corresponding maximal cut-off rate is

$$R_0^* = \log a - \log \left[ 1 + (a-1) \exp \left( -\frac{aE}{2(a-1)N_0} \right) \right]. \quad (18)$$

**Proof:** See [4].

*Remark:* It is interesting to note that the conjectured optimality of the simplex set with respect to the probability of error criterion, subject to (13) with  $P(i) = 1/a$ ,  $1 \leq i \leq a$  (the “strong” simplex conjecture) was recently disproved [9]. The “weak” simplex conjecture is still unresolved when (13) is replaced by the constraint  $\int_0^T s_i^2(t) dt = E$ ,  $1 \leq i \leq a$ .

Either by virtue of law or nature, it is generally necessary to impose constraints on the portion of the frequency spectrum that the signals transmitted by a sender can occupy. There is no universal measure of bandwidth for a signal of finite duration; several measures have been proposed, and we consider two of these below.

Consider the signal set  $\mathcal{S} = \{s_i(t), 0 \leq t \leq T, i = 1, \dots, a\}$  used according to the probability distribution  $\{P(i), i = 1, \dots, a\}$ . The *squared second moment bandwidth* of  $\mathcal{S}$  is defined by

$$B_{SM}(\mathcal{S}; P) = \frac{\sum_{i=1}^a P(i) \int_0^T [ds_i(t)/dt]^2 dt}{\sum_{i=1}^a P(i) \int_0^T [s_i(t)]^2 dt}. \quad (19)$$

If  $S_i(f)$ ,  $-\infty < f < \infty$ , denotes the Fourier transform of  $s_i(t)$ ,  $0 \leq t \leq T$ , the *fractional out-of-band energy* of  $\mathcal{S}$  is the fraction of the total average energy of  $\mathcal{S}$  lying outside a prespecified frequency band  $[-F, F]$  and is defined by

$$B_{OB(F)}(\mathcal{S}; P) = \frac{\sum_{i=1}^a P(i) \left[ \int_{-\infty}^{\infty} |S_i(f)|^2 df - \int_{-F}^F |S_i(f)|^2 df \right]}{\sum_{i=1}^a P(i) \int_{-\infty}^{\infty} |S_i(f)|^2 df}. \quad (20)$$

We now state two problems of coordinated encoder and signal design subject to constraints on the bandwidth of the signal set, using the cut-off rate criterion. These problems are obvious extensions of Problem 1 above, and areas yet unresolved in full generality.

*Problem 2A:* Same as Problem 1 above, with the additional squared second-moment bandwidth constraint

$$B_{SM}(\mathcal{S}; P) \leq \beta^2. \quad (21)$$

*Problem 2B:* Same as Problem 1 above, with the additional fractional out-of-band energy constraint

$$B_{OB(F)}(\mathcal{S}; P) \leq \epsilon. \quad (22)$$

*Remark:* Problems 2A and 2B reduce to Problem 1 upon setting  $\beta^2 = \infty$  in (21) and  $\epsilon = 1$  in (22), respectively.

Modified versions of Problems 2A and 2B above have been, in effect, solved in [6] albeit in the context of a multiaccess AWGN channel. Rather than determining signal sets that maximize  $R_0(W)$  under constraints on average energy and bandwidths, we instead seek to identify signal sets with minimal bandwidths, in the sense of (19) and (20), from among those that achieve the maximal cut-off rate  $R_0^*$  in (18).

*Problem 3A:* Consider the family  $\Sigma$  of all simplex signal sets  $\mathcal{S}$  satisfying (14)-(16), with the signals being used equiprobably in accordance with (17), that is,  $P(i) = 1/a, 1 \leq i \leq a$ . Determine

$$\min_{\mathcal{S} \in \Sigma} B_{SM}(\mathcal{S}; P). \quad (23)$$

*Problem 3B:* For the same setup as in Problem 3A above, determine

$$\min_{\mathcal{S} \in \Sigma} B_{OB(F)}(\mathcal{S}; P). \quad (24)$$

*Remark:* Clearly, if the minimal bandwidth in Problem 3A (resp. Problem 3B) satisfies constraint (19) (resp. (20)), then the corresponding (optimal) signal set is optimal for Problem 2A (resp. Problem 2B) too.

The solutions to Problems 3A and 3B are provided by the following

**Theorem 2 (Narayan-Snyder [6]):**

1. The minimum in Problem 3A is attained by the “raised-cosine” simplex set  $\mathcal{S}^{SM} = \{s_i^{SM}(t), 0 \leq t \leq T; 1 \leq i \leq a\}$  given by

$$s_i^{SM}(t) = \sqrt{\frac{aE}{a-1}} \left[ \left( \frac{a-1}{a} \right) \phi_i(t) - \frac{1}{a} \sum_{j=1, j \neq i}^a \phi_j(t) \right]. \quad (25)$$

where

$$\phi_i(t) = \left( \frac{2}{T} \right)^{1/2} \sin \frac{i\pi t}{T}, \quad 1 \leq i \leq a. \quad (26)$$

The corresponding minimal squared second-moment bandwidth equals

$$B_{SM}(\mathcal{S}^{SM}) = \frac{\pi^2}{6T^2} (a+1)(2a+1). \quad (27)$$

2. The minimum in Problem 3B is attained by the “prolate spheroidal wave” simplex set  $\mathcal{S}^{OB(F)} = \{s_i^{OB(F)}(t), 0 \leq t \leq T; 1 \leq i \leq a\}$  given by

$$s_i^{OB(F)}(t) = \sqrt{\frac{aE}{a-1}} \left[ (a-1)a \Psi_i(t) - \frac{1}{a} \sum_{j=1, j \neq i}^a \Psi_j(t) \right], \quad (28)$$

where

$$\Psi_i(t) = \sqrt{\frac{1}{\lambda_{i-1}(2\pi FT)}} \zeta_{i-1}(t), \quad (29)$$

with  $\zeta_{i-1}(t)$  being a prolate spheroidal wave function with eigenvalue  $\lambda_{i-1}(2\pi FT)$ . The corresponding minimal fractional out-of-band energy equals

$$B_{OB(F)} \left( \mathcal{S}^{OB(F)} \right) = a - \sum_{i=0}^{a-1} \lambda_i (2\pi F T). \quad (30)$$

**Proof:** The proof follows from [6, Section III].

## IV Conclusion

In addition to the results on the AWGN channel reviewed in this paper, several authors have gainfully used the cut-off rate parameter as a criterion for signal design. For instance, Snyder and Rhodes [8] have identified modulation formats that maximize the cut-off rate parameter of a single-user, shot-noise limited optical channel with infinite bandwidth under simultaneous constraints on average energy and peak amplitude (see also Wyner [12]). In [6], some of these results are extended to a two-sender multiaccess channel by maximizing the “cut-off rate region” and identifying the optimal signal sets. Also, conditions are established under which this optimality is preserved when constraints are imposed on signal bandwidth. This cumulative body of work bears out Massey’s thesis [4] that the cut-off rate parameter of a DMC leads to a rich communication theory of its own, offering useful insights into the coordinated design of efficient coding and modulation systems. At the same time, we should be careful not to overstate its importance as it lacks the fundamental significance of channel capacity.

## V Acknowledgments

Prakash Narayan considers it an honor to have been the beneficiary of Jim Massey’s expertise as well as sustained help and encouragement, and expresses with pleasure his deep sense of gratitude to Jim.

## References

- [1] E. Arikan, “An upper bound on the cut-off rate of sequential decoding,” *IEEE Trans. Inform. Theory*, vol. 34, no. 1, pp. 55-63, 1988.
- [2] I. Csiszár, “Generalized cut-off rates and Rényi’s information measures,” *Proc. IEEE Int. Symp. Inform. Th.*, Budapest, Hungary, p. 73, June 1991.
- [3] R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley, New York, 1968.
- [4] J. L. Massey, “Coding and modulation in digital communications,” *Proc. Int. Zurich Seminar on Dig. Comm.*, Switzerland, pp. E2(1)-E2(4), March 1974.
- [5] R. J. McEliece and W. E. Stark, “Channels with block interference,” *IEEE Trans. Inform. Theory*, vol. IT-30, no. 1, pp. 44-53, 1984.

- [6] P. Narayan and D. L. Snyder, "Signal set design for band-limited multiple-access channels with soft decision demodulation," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 4, pp. 539-556, 1987.
- [7] A. Rényi, "On measures of entropy and information," *Proc. 4th Berkeley Symp. Math. Statist. Probability*, vol. 1, Univ. Calif. Press, Berkeley, pp. 547-561, 1961.
- [8] D. L. Snyder and I. B. Rhodes, "Some implications of the cut-off rate criterion for coded direct detection optical communication systems," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 3, pp. 327-338, 1980.
- [9] M. Steiner, "New results in signal design for the AWGN channel," *Proc. IEEE Int. Symp. Inform. Th.*, Budapest, Hungary, p. 204, June 1991.
- [10] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 3, pp. 327-338, 1980.
- [11] J. M. Wozencraft and R. S. Kennedy, "Modulation and demodulation for probabilistic coding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 291-297, 1966.
- [12] A. D. Wyner, "Capacity and error exponent for the direct detection photon channel-part I," *IEEE Trans. Inform. Theory*, vol. 34, no. 6, pp. 1449-1461, 1988.

# Construction of Invertible Sequences for Multipath Estimation

A. M. Odlyzko  
AT&T Bell Laboratories  
Murray Hill, New Jersey 07974

*Dedicated to Jim Massey on the occasion of his 60th birthday*

## ABSTRACT

J. Ruprecht has proposed coding schemes that allow for multipath estimation. His schemes use sequences  $a_0, \dots, a_n$  with  $a_j = \pm 1$  for each  $j$  such that the associated polynomial  $f(z) = \sum a_j z^j$  has a large

$$R_p(f) = \frac{n+1}{\sum_{k=0}^n |f(e^{2\pi i k/(n+1)})|^{-2}}.$$

Most sequences have a small  $R_p(f)$ , and those with maximal  $R_p(f)$  are hard to find. This note shows for  $n$  of the form  $n = q-1$ ,  $q$  a prime, one can construct sequences with  $R_p(f) \geq n - O(n^{1/3})$ . Since  $R_p(f) \leq n+1$  for any sequence, this construction is asymptotically close to optimal. It also produces large values of  $R_p(f)$  for small  $n$ .

It is also shown that for  $n = q-1$ ,  $q$  a prime, there exist sequences  $a_0, \dots, a_n$  such that the associated polynomial  $f(z)$  satisfies

$$|f(e^{2\pi i k/(n+1)})| = (1 + o(1))n^{1/2} \quad \text{as } n \rightarrow \infty$$

uniformly for  $0 \leq k \leq n$ .

## I. Introduction

In the Ph.D. thesis [18], written under the supervision of Jim Massey, Jürg Ruprecht has proposed coding schemes designed for effective multipath estimation. Such schemes might be useful in indoor wireless systems [19, 21] or other communication settings. These schemes use *invertible sequences*, which are sequences  $a_0, \dots, a_n$ , with  $a_j = \pm 1$  for each  $j$ , such that the associated polynomial

$$f(z) = \sum_{j=0}^n a_j z^j \tag{1.1}$$

satisfies

$$f(e^{2\pi i t}) \neq 0 \quad \text{for all real } t. \tag{1.2}$$

In some situations these schemes use *invertible periodic sequences*, for which the polynomial  $f(z)$  only has to satisfy

$$f(e^{2\pi i k/(n+1)}) \neq 0, \quad 0 \leq k \leq n. \tag{1.3}$$

(These invertible periodic sequences possess inverses under periodic convolution, which is required for Ruprecht's maximum-likelihood estimation methods [18].) For best performance in estimating multipath interference, it is desirable to find invertible periodic sequences that maximize

$$R_p(f) = \frac{n+1}{\sum_{k=0}^n |f(e^{2\pi ik/(n+1)})|^{-2}}. \quad (1.4)$$

In [18], this figure of merit is referred to as even processing gain  $G_e^{(vs)}$  of a sequence  $s$  and its periodic inverse  $v$ , and is defined in a much more complicated form. However, a short calculation based on the formulas on p. 27 and in Appendix A of [18] shows that it equals our  $R_p(f)$ . We will call  $R_p(f)$  the periodic Ruprecht merit factor, to distinguish it from other merit factors, such as that of Golay [5, 12, 16], as well as the aperiodic Ruprecht merit factor, defined as

$$R_a(f) = \left( \int_0^1 |f(e^{2\pi it})|^{-2} dt \right)^{-1}. \quad (1.5)$$

(For  $R_a(f)$  to exist, we require that  $f(z)$  be an invertible sequence.) Sequences with large  $R_a(f)$  are more desirable than those with large  $R_p(f)$  because they can be used for transmission [19] not just for multipath estimation. Unfortunately while we will provide constructions of sequences with large  $R_p(f)$ , the problem of obtaining large  $R_a(f)$  remains open.

Since

$$\sum_{k=0}^n |f(e^{2\pi ik/(n+1)})|^2 = (n+1)^2 \quad (1.6)$$

and

$$\int_0^1 |f(e^{2\pi it})|^2 dt = n+1 \quad (1.7)$$

by a familiar calculation, the Cauchy-Schwarz inequality shows that  $R_p(f) \leq n+1$ ,  $R_a(f) \leq n+1$  for any sequence  $a_0, \dots, a_n$ . How close can  $R_a(f)$  and  $R_p(f)$  come to  $n+1$ ? Ruprecht [18] lists in Table B.6 the sequences  $a_0, \dots, a_n$  with the highest values of  $R_p(f)$  for  $n \leq 29$ , as well as some sequences with high values of  $R_p(f)$  for  $30 \leq n \leq 32$ . The maximal value of  $R_p(f)$  for  $n = 29$  is 26.6583, for example. Ruprecht also gives, in Table B.8, the best sequences drawn from a restricted class, that of the *skew-symmetric*  $a_j$  (i.e., those with even  $n$  and  $a_{n/2-r} = (-1)^r a_{n/2+r}$ ) for  $n \leq 44$ . (The value for  $n = 44$  is incorrect, though. See [16].) The maximal value of  $R_p(f)$  for  $n = 42$  is 37.4244. In Tables B.9 and B.10 of [18] Ruprecht lists sequences with large  $R_a(f)$ , for  $n \leq 23$  in the general case and  $n \leq 44$  for the skew-symmetric case. For example, for  $n = 44$  he gives a skew-symmetric sequence with  $R_a(f) = 39.7753$ . Most of the values, especially for large  $n$ , are not known to be maximal. Skew-symmetric sequences with large  $R_a(f)$  and  $R_p(f)$  for  $n \leq 90$  (obtained from a search for other types of extremal  $\pm 1$  sequences) are given in [16]. The nonexhaustive search for high  $R_a(f)$  and  $R_p(f)$  that is documented in that paper has produced a value of  $R_p(f) = 77.5820$  for  $n = 90$ , for example.

What can one do for larger lengths  $n$ ? Random choices of the  $a_j$  almost always give small values of  $R(f)$  (cf. [16]). This is because random trigonometric polynomials have small minimal absolute values [7, 17], as was conjectured by Littlewood [9, 10]. Thus the situation is completely different than it is in coding theory, where random codes are good.

Sometimes one can construct a sequence with a large Ruprecht merit factor from shorter sequences. For example, if  $n = 12$  and  $(a_0, \dots, a_n) = (1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1)$  is

the 13-term Barker sequence, with associated polynomial  $f(z)$ , then  $f(z^{13})f(z)$  is a polynomial associated to a  $\pm 1$  sequence of 169 terms, and

$$R_a(f(z^{13})f(z)) = 153.1014 \dots, \quad R_p(f(z^{13})f(z)) = 154.6331 \dots \quad (1.8)$$

However, even this construction does not produce good asymptotic results.

The main result of this note is to show that high periodic Ruprecht merit factors can be achieved for a dense sequence of values of  $n$ .

**Theorem 1.** *There is a constant  $c > 0$  such that if  $n = q - 1$  for  $q$  a prime, then there exists a sequence  $a_0, \dots, a_n$  with  $a_j = \pm 1$  for all  $j$  such that*

$$n - cn^{1/3} \leq R_p(f) \leq n + 1. \quad (1.9)$$

The proof of Theorem 1, given in Section 2, shows how to construct these sequences. The sequences of Theorem 1 do have higher  $R_a(f)$  than random sequences, but not very high ones. There is a discussion of this disappointing behavior in Section 4.

The search for  $\pm 1$  sequences with large Ruprecht merit factors is just one part of the huge subject of extremal and statistical properties of  $\pm 1$  sequences. For results, surveys, and applications, see [11, 16, 20]. In particular, there are connections to the search for sequences with large Golay merit factor [5, 12, 16].

For  $R_p(f)$  to be large,  $|f(\exp(2\pi ik/(n+1)))|$  has to be large for most  $k$ . Erdős [4] and Littlewood [9, 10] have raised the question of whether there exist  $\pm 1$  sequences  $a_0, \dots, a_n$  such that the associated polynomials  $f(z)$  satisfy

$$\min_{|z|=1} n^{-1/2}|f(z)| = 1 + o(1) \quad \text{as } n \rightarrow \infty. \quad (1.10)$$

If such sequences existed, then we would have  $R_a(f) \sim n$  and  $R_p(f) \sim n$  as  $n \rightarrow \infty$  for their polynomials. The current evidence is that such sequences do not exist (cf. [16]). However, to obtain large  $R_p(f)$  we do not require (1.10) to hold. We even do not require  $n^{-1/2}|f(\exp(2\pi ik/(n+1)))| = 1 + o(1)$  as  $n \rightarrow \infty$  to hold uniformly for all  $k$ ,  $0 \leq k \leq n$ . Instead, we prove Theorem 1 by modifying the Legendre sequence  $a_j = \left(\frac{j}{q}\right)$ . It is easy to see that modifications of that sequence achieve  $R_p(f) \sim n$  as  $n = q - 1 \rightarrow \infty$ , but the difference  $R_p(f) - n$  usually turns out to be much larger than  $n^{1/3}$  when one uses some of the obvious methods. By a careful analysis of what happens to  $R_p(f)$  as the Legendre sequence is changed, we can obtain the bound of Theorem 1.

The construction of Theorem 1 produces sequences for which  $n^{-1/2}|f(\exp(2\pi ik/(n+1)))| = 1 + o(1)$  as  $n \rightarrow \infty$  uniformly in  $k$  satisfying  $1 \leq k \leq n$ . For  $k = 0$ , though,  $|f(1)|$  is of order  $n^{1/3}$ . However, we prove the following result.

**Theorem 2.** *If  $n = q - 1$  for  $q$  a prime, then there exists a sequence  $a_0, \dots, a_n$  with  $a_j = \pm 1$  for all  $j$  such that*

$$n^{-1/2}|f(\exp(2\pi ik/(n+1)))| = 1 + O(n^{-1/4}(\log n)^{1/2}) \quad \text{as } n \rightarrow \infty \quad (1.11)$$

*uniformly in  $k$ ,  $0 \leq k \leq n$ .*

If we use only the bound (1.11) for the sequences of Theorem 2, we find that these sequences have  $R_p(f) \geq n - c'n^{3/4}(\log n)^{1/2}$  for some constant  $c' > 0$ . With more care, one can show that these sequences have larger  $R_p(f)$ , but the bound for  $n - R_p(f)$  that one can prove for these sequences appears to be considerably weaker than that given by Theorem 1 for its sequences.

We note that if

$$n^{-1/2}|f(e^{2\pi ik/(n+1)})| = 1, \quad 0 \leq k \leq n, \quad (1.12)$$

which is equivalent to  $R_p(f) = n + 1$ , then  $a_0, \dots, a_n$  is a Barker sequence and also the first row of a circulant Hadamard matrix, and so is thought not to exist for  $n > 3$  [3, 21]. However, there is still no proof of this conjecture.

We leave several problems open. For example, can Theorems 1 and 2 be generalized so that  $n$  does not have to be of the form  $n = q - 1$  for  $q$  a prime? Also, can one prove analogs of Theorems 1 and 2 for the aperiodic merit factor  $R_a(f)$ ? Numerical evidence (cf. [16]) suggests that there do exist  $\pm 1$  sequences  $a_0, \dots, a_n$  for  $n \geq 10$  such that the associated polynomials  $f(z)$  have

$$\min_{|z|=1} n^{-1/2}|f(z)| \geq 1/2. \quad (1.13)$$

A sequence satisfying (1.13) is guaranteed to have  $R_a(f) \geq n/4$ . However, since  $R_a(f)$  is an average result, we might expect that some of these sequences might have  $R_a(f) \sim n$  as  $n \rightarrow \infty$ . That is what seems to happen for the sequences listed in [16].

## II. Proof of Theorem 1

Let  $q$  be an odd prime, and define

$$\zeta = \exp(2\pi i/q), \quad (2.1)$$

$$g(z) = \sum_{k=1}^{q-1} \left(\frac{k}{q}\right) z^k, \quad (2.2)$$

where  $\left(\frac{k}{q}\right)$  is the Legendre symbol. (Thus  $\left(\frac{k}{q}\right)$  is 0 for  $k = 0$ , 1 if  $k$  is a nonzero quadratic residue modulo  $q$ , and  $-1$  if  $k$  is a nonresidue modulo  $q$ .) The  $g(\zeta^j)$  are Gauss sums, and have an extensive literature. It is known (and easy to derive [1]) that

$$g(1) = 0, \quad g(\zeta^j) = \left(\frac{j}{q}\right) g(\zeta) \quad \text{for } 1 \leq j \leq q-1. \quad (2.3)$$

It is also easy to see (cf. [1]) that

$$g(\zeta)^2 = (-1)^{(q-1)/2} q. \quad (2.4)$$

It is further known that

$$g(\zeta) = \begin{cases} q^{1/2}, & q \equiv 1 \pmod{4}, \\ iq^{1/2}, & q \equiv 3 \pmod{4}, \end{cases} \quad (2.5)$$

but this is much harder to prove, and we will not use it. It is also known that  $g(z)$  is large for some  $z$  with  $|z| = 1$  [14].

We cannot use the sequence of coefficients of  $g(z)$ , because (i)  $a_0 = 0$  and (ii)  $g(1) = 0$ . The main idea behind the construction below is to modify  $g(z)$  slightly. We note that if we take  $f(z) = 1 + g(z)$ , then the coefficient sequence does consist of  $\pm 1$ 's, and  $f(1) = 1$ ,  $|f(\zeta^k)| \geq q^{1/2} - 1$  for  $1 \leq k \leq q - 1$ . Therefore  $R(f) \sim q/2$  as  $q \rightarrow \infty$ , and this already gives a merit factor far superior to that of almost all  $\pm 1$  sequences.

We set

$$f(z) = g(z) + h(z), \quad (2.6)$$

where

$$h(z) = a - 2 \sum_{k \in S} \left( \frac{k}{q} \right) z^k , \quad (2.7)$$

$a = \pm 1$ , and  $S \subseteq \{1, \dots, q-1\}$ ,  $|S| < q^{1/2}/100$ . It is easy to see, using the results on maximal values of random trigonometric polynomials, that random choices of  $S$  give  $R(f) \sim n$  as  $n \rightarrow \infty$ . What we show, however, is that a nonrandom choice produces much better answers due to the special number-theoretic properties of the Legendre sequence. We will select  $S$  to consist entirely of residues or else entirely of nonresidues, so that

$$\left( \frac{k}{q} \right) = b \quad \text{for all } k \in S , \quad (2.8)$$

where  $b = \pm 1$ . The precise selection of  $a$  and  $S$  will be discussed later. We now observe that all coefficients of  $f(z)$  are  $\pm 1$ . Further, we have

$$f(1) = g(1) = a - 2b|S| . \quad (2.9)$$

For  $1 \leq j \leq q-1$ ,

$$|f(\zeta^j)|^2 = |g(\zeta^j) + h(\zeta^j)|^2 = q + |h(\zeta^j)|^2 + 2 \operatorname{Re} \left( \overline{g(\zeta^j)} h(\zeta^j) \right) . \quad (2.10)$$

Since  $|S| < q^{1/2}/100$ , we find that for large  $q$ ,

$$|h(\zeta^j)| < q^{1/2}/10 = |g(\zeta^j)|/10 . \quad (2.11)$$

Therefore, we can write, for  $1 \leq j \leq q-1$ ,

$$|f(\zeta^j)|^{-2} = q^{-1} \left( 1 - 2q^{-1} \operatorname{Re} \left( \overline{g(\zeta^j)} h(\zeta^j) \right) + O(q^{-1}|h(\zeta^j)|^2) \right) . \quad (2.12)$$

This implies, by (2.3), that

$$|f(\zeta^j)|^{-2} = q^{-1} \left( 1 - 2q^{-1} \left( \frac{j}{q} \right) \operatorname{Re} \left( \overline{g(\zeta)} h(\zeta^j) \right) + O(q^{-1}|h(\zeta^j)|^2) \right) , \quad (2.13)$$

and therefore

$$\sum_{j=1}^{q-1} |f(\zeta^j)|^{-2} = \frac{q-1}{q} - 2q^{-2} \operatorname{Re} \overline{g(\zeta)} \sum_{j=1}^{q-1} \left( \frac{j}{q} \right) h(\zeta^j) + O \left( q^{-2} \sum_{j=1}^{q-1} |h(\zeta^j)|^2 \right) . \quad (2.14)$$

Now

$$\sum_{j=1}^{q-1} |h(\zeta^j)|^2 \leq \sum_{j=0}^{q-1} |h(\zeta^j)|^2 = 4q|S| . \quad (2.15)$$

On the other hand, by (2.8),

$$\begin{aligned} \sum_{j=1}^{q-1} \left( \frac{j}{q} \right) h(\zeta^j) &= a \sum_{j=1}^{q-1} \left( \frac{j}{q} \right) - 2b \sum_{k \in S} \sum_{j=1}^{q-1} \left( \frac{j}{q} \right) \zeta^{kj} \\ &= -2b \sum_{k \in S} \left( \frac{k}{q} \right) g(\zeta) = -2g(\zeta)|S| . \end{aligned} \quad (2.16)$$

If we now combine (2.9), (2.14), (2.15), and (2.16), we find that

$$\sum_{j=0}^{q-1} |f(\zeta^j)|^{-2} = (2b|S| - a)^{-2} + \frac{q-1}{q} + O(q^{-1}|S|) . \quad (2.17)$$

If we select  $|S| \sim q^{1/3}$  as  $q \rightarrow \infty$ , say, then we obtain

$$\sum_{j=0}^{q-1} |f(\zeta^j)|^{-2} = 1 + O(q^{-2/3}) , \quad (2.18)$$

which yields the claim of Theorem 1.

### III. Proof of Theorem 2

Theorem 2 follows from a modification of the proof of Theorem 1, using methods similar to those of [15]. As in the preceding section, we define  $f(z)$  by (2.6) and (2.7) with  $a = 1$ . However, this time we will take  $S$  to be of size about  $q^{1/2}$ , and it will contain only nonresidues. The set  $S$  will be chosen at random, with each  $k$ ,  $1 \leq k \leq q-1$ ,  $\left(\frac{k}{q}\right) = -1$ , selected independently to be in  $S$  with probability

$$Pr(k \in S) = q^{-1/2}/2 . \quad (3.1)$$

Thus we have

$$h(z) = 1 - 2 \sum_{k=1}^{q-1} \eta_k \left(\frac{k}{q}\right) z^k , \quad (3.2)$$

where  $\eta_k = 0$  or  $1$  is a random variable with  $\eta_k$  identically  $0$  if  $\left(\frac{k}{q}\right) = 1$ , and  $\mathcal{E}(\eta_k) = q^{-1/2}/2$  if  $\left(\frac{k}{q}\right) = -1$ .

We need to determine the behavior of  $h(\zeta^j)$  for  $0 \leq j \leq q-1$ , where  $\zeta$  is defined by (2.1). We first consider the expected value  $\mathcal{E}(h(\zeta^j))$  for a fixed  $j$ . For  $j = 0$  we have

$$\mathcal{E}(h(1)) = 1 + 2 \sum_{\substack{k=1 \\ \left(\frac{k}{q}\right)=-1}}^{q-1} \mathcal{E}(\eta_k) = 1 + (q-1)q^{-1/2} = q^{1/2} + 1 - q^{-1/2} . \quad (3.3)$$

For  $1 \leq j \leq q-1$ , on the other hand,

$$\mathcal{E}(h(\zeta^j)) = 1 + q^{-1/2} \sum_{\substack{k=1 \\ \left(\frac{k}{q}\right)=-1}}^{q-1} \zeta^{kj} . \quad (3.4)$$

Since  $\sum_{k=0}^{q-1} \zeta^{kj} = 0$  for  $1 \leq j \leq q-1$ , the sum in (3.4) is  $-\left(\left(\frac{j}{q}\right)q(\zeta) + 1\right)/2$ . Hence

$$\mathcal{E}(h(\zeta^j)) = 1 - q^{-1/2}/2 - \left(\frac{j}{q}\right)q^{-1/2}g(\zeta)/2 , \quad (3.5)$$

and so

$$\mathcal{E}(h(\zeta^j)) = O(1) . \quad (3.6)$$

We conclude that  $\mathcal{E}(h(\zeta^j))$  has the desired behavior uniformly for all  $j$ ,  $0 \leq j \leq q-1$ .

It remains to prove that for some choice of coefficients,  $h(\zeta^j)$  will be close to  $\mathcal{E}(h(\zeta^j))$  for all  $j$ . This will follow from the following result, which is similar to those in [6, 15].

**Lemma 1.** *There exists a constant  $C > 0$  such that if*

$$W = \sum_{k=1}^n \tau_k a_k , \quad (3.7)$$

where the  $a_k$  are real constants,  $|a_k| \leq 1$  for all  $k$ , and the  $\tau_k$  are independent random variables with

$$\Pr(\tau_k = -\gamma_k) = 1 - \gamma_k, \quad \Pr(\tau_k = 1 - \gamma_k) = \gamma_k , \quad (3.8)$$

for some constants  $\gamma_k$ ,  $0 \leq \gamma_k \leq 1$ , then

$$\Pr\left(|W| > C \left(\sum_{k=1}^n \gamma_k\right)^{1/2} (\log n)^{1/2}\right) < n^{-10} . \quad (3.9)$$

**Proof.** We have, for any  $\lambda > 0$ ,

$$\Pr(W > x) e^{\lambda x} \leq \mathcal{E}(e^{\lambda W}) . \quad (3.10)$$

Now the  $\tau_k$  are independent, so

$$\mathcal{E}(e^{\lambda W}) = \prod_{k=1}^n \mathcal{E}(e^{\lambda \tau_k a_k}) . \quad (3.11)$$

We next note that

$$\mathcal{E}(e^{\lambda \tau_k a_k}) = e^{-\lambda \gamma_k a_k} (1 - \gamma_k) + e^{\lambda(1-\gamma_k)a_k} \gamma_k \leq e^{C' \lambda^2 \gamma_k} \quad (3.12)$$

if  $C'$  is sufficiently large. Therefore

$$\Pr(W > x) \leq \exp\left(C' \lambda^2 \sum_{k=1}^n \gamma_k - \lambda x\right) . \quad (3.13)$$

This bound holds for all  $\lambda > 0$ , so for  $x > 0$  we select  $\lambda = x(2C' \sum \gamma_k)^{-1}$  and obtain

$$\Pr(W > x) \leq \exp\left(-x^2 \left(4C' \sum_{k=1}^n \gamma_k\right)^{-1}\right) . \quad (3.14)$$

Since the same bound for  $\Pr(W < -x)$  follows by applying (3.14) to the problem with  $a_k$  replaced by  $-a_k$ , we easily obtain the claim of the lemma. ■

To conclude the proof of Theorem 2, we apply Lemma 1 to the real and imaginary parts of

$$h(\zeta^j) - \mathcal{E}(h(\zeta^j)) , \quad 0 \leq j \leq q-1 .$$

We find that with probability  $\geq 1 - n^{-8}$ ,

$$\left|h(\zeta^j) - \mathcal{E}(h(\zeta^j))\right| < 10Cq^{1/4}(\log q)^{1/2} \quad (3.15)$$

holds for all  $j$ ,  $0 \leq j \leq q-1$ . Therefore

$$|f(\zeta^j)| = q^{1/2} + O(q^{1/4}(\log q)^{1/2}) \quad (3.16)$$

for all  $j$ , which yields Theorem 2.

There is a method of Kolountzakis [8] that often manages to remove factors such as the  $(\log q)^{1/2}$  in the estimate (3.16). However, it does not seem to apply in this case.

## IV. Final remarks

How large are the  $R_p(f)$  produced by the above construction for moderate lengths  $n$ ? For  $n = 82$ , the largest  $R_p(f)$  that is known is 72.02 [16]. The construction of this section produces a sequence with  $R_p(f) = 69.90$ . Surprisingly, this result is achieved with  $|S| = 1$ . As was noted at the beginning of this section, if  $|S| = 0$ , then  $R_p(f) \sim q/2$  as  $q \rightarrow \infty$  (for  $n = q - 1$ ,  $q$  a prime). However, if we choose  $|S| = 1$ ,  $a = -b = 1$ , then we obtain  $R_p(f) \sim 9q/10$  as  $q \rightarrow \infty$ . Choices of  $S$  with  $|S| \geq 2$  give better  $R_p(f)$  only for  $q \gtrsim 130$ , and the improvement is slight initially. (We note also that while  $R_p(f)$  is the same for all choices of  $S$  with  $|S| = 1$ ,  $a = -b$ , the precise selection of  $S$  does make a slight difference for  $|S| \geq 2$ .) The resulting sequences for  $p < 180$  are not as good (say, when judged by the value of the ratio  $R_p(f)/(n + 1)$ ) as the sequence obtained from the 13-term Barker sequence (see the discussion preceding Eq. (1.8)), but they are better than some other sequences that have been proposed. For example, Ruprecht, Neeser, and Hufschmid [19] list a sequence with  $n = 143$  and  $R_p(f) = 120.69862$ . Our construction with  $n = q - 1 = 138$  and  $|S| = 2$  yields a value of  $R_p(f) = 121.32578$ , so that  $R_p(f)$  is higher even though  $n$  is lower. (It should be mentioned, though, that the Ruprecht et al. sequence was chosen to have a high  $R_a(f)$ , not a high  $R_p(f)$ .)

The construction of Theorem 1 produces a sequence with high  $R_p(f)$  because the polynomials associated to the Legendre sequences already have the desired behavior at the points  $z = \exp(2\pi i k/q)$  for  $1 \leq k \leq q - 1$ , and it is only at  $z = 1$  that they need to be modified. Unfortunately the behavior of these polynomials at other points on the unit circle is not as well controlled. Montgomery [14] showed that

$$\max_{|z|=1} |f(z)| > 2\pi^{-1} q^{1/2} \log \log q \quad (4.1)$$

for all sufficiently large  $q$ , and he conjectured that this bound is of the right order of magnitude. If Montgomery's conjecture is right, these polynomials will be smaller than random ones, which reach  $q^{1/2}(\log q)^{1/2}$  in size (cf. [16]). However, these polynomials do have small minimal absolute values. B. Conrey and A. Granville have observed (unpublished) that the polynomial  $g(z)$  of Eq. (2.2) has  $> p/2$  zeros with  $|z| = 1$ . Therefore it is not straightforward to modify those polynomials to obtain large  $R_a(f)$ . The highest value of  $R_a(f)$  that our construction obtains for  $n = 138$ ,  $|S| \leq 2$  is 28.764, while the Ruprecht et al. sequence of [19] has  $R_a(f) = 110.57658$ . However, there are ways of modifying our construction to obtain higher values of  $R_a(f)$ . For example, it is known (see [16] for references) that cyclic shifts of the coefficients of  $1 + g(z)$  (with  $g(z)$  given by (2.2)) produce improved values for the Golay merit factor, which measures how far  $|f(z)|$  is away from  $(n + 1)^{1/2}$  on average as  $z$  runs over  $|z| = 1$ . (We note that cyclic shifts of coefficients of  $f(z)$  do not affect the value of  $R_p(f)$ .) A nonexhaustive search of cyclic shifts of the sequences constructed in Section II with  $n = 138$ ,  $|S| \leq 2$ , found a sequence with  $R_a(f) = 110.2457$ , which is better than the Ruprecht et al. sequence of [19], since the length is less. Thus modifications of our construction yield good values even for  $R_a(f)$ , although there is no proof that they will work for large lengths. It is also possible to try other modifications, which might yield even better results.

## Acknowledgement

The author thanks Jürg Ruprecht for helpful correspondence.

## References

- [1] T. M. Apostol, *Introduction to Analytic Number Theory*, Springer, 1976.

- [2] G. Björck, Functions of modulus 1 on  $Z_n$  whose Fourier transforms have constant modulus, and “cyclic  $n$ -roots,” pp. 131–140 in *Recent Advances in Fourier Analysis and its Applications*, J. S. Byrnes and J. F. Byrnes, eds., Kluwer, 1990.
- [3] P. J. Davis, *Circulant Matrices*, Wiley, 1979.
- [4] P. Erdős, Some unsolved problems, *Michigan Math. J.* 4 (1957), 291–300.
- [5] M. J. E. Golay, A class of finite binary sequences with alternate autocorrelation values equal to zero, *IEEE Trans. Information Theory IT-18*, 449–450, 1972.
- [6] J.-P. Kahane, *Some Random Series of Functions*, Heath, 1968.
- [7] B. S. Kashin, Properties of random trigonometric polynomials with coefficients  $\pm 1$ , *Moscow Univ. Math. Bull.* 42, no. 5, 45–51, 1987.
- [8] M. N. Kolountzakis, On nonnegative cosine polynomials with nonnegative integral coefficients, *Proc. Amer. Math. Soc.*, to appear.
- [9] J. E. Littlewood, On polynomials  $\Sigma z^m$ ,  $\Sigma e^{\alpha m i} z^m$ ,  $z = e^{i\theta}$   $\Sigma e^{\alpha m i} z^m$ ,  $z = e^{i\theta}$ , *J. London Math. Soc.* 41 (1966), 367–376. *Reprinted in the Collected Papers of J. E. Littlewood*, vol. 2, pp. 1423–1433, Oxford Univ. Press, 1982.
- [10] J. E. Littlewood, *Some Problems in Real and Complex Analysis*, Heath, 1968.
- [11] H. D. Lüke, *Korrelationssignale*, Springer, 1992.
- [12] J. L. Massey, Marcel E. Golay (1902–1989), *IEEE Inform. Theory Newsletter*, June 1990.
- [13] O. C. McGehee, Gaussian sums and harmonic analysis on finite fields, pp. 171–191 in *Contemporary Mathematics* #91, Am. Math. Soc. 1989.
- [14] H. L. Montgomery, An exponential polynomial formed with the Legendre symbol, *Acta Arith.* 37, 375–380, 1980.
- [15] A. M. Odlyzko, Minima of cosine sums and maxima of polynomials on the unit circle, *J. London Math. Soc.* (2), 26, 412–420, 1982.
- [16] A. M. Odlyzko, Extremal and statistical properties of trigonometric polynomials with  $\pm 1$  and 0, 1 coefficients, manuscript in preparation.
- [17] A. M. Odlyzko, Minimal absolute values of random trigonometric polynomials with  $\pm 1$  coefficients, manuscript in preparation.
- [18] J. Ruprecht, Maximum-likelihood estimation of multipath channels, Ph.D. thesis, ETH, 1989. (Published by Hartung Gorre Verlag, Konstanz, Germany, 1989, ISBN 3-89191-270-6.)
- [19] J. Ruprecht, F. D. Neeser, and M. Hufschmid, Code time division multiple access: an indoor cellular system, *Proc. IEEE Vehicular Techn. Conf. VTC '92*, pp. 1–4, 1992.
- [20] M. R. Schroeder, *Number Theory in Science and Communication*, Springer 1984.
- [21] W. D. Wallis, A. P. Street, and J. S. Wallis, *Combinatorics: Room Squares, Sum-Free Sets, Hadamard Matrices*, Lecture Notes in Math. #292, Springer, 1972.
- [22] J.-P. de Weck and J. Ruprecht, Real-time ML estimation of very frequency selective multipath channels, pp. 908.5.1–908.5.6 in *Proc. Globecom 1990*, (San Diego), IEEE Press, 1990.

# Five Views of Differential MSK: A Unified Approach

Bixio Rimoldi

Department of Electrical Engineering

Washington University

St. Louis, MO 63130, USA

Dedicated to James L. Massey on the occasion of his 60th birthday.

## Abstract

Minimum-Shift-Keying is a digital modulation which is of particular interest for both practical applications and theoretical study. It is of interest for theoretical study since it can be examined from (at least) five points of view, each one giving different insight and suggesting alternative implementations. While four of the five points of view are not new, in this paper we (re)discover them using a different approach which is particularly appealing: it is characterized by the fact that all five views follow in a concise and straightforward way from a single expression, which is Massey's maximum likelihood state sequence decoding rule for MSK. The usual additive white Gaussian noise channel is assumed.

## I Introduction

This paper is concerned with the study of minimum-shift-keying (MSK) and differential MSK (DMSK). Even though we will concentrate on the latter, which is more basic, all results apply indirectly to the former which is the more popular of the two. Throughout the paper it will be assumed that the channel is the additive white Gaussian noise channel.

DMSK is an outstanding modulation scheme since it has qualities that make it important for practical applications as well as for theoretical study. It is of interest for study because it can be examined from (at least) five points of view, each one giving different insight and suggesting alternative implementations. Briefly, these five points of view can be summarized as follows: (1) DMSK is a continuous-phase frequency shift keying (CPFSK) modulation scheme; (2) (Massey) DMSK is a special case of a set of encoded modulation schemes that has an optimum demodulator that needs to process the received signal over only two symbol intervals; (3) (New) DMSK is a form of diversity modulation; (4) (de Buda) DMSK is a special form of offset quadrature phase shift keying (OQPSK) modulation; (5) (Amoroso and Kivett) DMSK is a special case of antipodal modulation.

The author recalls being surprised when (during his doctoral study) he learned about the existence of so many aspects of DMSK. He also recalls being surprised by the fact that the approaches used in the literature to describe different implementations of DMSK appeared to be unrelated (compare e.g. [1],[2], and [3]). One would expect that for such a simple modulation scheme various implementation possibilities are obtained by minor manipulation of a single expression.

In this paper we suggest a concise derivation that, with obvious manipulations of a single expression, leads to the four known ways and a new way to describe DMSK.

The paper is organized as follows. We first describe MSK and DMSK as special cases of CPFSK and point out that a possible maximum-likelihood decoder can be implemented with the Viterbi algorithm. Then, following Massey [3], we derive a maximum-likelihood decoder for DMSK that needs to process the received signal over only two symbol intervals. Four obvious ways to implement this test lead to the remaining four receiver implementations which, in turn, suggest four different transmitter implementations.

This paper is dedicated to Jim L. Massey. There is no doubt that my most valuable educational and professional experiences in one way or another have involved him. First of all, Jim enticed me into the broad area of digital communications. He did it through his energetic and enthusiastic lectures filled with exciting technical results explained with lucidity and elegance. Learning information theory, coding, and cryptography from Jim was an exhilarating experience. Equally gratifying and, if possible, even more energizing, were our meetings in which I would report my research progress on continuous phase modulation. I was extremely anxious going into those meetings. (As a Swiss, with the package of formalities that comes with it, it took me a while to get rid of the anxiety of talking to a professor of such stature. The language and cultural differences weren't helping at all.) Early in the meeting Jim would often interrupt to ask questions. It felt good to realize that, after all, what we were discussing wasn't totally trivial to him either. As the meeting proceeded, Jim would generate more and more ideas and "play" with them for a while. He would find examples with such ease and work out details with such elegance, that to me it was like observing the prima ballerina of the Bolshoi practicing Swan Lake. Jim's influence on his students does not stop with graduation. Jim goes out of his way to talk about his students, both current and former, and tell what they do. There is no doubt that this helps them a lot in their careers. I wonder if anybody else would have changed their slides the night before giving the Shannon Lecture to include a last minute student observation. Jim did it and I was flattered that he valued my contribution enough to make a slide out of it for his Shannon Lecture. For a student, such things are priceless.

This paper is related to Jim Massey's work because its key ingredient is Massey's maximum-likelihood state sequence estimator for MSK derived in [3].

## II MSK and DMSK: Five Points of View

While each of the MSK viewpoints mentioned in the preceding section has its own merit, for us the most natural way to introduce MSK is to see it as a special case of continuous-phase frequency shift keying (CPFSK) modulation. This is our starting point.

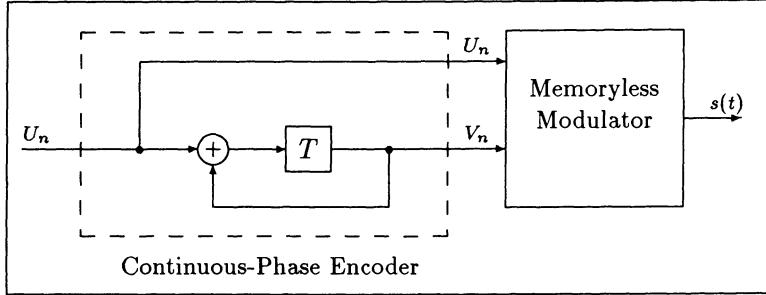


Figure 1: Implementation of the MSK transmitter: CPFSK approach. Addition is modulo 2.

### The CPFSK Approach

In this section we describe MSK as a special case of CPFSK. We use the notation introduced in [4].

The MSK signal can be described by

$$s(t, \mathbf{U}) := \sqrt{\frac{2E}{T}} \cos[2\pi f_0 t + \psi(t, \mathbf{U})], \quad t \geq 0, \quad (1)$$

where:

$$\psi(t, \mathbf{U}) := \pi V_n + \pi \frac{t - nT}{T} U_n, \quad 0 \leq t - nT < T, \quad (2)$$

is the information-carrying phase in the  $n$ th interval,  $n = 0, 1, 2, \dots$ ;  $\mathbf{U} := (U_0, U_1, U_2, \dots)$ ,  $U_i \in \{0, 1\}$ , is the information sequence; and

$$V_n := R_2[\sum_{i=0}^{n-1} U_i] \quad n = 1, 2, \dots \quad (3)$$

is the state during the  $n$ th interval, where  $R_2[\cdot]$  denotes the enclosed number taken modulo two and  $V_0 = 0$  by convention. Since the signal transmitted during the  $n$ th interval is completely specified by  $U_n$  and  $V_n$ , the MSK signal can be generated according to the block diagram shown in Figure 1. In this figure the memoryless modulator is viewed as a table-look-up.

From Equation (2) we see that in any given interval the information-carrying phase either stays constant or it increases (linearly) by  $\pi$  radians. Moreover, we see that  $U_n$  and  $V_n$  are related to the signal transmitted in the  $n$ th interval in a simple way:  $\pi V_n$  is the initial phase and  $\pi U_n$  is the phase increment.

The transfer function of the continuous-phase encoder in Figure 1 is  $[1, \frac{D}{1-D}]$ . If we precode MSK by means of a precoder with transfer function  $(1 - D)$ , then we obtain an equivalent continuous-phase encoder with transfer function  $[1 - D, D] = [1 + D, D]$ , where equality holds because modulo-two addition and subtraction are the same operation. The resulting modulation is denoted differential minimum shift keying (DMSK) and it can be implemented as shown in Figure 2(a). Notice now that the input to the equivalent continuous-phase encoder in Figure 2(a) equals  $V_{n+1}$ . Hence, for DMSK the  $n$ th information

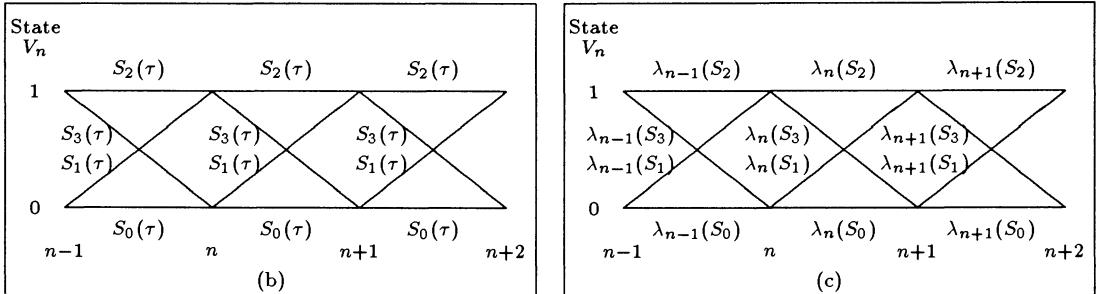
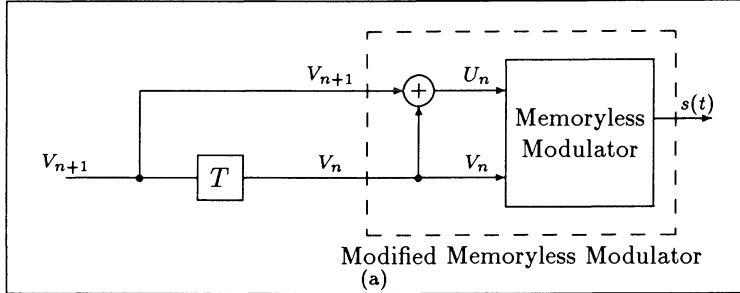


Figure 2: CPFSK implementation of DMSK: (a) Transmitter (addition is modulo 2); (b) State trellis diagram; (c) State trellis diagram labeled with path metric.

| $V_{n+1}$ | $V_n$ | OUTPUT  | $W_{n+1}$ | $W_n$ |
|-----------|-------|---|-----------|-------|
| 0         | 0     | $S_0(\tau) := \sqrt{\frac{2E}{T}} \cos 2\pi f_0 \tau$ | 1         | 1     |
| 1         | 0     | $S_1(\tau) := \sqrt{\frac{2E}{T}} \cos 2\pi f_1 \tau$ | -1        | 1     |
| 1         | 1     | $S_2(\tau) := -S_0(\tau)$                             | -1        | -1    |
| 0         | 1     | $S_3(\tau) := -S_1(\tau)$                             | 1         | -1    |

Table 1: Input/output relationship for the modified memoryless modulator.

symbol, when multiplied by  $\pi$ , equals the final phase. Looking at the transmitted signal, one cannot distinguish between MSK and DMSK; the only difference between the two modulation schemes is in the encoding rule, not in the waveform.

In this paper we will focus on DMSK. The reason is that DMSK is more basic than MSK since the information sequence equals the state sequence (up to a time shift). Thus, the maximum-likelihood decoder for DMSK is identical to the maximum-likelihood state sequence estimator, i.e., the Viterbi decoder [5]. However, any transmitter and any receiver for DMSK can be transformed into a transmitter and a receiver for MSK by premultiplying the input by  $(1/(1-D))$  and postmultiplying the output by  $(1-D)$ , respectively.

Hereafter we assume that  $f_0 T$  is an integer. This is not essential but it simplifies the notation. It implies that in the  $n$ th interval  $s(t, \mathbf{U})$  depends only on  $\tau := t - nT$  and not directly on  $t$ , as can be seen from (1). In other words, the signal set of an interval is a translate of that of any other interval. Table 1 describes the input/output relationship for the modified memoryless modulator, where for convenience we have defined  $f_1 := f_0 + \frac{1}{2T}$ .

The state trellis diagram for both MSK and DMSK is shown in Figure 2(b). The

maximum-likelihood estimator for the state sequence (and DMSK information sequence)  $\mathbf{V}$  is achieved by the Viterbi decoder. The branch metric  $\lambda_n(S_i)$ ,  $i = 0, 1, 2, 3$ , is the correlation between the received signal  $r(t)$  and  $S_i(t - nT)$ , namely

$$\lambda_n(S_i) = \int_{nT}^{nT+T} r(t)S_i(t - nT)dt, \quad i = 0, 1, 2, 3. \quad (4)$$

Figure 2(c) shows the state trellis diagram of MSK with branches labeled with the corresponding metric.

### Massey's Implementation

Let  $\hat{V}_0, \hat{V}_1, \hat{V}_2, \dots$  be the maximum-likelihood state sequence estimate. In this section we derive a sufficient statistic for  $\hat{V}_n$  that is obtained from the received signal in the interval  $(n-1)T \leq t < (n+1)T$ . We follow Massey's approach [3]. Essential for this approach is the fact that  $S_2(\tau) = -S_0(\tau)$  and  $S_3(\tau) = -S_1(\tau)$  imply

$$\lambda_n(S_2) = -\lambda_n(S_0) \text{ and } \lambda_n(S_3) = -\lambda_n(S_1). \quad (5)$$

Assume that a “genie” has told us that for the transmitted signal  $V_{n-1} = 0$  and  $V_{n+1} = 0$ . In Figure 2(c) we see that there are only two transitions between these two states, one going through  $V_n = 1$  and the other through  $V_n = 0$ . Both possibilities are equally likely. The maximum-likelihood decision rule between the two alternatives is

$$\hat{V}_n = 0 \text{ if and only if } \lambda_{n-1}(S_0) + \lambda_n(S_0) \geq \lambda_{n-1}(S_1) + \lambda_n(S_3). \quad (6)$$

But now suppose instead that the genie had told us that  $V_{n-1} = 0$  and  $V_{n+1} = 1$ . Then

$$\hat{V}_n = 0 \text{ if and only if } \lambda_{n-1}(S_0) + \lambda_n(S_1) \geq \lambda_{n-1}(S_1) + \lambda_n(S_2), \quad (7)$$

which, because of (5), is precisely the decoding rule (6).

Similar analyses for the case  $V_{n-1} = 1$  and  $V_{n+1} = 0$  and for the case  $V_{n-1} = 1$  and  $V_{n+1} = 1$  show that, regardless of what the genie says, (6) is always a maximum-likelihood decoding rule for  $\hat{V}_n$ .

For the rest of the paper it will be convenient to replace  $V_n \in \{0, 1\}$  with  $W_n \in \{\pm 1\}$  defined by  $W_n := -(2V_n - 1)$ . Replacing  $V_n$  with  $W_n$  in (6), using (5) to replace  $\lambda_n(S_3)$  with  $-\lambda_n(S_1)$ , and bringing each term to the left side of the inequality, we obtain

$$\hat{W}_n = 1 \text{ if and only if } \lambda_{n-1}(S_0) + \lambda_n(S_0) - \lambda_{n-1}(S_1) + \lambda_n(S_1) \geq 0. \quad (8)$$

This rule is implemented by the receiver in Figure 3(a). This is (a time invariant version of) Massey's receiver for DMSK [3]. This receiver suggests that the DMSK signal can be obtained as shown in Figure 3(b). Indeed this is the case as one can easily verify by comparing its input/output characteristic with Table 1.

The alternative implementations described in this paper will all be obtained from obvious alternative ways to implement Massey's decision rule (8).

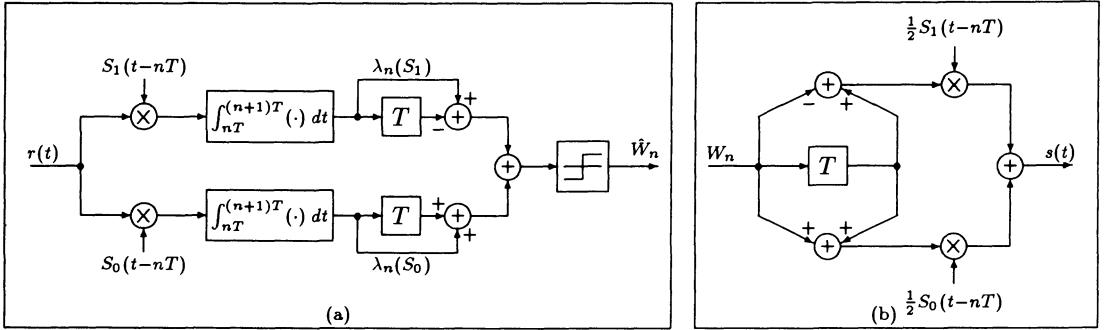


Figure 3: Massey's implementation of DMSK: (a) Optimal receiver; (b) Transmitter. Additions are over the reals.

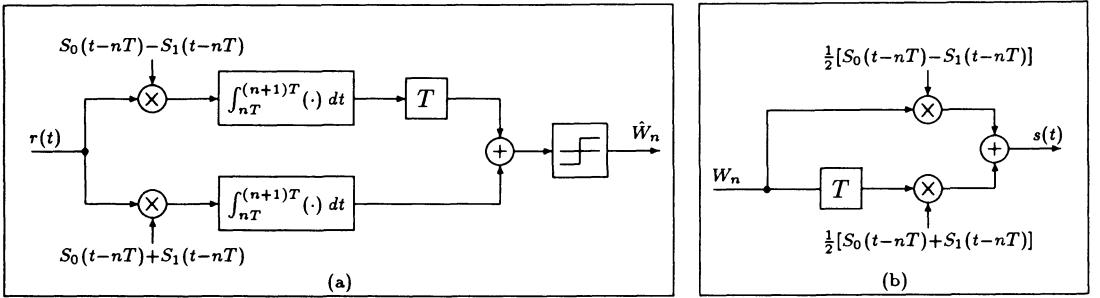


Figure 4: Diversity implementation of DMSK: (a) Optimal receiver; (b) Transmitter. Additions are over the reals.

## Diversity Implementation

The “front end” correlators in Massey’s receiver determine  $\lambda_n(S_0)$  and  $\lambda_n(S_1)$ . Another possibility that is somehow simpler (in terms of number of additions and delays required) is to have correlators that output  $\lambda_n(S_0) + \lambda_n(S_1)$  and  $\lambda_n(S_0) - \lambda_n(S_1)$  as shown in Figure 4(a). This receiver suggests that the DMSK signal can be generated as shown in Figure 4(b). Again, one can easily verify that this transmitter is consistent with the input/output mapping of Table 1.

An interesting interpretation of this implementation is that DMSK is a form of diversity transmission. Indeed, the information symbol  $W_n$  is transmitted twice: first it amplitude modulates  $[S_0(\tau) - S_1(\tau)]$  and then it amplitude modulates the orthogonal signal  $[S_0(\tau) + S_1(\tau)]$ . That  $[S_0(\tau) - S_1(\tau)]$  and  $[S_0(\tau) + S_1(\tau)]$  are orthogonal over one interval follows directly from the observation that  $[S_0(\tau) + S_1(\tau)][S_0(\tau) - S_1(\tau)] = S_0^2(\tau) - S_1^2(\tau)$  and that  $S_0(\tau)$  and  $S_1(\tau)$  are equal energy signals.

It is well known that diversity brings no coding gain on the additive white Gaussian noise channel. This explains why DMSK is as energy efficient as antipodal modulation, but not more.

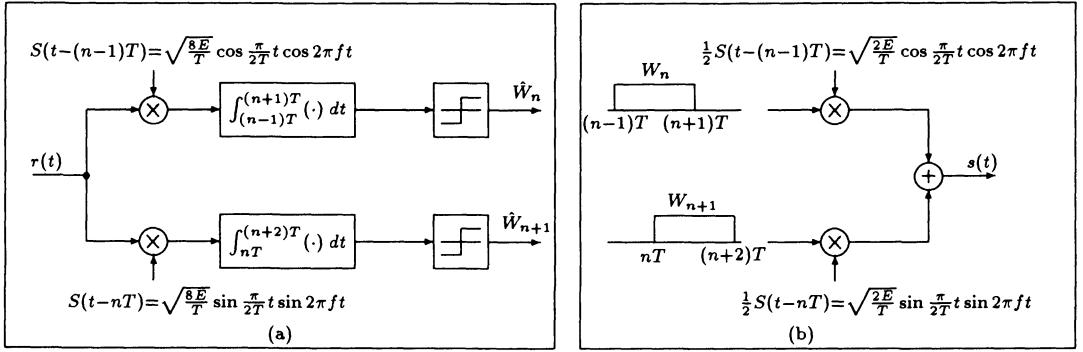


Figure 5: de Buda's parallel implementation of DMSK: (a) Optimal receiver; (b) Transmitter.

### de Buda's Parallel Implementation

A third way to implement Massey's test (8) is to compute the left side of the inequality by means of a single integral, i.e.,

$$\lambda_{n-1}(S_0) + \lambda_n(S_0) - \lambda_{n-1}(S_1) + \lambda_n(S_1) = \int_{(n-1)T}^{(n+1)T} r(t)S(t - (n - 1)T)dt, \quad (9)$$

where

$$S(t) := S_0(t) - S_1(t) + S_0(t - T) + S_1(t - T). \quad (10)$$

Note that now we have to integrate over two adjacent intervals. This is a problem since a single integrate-and-dump circuit can only provide the result for even (or odd) values of  $n$ . One way around this problem is to use a second integral to obtain (9) for odd (or even) values of  $n$ . The result is depicted in Figure 5(a), where  $n$  is assumed to be even and  $f := f_0 + \frac{1}{4T}$ . Showing that

$$S(t - (n - 1)T) = \sqrt{\frac{8E}{T}} \cos 2\pi ft \cos \frac{\pi t}{2T}, \quad (n - 1)T \leq t < (n + 1)T, \quad (11)$$

and that

$$S(t - nT) = \sqrt{\frac{8E}{T}} \sin 2\pi ft \sin \frac{\pi t}{2T}, \quad nT \leq t < (n + 2)T, \quad (12)$$

is straightforward. Details can be found in the Appendix. This suggests that the DMSK signal can be generated as shown in Figure 5(b). That this transmitter implements the mapping shown in Table 1 can be easily verified using (10). For instance, if  $W_n = 1$  and  $W_{n+1} = -1$ , then according to Figure 5(b) the output signal in the  $n$ th interval is  $1/2[S(t - (n - 1)T) - S(t - nT)]$ . For  $nT \leq t < (n + 1)T$  this signal is equal  $S_1(t - nT)$ , as can be verified from (10). The receiver in Figure 5 was first derived by de Buda in [1]. It is referred to as the parallel implementation. The parallel implementation shows that MSK can be seen as a special case of OQPSK.

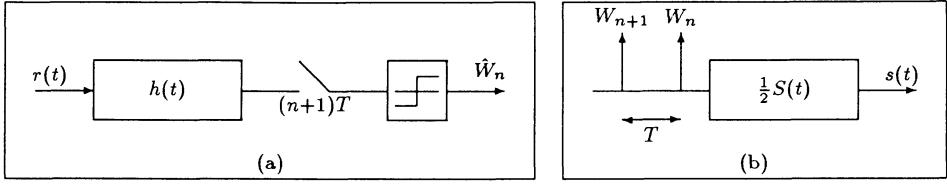


Figure 6: Amoroso and Kivett's serial implementation of DMSK: (a) Optimal receiver; (b) Transmitter.

### Amoroso and Kivett's Serial Implementation

The parallel implementation resulted from the impossibility of computing the left side of (8) for every  $n$  by means of an integrator. However, it can be done by means of a matched filter with impulse response

$$h(t) := \begin{cases} S(T_d - t), & t \in [T_d - 2T, T_d] \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

where  $T_d$  is some delay such that  $T_d \geq 2T$ . The resulting receiver is shown in Figure 6(a) where we have assumed  $T_d = 2T$ . This receiver suggests that the DMSK signal can be generated as in Figure 6(b). Also in this case Table 1 and Definition (10) serve to verify that we obtain the desired output. For obvious reasons, this is denoted the serial implementation.

The serial implementation was discovered by Amoroso and Kivett (see [2, Equation (2)]). It shows that DMSK can be seen as a special form of antipodal modulation. It is special in that  $S(\tau)$  has duration  $2T$  and, therefore, pulse translates overlap. However, this has no effect on the performance of the matched filter receiver since  $S(\tau)$  is orthogonal to  $S(\tau - T)$ . This follows directly from  $S(\tau)S(\tau - T) = [S_0(t - T) + S_1(t - T)][S_0(t - T) - S_1(t - T)] = S_0^2(t - T) - S_1^2(t - T)$  and from the fact that  $S_0(t - T)$  and  $S_1(t - T)$  are equal energy pulses.

The serial implementation is useful at high data rates (100 Mbps and above).

## III Conclusion and Final Comments

In this paper we have described a unified approach leading to the previously known four ways and a new way to interpret DMSK. All five interpretations follow in a straightforward way from Massey's maximum-likelihood test for decoding DMSK. Each transmitter and each receiver for DMSK can be transformed into a transmitter and a receiver for MSK by a simple invertible operation on the information sequence.

Further insight into the diversity interpretation of DMSK can be gained via the following approach suggested by an anonymous reviewer. We start with the CPFSK interpretation of Figure 2(b), with transitions relabeled with  $V_{n+1}, V_n$  for convenience as shown in Figure 7(a). When  $f_0T$  is an integer, ten  $S_0(\tau)$  and  $S_1(\tau)$  are orthogonal. Hence, the signals on the four trellis branches form a biorthogonal signal set as shown in Figure 7(b). By rotating the signal set by  $45^\circ$  and relabeling according to  $W_n := -(2V_n - 1)$ , one obtains the equivalent description of Figure 7(c) and (d). An implementation of this system is shown

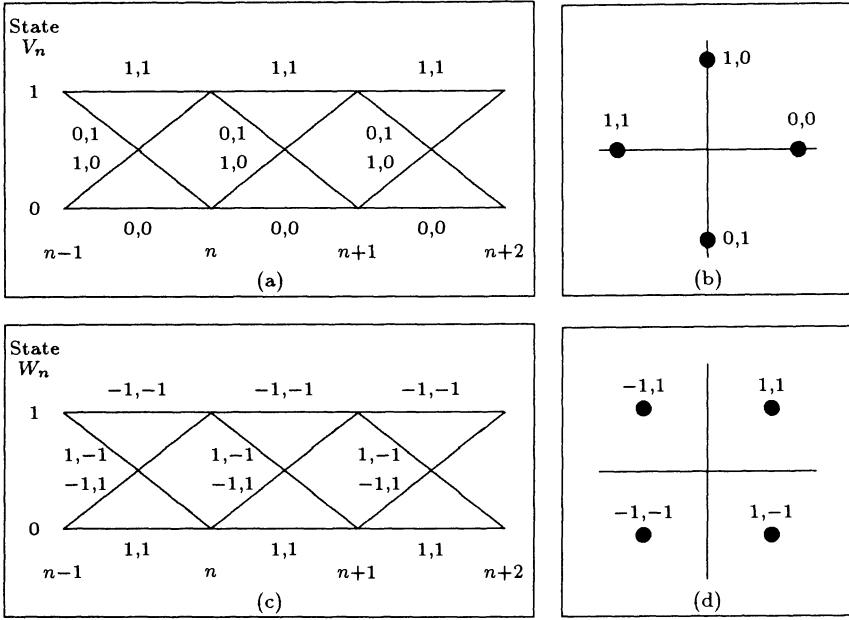


Figure 7: Diversity implementation of DMSK: alternative derivation. Transitions in (a) are labeled with  $V_{n+1}, V_n$ , whereas in (b) they are labeled with  $W_{n+1}, W_n$ .

in Figure 4(b). This provides an alternative way to obtain the diversity interpretation from the CPFSK interpretation.

## Appendix

In this Appendix we derive (11) and (12). For  $nT \leq t < (n+1)T$ ,

$$S_1(t - nT) = \sqrt{\frac{2E}{T}} \cos 2\pi f_1(t - nT) = \pm \sqrt{\frac{2E}{T}} \cos 2\pi f_1 t, \quad (14)$$

where the sign is positive for  $n$  even and negative for  $n$  odd, whereas

$$S_0(t - nT) = \sqrt{\frac{2E}{T}} \cos 2\pi f_0(t - nT) = \sqrt{\frac{2E}{T}} \cos 2\pi f_0 t \quad (15)$$

for all values of  $n$ .

Assuming  $n$  even, using (15) and (14), we obtain

$$\begin{aligned} S(t - (n-1)T) &= S_0(t - (n-1)T) - S_1(t - (n-1)T) + S_0(t - nT) + S_1(t - nT) \\ &= \sqrt{\frac{2E}{T}} [\cos 2\pi f_0 t + \cos 2\pi f_1 t], \quad (n-1)T \leq t < (n+1)T \end{aligned} \quad (17)$$

$$= \sqrt{\frac{8E}{T}} \cos 2\pi f_0 t \cos \frac{\pi t}{2T}, \quad (n-1)T \leq t < (n+1)T \quad (18)$$

where  $f := f_0 + \frac{1}{4T}$ . Similarly, replacing  $n$  with  $n + 1$ , we obtain

$$S(t - nT) = \sqrt{\frac{8E}{T}} \sin 2\pi ft \sin \frac{\pi t}{2T} \quad nT \leq t < (n+2)T. \quad (19)$$

## References

- [1] R. de Buda, “A coherent demodulation of frequency-shift keying with low deviation ratio,” *IEEE Transactions on Communications*, vol. 20, pp. 429–435, June 1972.
- [2] F. Amoroso and J. A. Kivett, “Simplified MSK signaling technique,” *IEEE Transactions on Communications*, vol. 25, pp. 433–441, April 1977.
- [3] J. L. Massey, “A generalized formulation of minimum shift keying modulation,” in *Proceedings IEEE Int. Commun. Conf.*, (Seattle, WA), pp. 26.5.1–26.5.4, June 1980.
- [4] B. Rimoldi, “A decomposition approach to CPM,” *IEEE Transactions on Information Theory*, vol. 34, pp. 260–270, March 1988.
- [5] G. D. Forney, Jr., “The Viterbi algorithm,” *IEEE Proc.*, vol. 61, pp. 268–278, March 1973.

# **Leaf-Average Node-Sum Interchanges in Rooted Trees with Applications**

Rainer A. Rueppel  
 $\mathcal{R}^3$  Security Engineering AG  
Swiss Federal Institute of Technology  
CH-8607 Aathal,  
Switzerland

James L. Massey  
 $\mathcal{R}^3$  Security Engineering AG  
Swiss Federal Institute of Technology  
CH-8092 Zurich  
Switzerland

## **Abstract**

This paper is divided into an anecdotal section and a technical section. The anecdotal section tells how it all began in Zurich seen with the eyes of the first research assistant that joined Jim Massey in 1980. The technical section contains one of the results that Jim and I had wanted to publish jointly for many years, but never had managed to do so: the LANSIT (Leaf Average Node Sum Interchange Theorem). More than ten years have passed since we were working on these ideas.

## **I Prologue**

In the spring of 1980 three young research assistants were called into the office of George Moschytz who was the head of the Institute for Telecommunications (now called Institute for Signal and Information Processing) at the ETH in Zurich (their names were Godi Fischer, Markus Thaler, and Rainer Rueppel). George Moschytz told us that a new professor from the U.S., James L. Massey, would join our Institute. We had never heard that name before, which was not surprising, since we were about to specialize in topics such as filter design and signal processing. George Moschytz suggested more specifically that one of us should sacrifice his current research field to join the new professor. We asked why he did not simply employ a new research assistant for that purpose. His answer was that he would like an experienced research assistant to make Jim Massey's start at the ETH as smooth as possible. This statement seems to somehow imply that it is pretty complicated for a foreign professor to find his way at the ETH. Naturally, we asked what the preferred topics of the new man were. The answer was coding and information theory, which did not help either of us to reach a decision. We did not have any exposure to these topics before. Also naturally, we asked if we could talk to the new man before committing our scientific future. The answer was negative, since winter semester had to be prepared and arrangements had to be made before the arrival of the new professor. As the reader may already have guessed, it was I who finally took the risk to join Jim Massey without even having talked to him before (I tried to convince myself with the reasoning that if all went wrong I would at least

have improved my English). And so it happened that my first attempt at getting a Ph.D. (in digital filters and signal processing) had to be abandoned. But even worse, the new man was to arrive shortly before the winter semester started and I was due for my military service just around the same time. In addition, there did not yet exist any course notes. Hence, when I came back from the military service, I knew less about information theory than the students who had already had two weeks of lectures. It became my first exposure to the American principle that the best way to learn a subject was to teach it. The course was called Applied Digital Information Theory. Jim Massey deeply impressed me through his personality and I immediately enjoyed working with him. At that time (1980) English was not (and probably still is not) an official teaching language (the ETH required that courses had to be delivered in one of the official teaching languages German, French, or Italian. Since Jim Massey did not know any of these languages he was given permission to teach in English during a one-year transitory period during which he was supposed to learn German. Until 1984 when I left the Institute, he was still teaching in English. And as far as ADIT is concerned, the one-year exceptional permission seems to have extended to more than 13 years by now. Of course, Jim made several attempts to deliver the lectures in German. But the students protested in general, since his English lectures were much more vivid and enjoyable. Hence, these attempts never lasted. To his honor it must be said that in 1987 Jim Massey began to teach an undergraduate course in German, called Mathematische Grundlagen der Nachrichtentechnik.

In the beginning Jim Massey was particularly interested in random access communication. There was a gap between the best upper bound on the capacity of the random access channel and the throughput of algorithms. So we defined a research project called Data Networks whose goal was to find the capacity of the random access channel or, at least, to improve the upper bound. Initially, since other obligations were slow to take off, Jim Massey had quite some time to supervise me. Every week he used to call me into his office and asked me what the new results were. And when I said that I did not have any, he simply told me that I was not working hard enough. He then, together with me, went painstakingly through all the little things that I had done during the week. This process began to worry me so much that I frantically was searching for new ideas whenever Friday came close.

The Swiss Army also played a key role in my scientific development. After my basic training I was certified to be a 'Microwave Pioneer' (translated directly). The title sounds much more interesting than the work. Since I had officially complained about the work, I was given the opportunity to do my military duties in the Cipher Section in Bern in 1982. The Cipher Section was (and still is) run by Dr. Peter Nyffele. We had some interest in shift registers and nonlinear operations (either applied to combine the shift registers or as a nonlinear feedback). So I asked Jim if he could provide me with some articles on shift registers for my three weeks of mandatory service. And I went with a big case of old reports and papers. The three weeks were quite satisfactory for me; results were almost immediate. And I got infatuated with cryptology. When I came back after the military service I had lost all interest in random access communication. Considering the new interesting world of cryptology, the problem of the random access channel seemed somehow unimportant. I continued to work on shift registers and their nonlinear combinations. Of course, Jim noticed my new ambitions. For several weeks, he tried to force me back onto the old problem

arguing “you have basically done research and you should write up your results.” After an intense row, we settled on a compromise: the Ph.D. thesis was to be called “Selected topics in Data Communication and Cryptography” and was to contain the previous results on random access communications combined with the new results on stream ciphers. As many of you know, the final result was a Ph.D thesis “New approaches to stream ciphers” and did not contain any results on the random access channel. It might be consoling to many students to hear that I succeeded only with my third attempt at writing a Ph.D. thesis. Jim accepted the situation without great difficulty because one of his principles was that research should be fun. Looking back, the quick and unreflected decision that I had made as a young research assistant very likely was one of the luckiest in my life.

This paper will state one of the results that Jim and I had wanted to publish jointly for many years but never had managed to do so: the LANSIT (Leaf Average Node Sum Interchange Theorem). More than 10 years have passed since we were working on these ideas. In addition, since Jim did not have a chance to look at the final paper, any errors encountered are to be attributed to the second author.

## II Introduction to the Technical Part

### Rooted Trees with Probabilities

A directed graph is a set of vertices and a set of branches, where each branch can be represented by an ordered pair of vertices. We say that a vertex  $v$  in a directed graph is a predecessor of the vertex  $v'$ , if there exists a directed branch from  $v$  to  $v'$ . A *general rooted tree* is a directed graph with a specified vertex (designated as the root) such that:

1. Each vertex except the root has exactly one predecessor.
2. The root has no predecessor.
3. Each vertex has at most countably many successors.
4. There is a directed path from the root to each other vertex.

We have included the adjective “general” to emphasize that we are not assuming the graph to be finite as is often tacitly done in definitions of a rooted tree. Note that the definition implies that the directed path from the root to each other vertex will be unique. The root together with the intermediate vertices will be called nodes. Those vertices where the tree terminates are called leaves. Let  $N$  denote the set of all nodes and  $L$  the set of all leaves. The depth of a vertex is its distance from the root measured in branches. The  $i$ th step in a path starting at the root is defined to be the transition from the node in depth  $i - 1$  to its successor in depth  $i$ .

A general rooted tree has a countable number of vertices. This can be shown by construction. Suppose the vertices in the general rooted tree are labeled, starting from the root by applying recursively the following rule: “Label a new vertex at a new depth, then return to the previously labeled nodes and label a new successor (if necessary) for each of them.” This labeling procedure establishes an equivalence relation between the vertices and the integers, which completes the proof.

Now let us assign to the vertices of the tree a real-valued function  $q$  which is defined by the following properties:

1. The function value at any vertex,  $q(i)$  for  $i \in N, L$ , is a nonnegative real number.
2. The sum of the function values over all leaves equals one.

$$\sum_{i \in L} q(i) = 1$$

3. The function value of a node  $q(i)$ ,  $i \in N$ , is equal to the sum of the function values of all leaves that belong to the subtree rooted at that node, and is zero (except for the root) if there are no such leaves.

Note that the function value at the root is always one and that the function  $q$  is nonincreasing with the depth of the tree. The vertex countability theorem allows us now to interpret the function  $q$  as a discrete probability measure over the vertices. Then  $q(i)$  for  $i \in L$ , the function value of the leaf  $i$ , is considered to be the probability that a path through the tree will terminate at that leaf and we will write  $p_1, p_2, p_3, \dots$  to denote the probabilities of the leaves. The function value of the  $j$ th node,  $q(j)$  for  $j \in N$ , now is considered to be a measure of the relative frequency with which this node is traversed on a path from the root to a leaf and we shall write  $P_1, P_2, P_3, \dots$  to denote the probabilities of the nodes. Without loss of generality we may assume the nodes to have nonzero probability, since by property (3) of  $q$  the subtree stemming from a node with zero-probability would incorporate only zero probability vertices, and thus could be deleted without affecting the statistical interpretation of the tree.

A general rooted tree together with a discrete probability measure  $q$  defined in the above way is called a *general rooted tree with probabilities*.

### III The Leaf-Average Node-Sum Interchange Theorem

The second function  $f$  that shall be assigned to the vertices of the tree is real-valued, but otherwise completely general. This function can be interpreted as a potential attached to each vertex. One quantity that will prove to be of particular interest is the expected function gain  $E[\Delta f(i)]$  from node  $i$  to its successors. Let  $R(i)$  be the set of all vertices that are successors of node  $i$  and let  $q(i, k)$  denote the probability associated to the  $k$ th successor of node  $i$ . We write  $q$  here to denote that the successor could be either a leaf or a node. Note that by Property 3 of  $q$  the following relation holds,

$$\sum_{k \in R(i)} q(i, k) = P_i \tag{1}$$

By normalizing with  $P_i$  we obtain the conditional probability distribution in node  $i$ . Let

$$q(k|i) = \frac{q(i, k)}{P_i} \tag{2}$$

denote the probability that the  $k$ th successor will be selected given that node  $i$  is reached. Then the expected function gain  $E[\Delta f(i)]$  from node  $i$  to its successors is defined by

$$E[\Delta f(i)] = \sum_{k \in R(i)} q(k|i)[f(k) - f(i)] \quad (3)$$

This quantity clearly belongs to node  $i$ , because after having proceeded to one of the successors of node  $i$  the exact function gain is known. Now we are ready to prove a useful theorem.

### Theorem 1: Leaf-Average Node-Sum Interchange Theorem (LANSIT)

*If in a rooted tree with probabilities to each vertex a real-valued function  $f$  is assigned, then the average of  $f$  over the leaves is equal to the function value at the root plus the sum of the expected function gains of all nodes,*

$$\sum_{j \in L} f(j)p_j = f(1) + \sum_{i \in N} E[\Delta f(i)]P_i \quad (4)$$

**Proof:**

$$\begin{aligned} E[\Delta f(i)] &= \sum_{k \in R(i)} q(k|i)[f(k) - f(i)] \\ &= -f(i) + \sum_{k \in R(i)} q(k|i)f(k) \\ &= -f(i) + \sum_{k \in R(i)} \frac{q(k)}{P_i} f(k) \end{aligned}$$

Multiplying with  $P_i$  and summing over all nodes  $i$  yields

$$\sum_{i \in N} P_i E[\Delta f(i)] = -\sum_{i \in N} P_i f(i) + \sum_{i \in N} \sum_{k \in R(i)} q(k)f(k)$$

Note that the double sum over all successors of all nodes can be replaced by the sum over all vertices except the root node, or equivalently by the sum over all nodes except the root plus the sum over all leaves. Thus

$$\begin{aligned} \sum_{i \in N} P_i E[\Delta f(i)] &= -\sum_{i \in N} P_i f(i) + \sum_{i \in N} P_i f(i) - P_1 f(1) + \sum_{i \in L} p_i f(i) \\ &= -f(1) + \sum_{i \in L} p_i f(i) \end{aligned}$$

which proves the theorem. But note that the proof made use of the fact that the sum (series)  $\sum_{i \in L} p_i f(i)$  is rearrangeable. A sufficient condition for the LANSIT to hold is that the node sum  $\sum_{i \in N} E[\Delta f(i)]P_i$  is absolutely convergent.

## IV Applications of the LANSIT

This section illustrates applications of the LANSIT. Any phenomenon that can conceptually be thought of as having an underlying tree structure is amenable to the analysis presented in this paper.

### Path Length Lemma

A first immediate consequence of the LANSIT is the well-known path length lemma which relates the average depth  $E[W]$  of the leaves with the node probabilities.

#### Corollary 1: Path Length Lemma

*In a rooted tree with probabilities the average depth  $E[W]$  of the leaves is equal to the sum of the probabilities of the nodes.*

Proof: Let the function of the  $i$ th vertex be its depth. The depth of the root node is 0 by definition and the function gain between a node and its successors is always 1. Under these conditions the LANSIT states that

$$E[W] = \sum_{j \in L} f(j)p_j = \sum_{i \in N} P_i \quad (5)$$

which proves the path length lemma.

### Bounding the Average Path Length

Often it is too tedious or even impossible to determine  $E[W]$  directly over the leaves or indirectly using the path length lemma. But nevertheless the LANSIT does also provide an efficient tool for bounding the average leaf-depth of rooted trees with probabilities.

#### Corollary 2: Bounding the Average Path Length

*If the function  $f$  that is assigned to the vertices of a rooted tree with probabilities has a known leaf-average  $E[f_L]$ , and if*

$$\Delta_u = \sup_i E[\Delta f(i)] \quad \text{for all nodes } i \in N \quad (6)$$

and

$$\Delta_l = \inf_i E[\Delta f(i)] \quad \text{for all nodes } i \in N \quad (7)$$

then the average leaf-depth  $E[W]$  is bound by

$$\frac{E[f_L] - f(1)}{\Delta_u} \leq E[W] \leq \frac{E[f_L] - f(1)}{\Delta_l} \quad (8)$$

**Proof:** Starting with the LANSIT

$$E[f_L] = \sum_{j \in L} f(j)p_j = f(1) + \sum_{i \in N} E[\Delta f(i)]P_i$$

and substituting (6) yields the inequality

$$E[f_L] \geq f(1) + \Delta_u \sum_{i \in N} P_i$$

The path length lemma now proves the left side of (8). Substituting (7) instead of (6) into the LANSIT then proves together with the path length lemma the right side of (8) and thus the corollary.

Corollary 2 allows us, for example, to bound the actual performance of a given testing algorithm after only a superficial analysis of its properties; it also allows us, as we shall see in the information-theoretic application, to make meaningful statements about the average codeword length of a prefix-free source code.

### Lemma on the Sum of a Variable Number of Random Variables

Now let us derive a result from the theory of stochastic processes, there known as Wald's inequality, but which we want to give the more suggestive name 'Lemma on the sum of a variable number of random variables'.

### Corollary 3: Lemma on the Sum of a Variable Number of Random Variables

*If  $f$  denotes the sum of  $W$  real-valued random variables  $\Delta f_i$ , e.g.  $f = \Delta f_1 + \Delta f_2 + \dots + \Delta f_W$ , where  $W$  is a positive-integer-valued random variable, then*

$$E[f] = \sum_{i=1}^{\infty} E[\Delta f_i | W \geq i] P(W \geq i) \quad (9)$$

Moreover, if for all  $i = 1, \dots, W$

$$E[\Delta f_i | W \geq i] \leq c \quad (10)$$

this implies that

$$E[f] \leq cE[W] \quad (11)$$

**Proof:** Before we can apply the LANSIT, we have to give a justification that the above random variables can be properly described in a rooted tree with probabilities. Suppose  $\Delta f_i$  denotes the function gain in step  $i$ . Clearly this is a discrete, real-valued random variable with a probability distribution that is induced by the probability assignment to the nodes in depth  $i - 1$  and to the vertices in depth  $i$ . In general  $\Delta f_i$  depends on the values of the previous random variables  $\Delta f_1, \dots, \Delta f_{i-1}$ . But by requiring that the conditional probability distribution of all nodes in the same depth be equal for all depths, we can make the  $\Delta f_i, i = 1, \dots, W$ , independent of each other. Suppose further that  $W$  counts the number of steps until a leaf is reached. Its probability distribution is induced by the

probabilities of the leaves. Thus the expected sum of the random variables  $\Delta f_i$  corresponds to the leaf-average of  $f$ . Now define  $N_i$  to be the set of nodes in depth  $i$ . Note that

$$\sum_{k \in N_{i-1}} p_k = P(W \geq i)$$

The LANSIT ( $f(1)$  is set to zero in this interpretation) yields

$$\begin{aligned} \sum_{j \in L} f(j)p_j &= \sum_{i \in N} E[\Delta f(i)]P_i \\ &= \sum_{i=1}^{\infty} \sum_{k \in N_{i-1}} E[\Delta f(k)]P_k \end{aligned}$$

Multiplication and division by  $P(W \geq i)$  results in

$$E[f] = \sum_{i=1}^{\infty} E[\Delta f_i | W \geq i]P(W \geq i)$$

which proves the main part of the corollary.

Suppose now that the expectation of the  $i$ th random variable, provided that it exists in the sum, can be bounded for all  $i = 1, \dots, W$ , that is,

$$E[\Delta f_i | W \geq i] \leq c$$

Substituting this into (9) yields the inequality

$$E[f] \leq c \sum_{i=1}^{\infty} P(W \geq i)$$

The sum could again be decomposed into the sum of the  $P_i$  over all nodes  $i$ , which in turn equals  $E[W]$  by the path length lemma, and thus proves the inequality (11) of the corollary.

Equivalently we might replace the upperbound by a lowerbound or by an equality, if that is desirable. Note that the LANSIT is in fact a generalization of the lemma on the sum of a variable number of random variables, since for each sum of random variables there exist a variety of rooted trees modeling the required probability distribution.

## Leaf-Entropy Theorem

Of special interest is the impact of the LANSIT in information theory. Nothing prevents us from choosing the vertex function to be the negative logarithm of the probability attached to each vertex of the rooted tree, that is  $f(i) = -\log q(i)$ ,  $i \in L, N$ . Then the leaf-average  $E[f_L]$  of  $f$  is exactly the uncertainty about at what leaf a path through the tree will terminate. Let us define this uncertainty to be the leaf-entropy  $H_L$ .

$$H_L = - \sum p_i \log p_i \tag{12}$$

The leaf-entropy  $H_L$  can equivalently be viewed as the uncertainty about which path through the tree will be taken, since each vertex has only one predecessor and thus each leaf

uniquely determines a path. Let us now evaluate the expected function gain  $E[\Delta f(i)]$  from node  $i$  to its successors, as defined in (3) with the chosen vertex function  $f(i) = -\log q(i)$ .

$$\begin{aligned} E[\Delta f(i)] &= - \sum_{k \in R(i)} q(k|i) [\log q(k) - \log P_i] \\ &= - \sum_{k \in R(i)} q(k|i) \log \frac{q(k)}{p_i} \\ &= - \sum_{k \in R(i)} q(k|i) \log q(k|i) \\ &= \Delta H(i) \text{ for } i \in N \end{aligned}$$

$\Delta H(i)$  is just the uncertainty about which branch will be taken next given that node  $i$  is reached. Now the leaf-entropy theorem as stated in [6] follows as a special case of the LANSIT and needs no proving.

### Theorem 2: Leaf-Entropy Theorem

*The leaf-entropy  $H_L$  in a rooted tree with probabilities equals the sum of the node branching entropies weighted by the node probabilities.*

$$H_L = \sum_{j \in N} p_j \Delta H(j) \quad (13)$$

If we define  $\Delta H_{max}, \Delta H_{min}$  to be

$$\Delta H_{max} = \sup_i \Delta H(i) \quad \text{for all } i \in N \quad (14)$$

$$\Delta H_{min} = \inf_i \Delta H(i) \quad \text{for all } i \in N \quad (15)$$

and apply corollary 2, we are able to bound the average leaf-depth in terms of entropies.

$$\frac{H_L}{H_{max}} \leq E[W] \leq \frac{H_L}{H_{min}} \quad (16)$$

Let us now interpret the rooted tree with probabilities as defined by a D-ary prefix-free encoding of a discrete random variable  $U$ , where each leaf corresponds to a particular value of  $U$  and the path to that leaf corresponds to the unique D-ary codeword associated to that particular value of  $U$ . Then (16) implies one of the fundamental source coding results, namely, that the average number of symbols,  $E[W]$ , to encode the discrete random variable  $U$  into D-ary prefix-free codewords satisfies

$$E[W] \geq \frac{H(U)}{\log D} \quad (17)$$

This follows trivially from the fact that in a D-ary rooted tree with probabilities the branching entropy is bounded by

$$H_{max} \leq \log D$$

Note that this derivation completely bypasses the Kraft inequality and could thus be regarded as a more fundamental proof.

## The Associated Markov Chain of a General Rooted Tree with Probabilities

Suppose the general rooted tree with probabilities is modified such that all the leaves and the root node are clustered together to form a new supernode. The modification does not affect the node probabilities because the sum of all leaf probabilities is equal to the root probability. Interpreting now the nodes as states and the conditional branching probabilities  $q(k|i)$  as stationary transition probabilities then the modified directed graph defines a homogeneous Markov chain which we will call the associated Markov chain of the general rooted tree with probabilities. The following lemma allows us to make very elegant use of the results in Markov theory when investigating the properties of phenomena that have a rooted tree with probabilities as underlying structure.

**Lemma 1** *The Markov chain associated with a general rooted tree with probabilities always has a stationary probability distribution  $\underline{P}' = [P'_1, P'_2, \dots]$ , where  $P'_i$  denotes the node probability  $P_i$  divided by the average leaf depth  $E[W]$ . Moreover,  $\underline{P}'$  coincides with the steady state probabilities if the Markov chain is aperiodic.*

**Proof:** Each state is reachable from every other state in at most countably many steps, since in the original rooted tree with probabilities there is by definition a directed path from the root to each other vertex, and each node with nonzero probability has at least one leaf in the subtree stemming from that node. Thus the associated Markov chain is irreducible. Provided that the average depth of the leaves  $E[W]$  is finite in the original rooted tree with probabilities, the average recurrence time for any state in the associated Markov chain must also be finite. It is well known in Markov theory, that for an irreducible and positive recurrent Markov chain there always exists a unique stationary (or invariant) probability distribution  $\underline{P}' = [P'_1, P'_2, \dots]$ , such that each component  $P'_i$  is a nonnegative real number, the sum of all components of  $\underline{P}'$  equals 1, and  $\underline{P}'$  multiplied with the transition matrix  $Q$  of the Markov chain yields again  $\underline{P}'$ . We will show now that the node probabilities in the original rooted tree with probabilities divided by  $E[W]$  satisfy these conditions, and thus must be the unique stationary probability distribution for the associated Markov chain. By definition of the conditional branching probabilities in the original rooted tree with probabilities each vertex except the root has an associated probability equal to the product of the probability of its sole predecessor multiplied with the corresponding branching probability. The supernode in the modified directed graph has as many incoming branches as there are leaves in the general rooted tree with probabilities, and its probability equals the sum of the leaf probabilities. Thus each of these node equations can be viewed as defining one column in the transition matrix  $Q$  of the associated Markov chain, and the original node probability vector  $\underline{P} = [P_1, P_2, \dots]$  provides a solution of the equation  $\underline{P} \cdot Q = \underline{P}$ . By the path length lemma the sum of all node probabilities in the original rooted tree with probabilities equals the average leaf depth  $E[W]$ . Hence normalizing the  $P_i$  by  $E[W]$  yields the unique stationary probability distribution for the associated Markov chain. Note that this stationary probability distribution coincides with the steady state probabilities if and only if the Markov chain is aperiodic, and thus ergodic. But if the Markov chain is periodic, that is, if  $\gcd(w_1, w_2, \dots) = d > 1$ , where  $w_i$  denotes the depth of leaf  $i$  in the original rooted tree with probabilities, then the stationary probability distribution defines the time average of the state occupancy. In this case the steady state probabilities no longer exist, since the

probability of reaching state  $i$  from state  $j$  in  $n$  steps oscillates between some positive real number for  $n = a + kd$ , where  $a$  denotes a fixed integer with  $0 \leq a < d$ , and zero otherwise.

## V Random Access Algorithms

Consider the idealized random-access situation in which an infinite number of stations try to communicate over a common noiseless broadcast channel. The transmissions on the channel are synchronized to fall into constant-length time intervals, called *slots*. For transmitting a message exactly one slot is needed. The stations have no possibilities to exchange control information over some private communication channel. Thus simultaneous transmissions of more than one message in a slot are possible and result in a collision. All the messages involved in a collision are completely destroyed and must be retransmitted successfully in some later slot to ensure reliable communications. The only means for coordinating their transmissions is the channel feedback information which tells all the stations immediately at the end of a slot whether zero, one, or more than one message was transmitted during the slot. A slot containing no messages at all is called an empty slot. Only if exactly one station attempts transmission in a slot, this message is assumed to arrive undistorted at the destination. In such a case the outcome of the slot is called a success.

Let the arrival process to the system consist of a series of independent Bernoulli points each having probability  $p$  of containing exactly one message, and  $1 - p$  of containing no message. Note that the Poisson arrival process is also included in the arrival model as special case. For, if  $M$  denotes the number of arrival points in a fixed-length time interval, taking the limit as  $p \rightarrow 0$  and  $M \rightarrow \infty$ , while the product  $\lambda = Mp$  is held constant, results in the Poisson arrival process with parameter  $\lambda$ . An arrival point containing a message is called an active point, whereas an arrival point without message is called an idle point. The active points are thought of as randomly assigned to a station. Because there are infinitely many such stations, the probability that more than one message is assigned to one station in its lifetime is zero. Any distributed protocol responsible for transmitting the messages in an infinite population cannot grant permission to individual stations or subsets of stations, since that would require infinite amounts of addressing. But it can directly grant transmission permission to subsets of arrival points. Thus the task of any such protocol is to choose a sequence of enabled subsets of the arrival time axis such that few slots are wasted as idles or collisions. Let us now precisely define the idealized random-access problem. The input data to the random access algorithm (*RAA*) is a finite interval of the arrival time axis containing  $M$  arrival points. By the independence of the arrival points we may assume that  $[1, M]$  is the interval to be explored. The test available to the *RAA* consists of enabling any subset  $E$  of  $[1, M]$  with the test outcome  $Y$  indicating whether zero ( $Y = 0$ ), one ( $Y = 1$ ), or more than one ( $Y = 2^+$ ) active point was contained in the enabled set. Let  $m$  denote the measure of  $E$ . An active arrival point is defined to be transmitted if it was included in a test with success outcome. Without loss of generality assume that no parts of previously enabled sets which resulted either in idle or success outcomes are again included in any further test. Since at any time of execution the state of knowledge of a *RAA* can be summarized by the previous tests and their outcomes, let these completely specify the choice of the next test to be applied. The *RAA* has accomplished its task when the input data is partitioned into disjoint subsets each of which contains either zero or one

active arrival point. It follows from this setup that the possible executions of a *RAA* can be interpreted by a rooted tree with probabilities, where each node  $i$  corresponds to an enabled set  $E(i) \subset [1, M]$  of size  $m(i)$ , whose outcome  $Y(i)$  indicates which of the three branches leaving node  $i$  has to be taken, and where each leaf  $j$  corresponds to a valid partition of  $[1, M]$ . The interesting open question is to determine the theoretical limit of performance for the whole class of unrestricted ternary random-access tree algorithms.

In 1982, the inability to achieve throughputs in excess of .488 had increased attention on the problem of determining capacity for the ternary random access problem by upperbounding the efficiency of possible such protocols. Pippenger [9] obtained, using a time-average approach and information-theoretic arguments, an upper bound of .744. Humblet [4] and Hajek [3] both tried to tighten that result by lower-bounding more accurately the information provided by a feedback symbol, but both approaches are based on arguments that unfortunately are not valid for general *RAAs* because their bounds implicitly assumed that the average size of an enabled set given that no messages are present equals the average size given that one message is present. Molle [7] devised a simple genie argument that improved the bound to .673. Cruz and Hajek [2] modified Molle's genie and, implicitly using an objective function approach, found a bound of .613. Tsybakov and Mikhailov [11] established an upper bound of .588, explicitly using an objective function approach.

It can be shown that in fact all the mentioned valid upper bounds can, despite their apparent discrepancy, be derived using the LANSIT. The key is to choose the right vertex function, such that a lowerbound on the lead-average and an upperbound on the expected function gain can be computed. A natural choice for the vertex function is, for instance, one that reflects the information gained, the number of successes and the number of processed arrival points. Since the field has developed rapidly in the last years, we refrain at this place from demonstrating the unified derivation of all the upper bounds.

## Acknowledgment

We are grateful to Thomas Mittelholzer, whose comments helped to improve the paper.

## References

- [1] J. I. Capetanakis, "Tree algorithms for packet broadcast channels", *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505-515, 1979.
- [2] R. Cruz and B. Hajek, "A new upper bound to the throughput of a multi-access broadcast channel", *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 402-405, 1982.
- [3] B. Hajek, "Information of partitions with applications to random access communications", *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 691-701, 1982.
- [4] P. A. Humblet, "Bounds on the utilization of ALOHA-like multiple-access broadcast channels", M.I.T. Cambridge MA, Rep. LIDS-P-1000, June 1980.
- [5] P. A. Humblet and J. Mosely, "Efficient accessing of a multiaccess channel", M.I.T. Cambridge MA, Rep. LIDS-P-1040, Sept. 1980.

- [6] Massey, J. L., "An Information-Theoretic Approach to Algorithms", Paper presented at the NATO Advanced Study Institute, July 1983.
- [7] M. L. Molle, "On the capacity of infinite population multiple access protocols", *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 396-401, 1982.
- [8] J. Mosely, "An efficient contention resolution algorithm for multiple access channels", M.S. thesis, MIT, Tech. Rep. LIDS-TH-918.
- [9] N. Pippenger, "Bounds on the performance of protocols for a multiple-access broadcast channel", *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 145-151, 1981.
- [10] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, 1953.
- [11] P. S. Tsybakov and V. A. Mikhailov, "An upper bound to capacity of random multiple access systems", *Probl. Peredach. Inform.*, vol. 17, no. 1, Jan. 1981.

# On the Performance of Aperiodic Inverse Filter Approximations

Jürg Ruprecht

Swiss Telecom PTT

R&D, Mobile Communications / FE 422

3000 Berne 29, Switzerland

## Abstract

In 1989, the author proposed the aperiodic inverse filter rather than the matched filter for multipath channel estimation. Thereafter, Massey specified a CDMA-like multiuser communication scheme called Code Time Division Multiple Access (CTDMA), which applies inverse filtering in the despreader for complete separation of the users. This paper evaluates the performance of implementable inverse filter approximations such as the truncated inverse filter, its improved replica by a modified gradient algorithm and its digital realization when its taps are quantized. These results are needed in order to implement a CTDMA-receiver in digital hardware.

## I Introduction

In my Ph.D. thesis [1], written under the excellent supervision of Jim Massey, we have studied the discrete-time multipath channel estimation scheme according to Figure 1. The aperiodic inverse filter was found to be the best solution to this estimation problem. One result was a channel sounding equipment that has been built at Swiss PTT Telecom [3]. Jim Massey then specified a CDMA-like multiuser communication scheme called *Code Time Division Multiple Access (CTDMA)* applying this filter as despreader [2]. CTDMA is an interesting accessing scheme that combines the advantages of *time division multiple access (TDMA)* and *code division multiple access (CDMA)* and is well tailored to an indoor cellular environment [4].

Aperiodic inverse filters have been implemented in *surface acoustic wave (SAW)* technology, where the taps can take on continuous values [5]. For an implementation in digital hardware, the tap values must be quantized. A first application will be the CTDMA-receiver to be defined and tested in the CLUB and SPOT projects, where Jim Massey and the author are involved. This paper explores the performance of implementable inverse filter approximations with continuous-valued and quantized taps by minimizing the highest correlation sidelobe. Performance bounds and examples for sequences of lengths  $L$  in the range  $3 \leq L \leq 32$  are given.

The principle of multipath channel estimation by the use of inverse filtering is summarized in Section II. In Section III, the performance of the truncated aperiodic inverse

filter of Section II is improved by applying a modified gradient algorithm to its tap values. The concept of inverse filter approximations with unbiased equal-step tap quantizations is introduced and their POP-ratio performance is bounded in Section IV. In Section V, this concept is generalized to unequal-step tap quantizations, where the corresponding bounds on the POP-ratio performance are also derived. Finally, Section VI concludes the paper and outlooks to remaining problems and possible improvements.

## II Multipath Channel Estimation

A summary of [1, 3] as far as it is relevant for this paper is given in this section by use of the discrete-time multipath channel estimation model as shown in Figure 1. The channel input sequence is the sequence  $s[.] = \dots s[-1], s[0], s[1] \dots$  that is assumed to be binary and length- $L$  aperiodic, i.e.,  $s[n] \in \{+1, -1\}$  for  $0 \leq n < L$  and  $s[n] = 0$  otherwise. For convenience of notation, we shall also use the corresponding (0,1)-representation where  $\{+1\} \rightarrow \{0\}$  and  $\{-1\} \rightarrow \{1\}$ .

The multipath channel is defined by its impulse response  $h[.]$ . The received sequence  $y[.]$ , which is the channel estimator input signal, is the noiseless channel output sequence  $x[.]$  corrupted by an *additive white Gaussian noise (AWGN)* sequence  $z[.]$ , i.e.,  $y[n] = x[n] + z[n]$ , all  $n$ , where the noise digits  $z[n]$  are statistically independent Gaussian random variables with zero mean and variance  $\frac{N_0}{2}$ .  $N_0$  is the one-sided noise spectral density. As channel estimator  $c[.]$ , the aperiodic inverse filter  $v[.]$  defined by

$$(v * s)[n] = \sum_{l=-\infty}^{\infty} v[l] s[n-l] = \sum_{l=0}^{L-1} v[n-l] s[l] = \begin{cases} 1 & \text{if } n=0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

is proposed<sup>1</sup>, which is the maximum-likelihood (ML) solution to the problem. In theory, the inverse filter completely eliminates the sidelobes that appear in processing by the matched filter  $m[.]$  defined by  $m[n] = s[-n]$ , all  $n$ . This is the reason why matched filtering is not recommended here. The drawback of the inverse filter is its poorer noise performance when compared to the matched filter that maximizes the signal-to-noise ratio  $SNR$  at the filter output or, equivalently, maximizes the *processing gain*

$$G^{(cs)} = \frac{\text{filter output SNR}}{\text{filter input SNR}} = \frac{|(c * s)[0]|^2}{\sum_n |c[n]|^2} \quad (2)$$

of any filter  $c[.]$  for a given sequence  $s[.]$ . However, there exist suitable binary sequences where the *noise enhancement factor*

$$\kappa^{(vs)} = \frac{G^{(ms)}}{G^{(vs)}} = \frac{L}{G^{(vs)}} = L \sum_n |v[n]|^2 \quad (3)$$

of the inverse filter  $v[.]$  is only in the order of 1dB (cf. Table 1 and Figure 1a).

As can be easily seen from Equation (1), the inverse filter  $v[.]$  has nonzero coefficients  $v[n]$  in the whole range  $-\infty < n < \infty$ , and thus the filter has no causal equivalent with finite

---

<sup>1</sup>The variable  $v$  in  $v[.]$  is chosen since “v” is the first letter in “inverse” that is not commonly used as a counter.

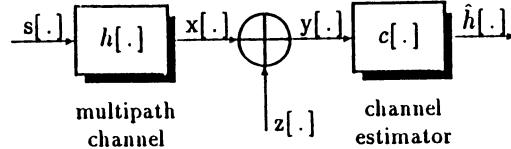


Figure 1: Estimation scheme for the multipath channel with impulse response  $h[\cdot]$ .

| $L$ | $s[\cdot]$        | $G^{(vs)} [dB]$ | $L$ | $s[\cdot]$                       | $G^{(vs)} [dB]$ |
|-----|-------------------|-----------------|-----|----------------------------------|-----------------|
| 3   | 001               | 3.49            | 18  | 001011010001110111               | 11.54           |
| 4   | 0001              | 4.34            | 19  | 0010111010000100001              | 11.98           |
| 5   | 00010             | 6.36            | 20  | 00000101110100111001             | 12.24           |
| 6   | 000010            | 6.08            | 21  | 001111100010001010010            | 12.44           |
| 7   | 0001101           | 6.93            | 22  | 0011111000100010100100           | 12.49           |
| 8   | 00011010          | 7.82            | 23  | 000000011100110110101            | 12.78           |
| 9   | 001101000         | 7.46            | 24  | 000100010001111000101101         | 13.02           |
| 10  | 0011010000        | 8.31            | 25  | 0011100111111010100110110        | 13.29           |
| 11  | 00001100101       | 9.47            | 26  | 00011000111111010101101101       | 13.58           |
| 12  | 000011001010      | 10.10           | 27  | 0001111000100010001001101        | 13.67           |
| 13  | 0000011001010     | 10.93           | 28  | 0011100011111110101001001101     | 13.75           |
| 14  | 00000110010101    | 10.68           | 29  | 00011000111111101010110110010    | 13.93           |
| 15  | 000111011101101   | 11.16           | 30  | 001001101111111000111010110101   | 14.14           |
| 16  | 0010100000110011  | 11.01           | 31  | 0011100111101010110110111110000  | 14.26           |
| 17  | 00100110000101011 | 11.19           | 32  | 00000000111100101101010100110011 | 14.31           |

Table 1: Aperiodic inverse filter processing gains  $G^{(vs)}$  vs. length  $L$  for the best binary sequences  $s[\cdot]$  [1, 6], where the term “best” refers to a maximization of  $G^{(vs)}$  over all binary sequences  $s[\cdot]$  of length  $L$ .

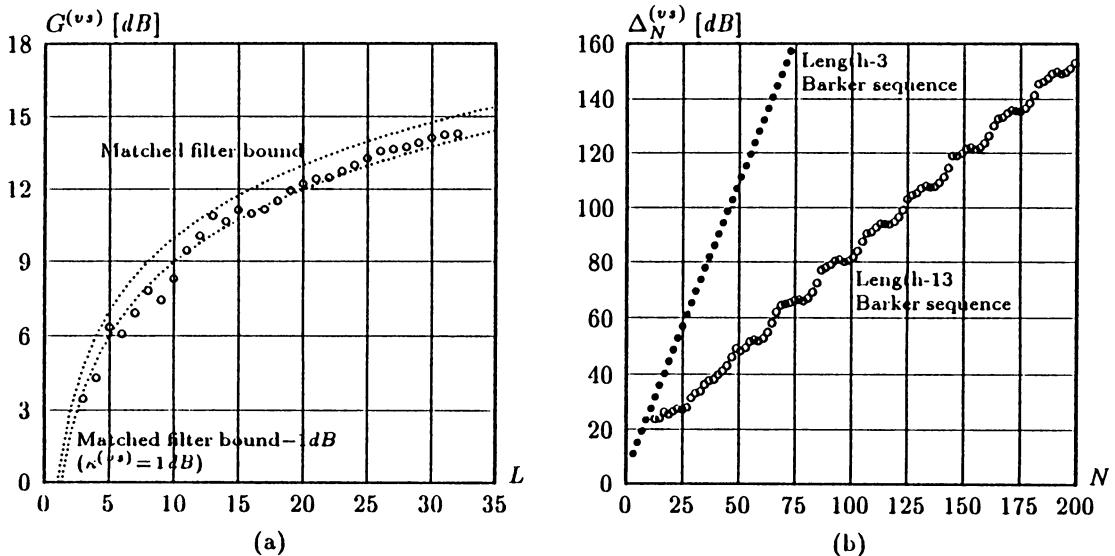


Figure 2: (a) Aperiodic inverse filter processing gains  $G^{(vs)}$  vs. length  $L$  for the best binary sequences  $s[\cdot]$  of Table 1, and (b) POP-ratios  $\Delta_N^{(vs)}$  vs. length  $N$  of the truncated aperiodic inverse filter for the length-3 and length-13 Barker sequences that are also listed in Table 1.

estimation delay. However, for inverse filters with small  $\kappa^{(vs)}$ , the values  $|v[n]|$  decrease exponentially for large  $|n|$  and therefore  $v[.]$  can be truncated to a causal filter  $v_N[.]$  of length  $N$  with a finite estimation delay that is virtually an ML-estimator of the multipath channel, i.e., with a non-infinite but large enough *peak/off-peak ratio (POP-ratio)* defined by

$$\Delta_N^{(vs)} = \frac{(v_N * s)[0]}{\max_{n \neq 0} |(v_N * s)[n]|} \quad (4)$$

(cf. Figure 1b). In this paper, we shall usually consider symmetric truncations where  $v[.]$  is truncated to  $v_N[.]$  with the filter length  $N \equiv L \pmod{2}$  according to

$$v_N[n] = \begin{cases} v[n] & -\frac{N+L}{2} < n \leq \frac{N-L}{2} \\ 0 & \text{otherwise.} \end{cases}$$

### III Improvement of the POP-Ratio Performance

The POP-ratio performance  $\Delta_N^{(vs)}$  of the truncated inverse filter  $v_N[.]$  can be improved because truncation is a simple but in general not an optimum method for this purpose. In this section, some properties and approximations on the optimum filter  $p_N[.]$  are derived<sup>2</sup> that maximizes the POP-ratio  $\Delta_N^{(ps)}$  of  $(p_N * s)[.]$  for a given filter length  $N$ . First we observe the following

*Monotony Property of  $p_N[.]$ :* The optimum filters  $p_N[.]$  and  $p_{N+1}[.]$  of lengths  $N$  and  $N+1$ , which maximize  $\Delta_N^{(ps)}$  and  $\Delta_{N+1}^{(ps)}$  for all filters of the same lengths  $N$  and  $N+1$ , yield  $\Delta_N^{(ps)} \leq \Delta_{N+1}^{(ps)}$ .  $\square$

*Proof:* Given  $p_N[.]$ , we can always construct a filter  $\bar{p}_{N+1}[.]$  of length  $N+1$  with  $\bar{p}_{N+1}[n] = p_N[n]$  (all  $n$ ) with one (additional) tap value of zero within the filter length  $N+1$  so that  $\Delta_{N+1}^{(\bar{p}s)} = \Delta_N^{(ps)}$ . A longer optimum filter can therefore not decrease the POP-ratio performance.  $\square$

An approximation  $\tilde{p}_N[.]$  on the optimum filter  $p_N[.]$  can be obtained by applying a modified gradient algorithm which uses  $v_N[.]$  as an initial value for  $\tilde{p}_N[.]$ . The POP-ratio performances  $\Delta_N^{(vs)}$  of the truncated inverse filter and  $\Delta_N^{(\tilde{p}s)}$  of the improved inverse filter approximation  $\tilde{p}_N[.]$  are compared in Figure 2 for the length-3 and length-13 Barker sequences, respectively. We observe a gain in POP-ratio of up to several dB for  $\tilde{p}_N[.]$  when compared to  $v_N[.]$ .

### IV Unbiased Equal-Step Tap Quantizations

For a digital implementation, we explore an inverse filter approximation  $e_{N,Q}[.]$  of length  $N$  with its taps quantized to  $Q$  unbiased levels of equal step size<sup>3</sup> yielding a maximum POP-ratio  $\Delta_{N,Q}^{(es)}$  of  $(e_{N,Q} * s)[.]$  for a given filter length  $N$ , where we shall only consider

---

<sup>2</sup>The variable  $p$  in  $p_N[.]$  is chosen since “p” is the first consonant in “optimum”.

<sup>3</sup>The variable  $e$  in  $e_{N,Q}[.]$  is chosen since “e” is the first letter in “equal”.

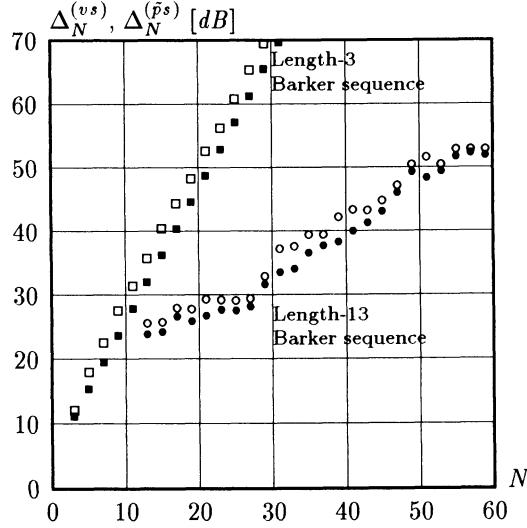


Figure 2: POP-ratios  $\Delta_N^{(vs)}$  ( $\blacksquare, \bullet$ ) and  $\Delta_N^{(ps)}$  ( $\square, \circ$ ) vs. length  $N$  for the length-3 and length-13 Barker sequences, which are both listed in Table 1.

values  $Q \geq 3$  ( $Q=3$  corresponds to tap values  $-1, 0, +1$ ). In general, we shall choose  $e_{N,Q}[\cdot]$  such that  $e_{N,Q}[n] \in \mathcal{S}$ , where the set  $\mathcal{S}$  is given by

$$\mathcal{S} = \begin{cases} \left\{ -\frac{(Q-1)}{2}, -\frac{(Q-1)}{2} + 1, \dots, \frac{(Q-1)}{2} \right\} & \text{if } Q \text{ odd} \\ \left\{ -\frac{Q}{2}, -\frac{Q}{2} + 1, \dots, \frac{Q}{2} - 1 \right\} & \text{if } Q \text{ even and } |\min_n\{p_N[n]\}| \geq |\max_n\{p_N[n]\}| \\ \left\{ -\frac{Q}{2} + 1, -\frac{Q}{2} + 2, \dots, \frac{Q}{2} \right\} & \text{if } Q \text{ even and } |\min_n\{p_N[n]\}| < |\max_n\{p_N[n]\}|. \end{cases} \quad (5)$$

Again, we observe the

*Monotony Property of  $e_{N,Q}[\cdot]$ :* The optimum filters with  $Q$ -level unbiased equal-step tap quantization  $e_{N,Q}[\cdot]$  and  $e_{N+1,Q}[\cdot]$  of lengths  $N$  and  $N+1$ , which maximize  $\Delta_{N,Q}^{(es)}$  and  $\Delta_{N+1,Q}^{(es)}$  for all such filters of the same lengths, yield  $\Delta_{N,Q}^{(es)} \leq \Delta_{N+1,Q}^{(es)}$ .  $\square$

*Proof:* The same proof applies as for the monotony property of  $p_N[\cdot]$ .  $\square$

Next, we derive bounds on the POP-ratio  $\Delta_{N,Q}^{(es)}$  of the optimum filter  $e_{N,Q}[\cdot]$ . For this purpose, we need the following lemmas:

*Lemma 1:* The main peak  $(e_{N,Q} * s)[0]$  is upper bounded by  $(e_{N,Q} * s)[0] \leq L \lfloor \frac{Q}{2} \rfloor$  with equality if and only if  $e_{N,Q}[-l] s[l] = \lfloor \frac{Q}{2} \rfloor$  for all  $l$  in the range  $0 \leq l < L$ .<sup>4</sup>  $\square$

*Proof:* If  $e_{N,Q}[-l] s[l] = \lfloor \frac{Q}{2} \rfloor$ ,  $0 \leq l < L$ , then the main peak  $f_0$  is evaluated by

$$f_0 = (e_{N,Q} * s)[0] = \sum_{l=0}^{L-1} e_{N,Q}[-l] s[l] = L \left\lfloor \frac{Q}{2} \right\rfloor.$$

<sup>4</sup>  $\lfloor \cdot \rfloor$  denotes truncation to the next integer equal or smaller than the argument.

If, for at least one  $l$  in  $0 \leq l < L$ ,  $e_{N,Q}[-l] s[l] \neq \lfloor \frac{Q}{2} \rfloor$ , then  $e_{N,Q}[-l] s[l] < \lfloor \frac{Q}{2} \rfloor$  because  $|e_{N,Q}[-l]| \leq \lfloor \frac{Q}{2} \rfloor$  and  $|s[l]| = 1$ . Therefore, in this case,  $f_0 < L \lfloor \frac{Q}{2} \rfloor$ , which completes the proof.  $\square$

*Lemma 2:* The largest sidelobe of  $(e_{N,Q} * s)[.]$  is lower bounded by  $|(\epsilon_{N,Q} * s)[n]| \geq 1$  for  $n \neq 0$ .  $\square$

*Proof:* Define  $n_-$  such that  $e_{N,Q}[n_-] \neq 0$  and  $e_{N,Q}[l] = 0$  for all  $l < n_-$ , and define accordingly  $n_+$  such that  $e_{N,Q}[n_+] \neq 0$  and  $e_{N,Q}[l] = 0$  for all  $l > n_+$ . Then,

$$\begin{aligned} f_- &= (e_{N,Q} * s)[n_-] = \sum_{l=0}^{L-1} e_{N,Q}[n_- - l] s[l] = e_{N,Q}[n_-] s[0] \neq 0, \\ f_+ &= (e_{N,Q} * s)[n_+ + L - 1] = \sum_{l=0}^{L-1} e_{N,Q}[n_+ + L - 1 - l] s[l] = e_{N,Q}[n_+] s[L-1] \neq 0, \end{aligned}$$

and therefore  $|f_-| \geq 1$  and  $|f_+| \geq 1$ . If  $|f_-| = |f_+| = 1$  and additionally  $|(\epsilon_{N,Q} * s)[n]| \leq 1$  ( $n \neq n_-, 0, n_+$ ), then  $\max_{n \neq 0} |(\epsilon_{N,Q} * s)[n]| = 1$ . If one of these conditions does not hold, then  $\max_{n \neq 0} |(\epsilon_{N,Q} * s)[n]| > 1$ . This completes the proof.  $\square$

*Lemma 3:* The POP-ratio  $\Delta_{N,Q}^{(es)}$  is upper bounded by  $\Delta_{N,Q}^{(es)} \leq L \lfloor \frac{Q}{2} \rfloor$ .  $\square$

*Proof:* The combination of Lemmas 1 and 2 directly proves Lemma 3.  $\square$

*Lemma 4:* The POP-ratio  $\Delta_{N,Q}^{(es)}$  is lower bounded by  $\Delta_{N,Q}^{(es)} \geq \Delta^{(ms)}$ , where  $m[.]$  is the matched filter for  $s[.]$ .  $\square$

*Proof:* Since  $s[.]$  is binary and length- $L$  aperiodic by assumption, the matched filter  $m[.]$  is also binary and length- $L$  aperiodic, i.e.,  $m[n] = \pm 1$  for  $-L < n \leq 0$  and  $m[n] = 0$  otherwise, and  $m[.]$  is therefore a filter with three-level unbiased equal-step tap quantizations. If  $\Delta^{(ms)} > \Delta_{N,Q}^{(es)}$ , then this would contradict with the assumption on  $e_{N,Q}[.]$  (note that we required  $Q \geq 3$  for  $e_{N,Q}[.]$ ).  $\square$

Lemmas 3 and 4 can be combined to get the desired bounds on the POP-ratio  $\Delta_{N,Q}^{(es)}$ :

**POP-Ratio Bounds for Optimum Unbiased Equal-Step Tap Quantization:** The POP-ratio  $\Delta_{N,Q}^{(es)}$  of the optimum filter  $e_{N,Q}[.]$  with  $Q$ -level unbiased equal-step tap quantization is bounded by

$$\epsilon_{N,Q}^{(es)} \leq \Delta_{N,Q}^{(es)} \leq \mathcal{E}_{N,Q}^{(es)},$$

where the lower bound is  $\epsilon_{N,Q}^{(es)} = \Delta^{(ms)}$  and the upper bound is  $\mathcal{E}_{N,Q}^{(es)} = L \lfloor \frac{Q}{2} \rfloor$ .  $\square$

Hereafter, we shall determine an approximation  $\tilde{\epsilon}_{N,Q}[.]$  on  $e_{N,Q}[.]$ , which is obtained by the “Multiply and Round” quantization rule

$$\tilde{\epsilon}_{N,Q}[n] = \lfloor c \tilde{p}_N[n] + 0.5 \rfloor, \quad (6)$$

where

$$c = \min \left\{ \left| \frac{S_{min}}{\min_n \{\tilde{p}_N[n]\}} \right|, \left| \frac{S_{max}}{\max_n \{\tilde{p}_N[n]\}} \right| \right\}$$

| $L$ | (all values in [dB])       |                            |                                    |                            |                                    |                             |                                     |                             |                                     |  |
|-----|----------------------------|----------------------------|------------------------------------|----------------------------|------------------------------------|-----------------------------|-------------------------------------|-----------------------------|-------------------------------------|--|
|     | $\varepsilon_{N,Q}^{(es)}$ | $\mathcal{E}_{N,3}^{(es)}$ | $\Delta_{\infty,3}^{(\tilde{e}s)}$ | $\varepsilon_{N,8}^{(es)}$ | $\Delta_{\infty,8}^{(\tilde{e}s)}$ | $\mathcal{E}_{N,16}^{(es)}$ | $\Delta_{\infty,16}^{(\tilde{e}s)}$ | $\mathcal{E}_{N,32}^{(es)}$ | $\Delta_{\infty,32}^{(\tilde{e}s)}$ |  |
| 3   | 9.5                        | 9.5                        | 9.5                                | 21.6                       | 18.1                               | 27.6                        | 25.1                                | 33.6                        | 31.1                                |  |
| 4   | 12.0                       | 12.0                       | 6.0                                | 24.1                       | 19.1                               | 30.1                        | 26.4                                | 36.1                        | 26.4                                |  |
| 5   | 14.0                       | 14.0                       | 14.0                               | 26.0                       | 21.6                               | 32.0                        | 28.3                                | 38.1                        | 35.1                                |  |
| 6   | 9.5                        | 15.6                       | 9.5                                | 27.6                       | 22.3                               | 33.6                        | 27.6                                | 39.7                        | 33.8                                |  |
| 7   | 16.9                       | 16.9                       | 15.6                               | 28.9                       | 20.4                               | 35.0                        | 26.0                                | 41.0                        | 32.3                                |  |
| 8   | 12.0                       | 18.1                       | 8.0                                | 30.1                       | 24.6                               | 36.1                        | 24.9                                | 42.1                        | 30.6                                |  |
| 9   | 13.1                       | 19.1                       | 10.9                               | 31.1                       | 20.8                               | 37.2                        | 23.7                                | 43.2                        | 23.7                                |  |
| 10  | 14.0                       | 20.0                       | 9.5                                | 32.0                       | 16.5                               | 38.1                        | 27.6                                | 44.1                        | 27.9                                |  |
| 11  | 11.4                       | 20.8                       | 13.1                               | 32.9                       | 20.8                               | 38.9                        | 28.1                                | 44.9                        | 34.7                                |  |
| 12  | 15.6                       | 21.6                       | 14.8                               | 33.6                       | 19.4                               | 39.7                        | 30.0                                | 45.7                        | 33.1                                |  |
| 13  | 22.3                       | 22.3                       | 22.3                               | 34.3                       | 24.6                               | 40.3                        | 31.8                                | 46.4                        | 37.8                                |  |
| 14  | 16.9                       | 22.9                       | 14.0                               | 35.0                       | 19.1                               | 41.0                        | 30.8                                | 47.0                        | 33.6                                |  |
| 15  | 14.0                       | 23.5                       | 16.3                               | 35.6                       | 25.1                               | 41.6                        | 28.7                                | 47.6                        | 38.8                                |  |
| 16  | 18.1                       | 24.1                       | 14.8                               | 36.1                       | 23.5                               | 42.1                        | 28.0                                | 48.2                        | 37.9                                |  |
| 17  | 15.1                       | 24.6                       | 11.3                               | 36.7                       | 22.5                               | 42.7                        | 29.3                                | 48.7                        | 35.3                                |  |
| 18  | 15.6                       | 25.1                       | 11.3                               | 37.2                       | 21.6                               | 43.2                        | 26.9                                | 49.2                        | 36.0                                |  |
| 19  | 16.0                       | 25.6                       | 17.5                               | 37.6                       | 26.9                               | 43.6                        | 30.5                                | 49.7                        | 40.7                                |  |
| 20  | 16.5                       | 26.0                       | 18.1                               | 38.1                       | 20.4                               | 44.1                        | 30.5                                | 50.1                        | 34.6                                |  |
| 21  | 20.4                       | 26.4                       | 13.4                               | 38.5                       | 21.2                               | 44.5                        | 31.2                                | 50.5                        | 34.3                                |  |
| 22  | 17.3                       | 26.9                       | 14.0                               | 38.9                       | 23.1                               | 44.9                        | 30.6                                | 50.9                        | 34.2                                |  |
| 23  | 13.3                       | 27.2                       | 7.4                                | 39.3                       | 21.2                               | 45.3                        | 29.7                                | 51.3                        | 35.7                                |  |
| 24  | 15.6                       | 27.6                       | 15.6                               | 39.7                       | 25.3                               | 45.7                        | 29.4                                | 51.7                        | 35.4                                |  |
| 25  | 14.0                       | 28.0                       | 11.5                               | 40.0                       | 21.4                               | 46.0                        | 28.2                                | 52.0                        | 35.0                                |  |
| 26  | 18.8                       | 28.3                       | 16.0                               | 40.3                       | 25.1                               | 46.4                        | 30.0                                | 52.4                        | 39.2                                |  |
| 27  | 19.1                       | 28.6                       | 16.9                               | 40.7                       | 24.1                               | 46.7                        | 32.7                                | 52.7                        | 36.2                                |  |
| 28  | 22.9                       | 28.9                       | 14.5                               | 41.0                       | 23.8                               | 47.0                        | 29.5                                | 53.0                        | 38.1                                |  |
| 29  | 19.7                       | 29.3                       | 18.8                               | 41.3                       | 27.5                               | 47.3                        | 34.7                                | 53.3                        | 38.4                                |  |
| 30  | 17.5                       | 29.5                       | 14.8                               | 41.6                       | 24.0                               | 47.6                        | 31.4                                | 53.6                        | 38.0                                |  |
| 31  | 17.8                       | 29.8                       | 18.8                               | 41.9                       | 25.0                               | 47.9                        | 32.2                                | 53.9                        | 34.8                                |  |
| 32  | 18.1                       | 30.1                       | 17.7                               | 42.1                       | 24.9                               | 48.2                        | 32.2                                | 54.2                        | 36.9                                |  |

Table 2: The POP-ratios  $\Delta_{\infty,Q}^{(\tilde{e}s)}$  for the best binary sequences  $s[.]$  of length  $L$  in the range  $3 \leq L \leq 32$  (cf. Table 1) together with their bounds for unbiased equal-step tap quantization  $\varepsilon_{N,Q}^{(es)}$  (lower bound) and  $\mathcal{E}_{N,Q}^{(es)}$  (upper bound).

and where  $S_{min}$  and  $S_{max}$  are the smallest and largest elements of the set  $\mathcal{S}$  defined in 5, respectively. Table 2 and Figures 4a and c show the obtained values  $\Delta_{N,Q}^{(\tilde{e}s)}$  for some sequences  $s[.]$ . Table 2 additionally lists  $\varepsilon_{N,Q}^{(es)}$  and  $\mathcal{E}_{N,Q}^{(es)}$ .

In Figures 4a and c, we observe that the POP-ratio  $\Delta_{N,Q}^{(\tilde{e}s)}$  increases with increasing filter length  $N$  only up to a saturation length  $N_{sat}$ . Afterwards, for  $N > N_{sat}$ , the sidelobes due to quantization can no longer be decreased. As indicated by the derived bounds given numerically in Table 2, we notice that  $\tilde{e}_{N,Q}[.]$  is only a good approximation for small values of  $N$  and  $Q$ . For large products  $NQ$ , the bounds anticipate better solutions.

## V Unequal-Step Tap Quantizations

In order to improve the POP-ratio performance of  $e_{N,Q}[.]$ , we are now looking for an inverse filter approximation  $u_{N,Q}[.]$  of length  $N$  with  $Q$ -level tap quantizations of unequal step size<sup>5</sup> yielding a maximum POP-ratio  $\Delta_{N,Q}^{(us)}$  of  $(u_{N,Q}*s)[.]$  for a given filter length  $N$ . Because,

<sup>5</sup>The variable  $u$  in  $u_{N,Q}[.]$  is chosen since “u” is the first letter in “unequal”.

in this case, a tap value of zero is not necessarily included in the  $Q$  quantization levels, we shall allow an additional level of zero for this optimum filter, i.e., we will eventually have  $Q+1$  levels in  $u_{N,Q}[\cdot]$ . Now, we can observe the

*Monotony Property of  $u_{N,Q}[\cdot]$ :* The optimum filters with  $Q$ -level unequal-step tap quantization  $u_{N,Q}[\cdot]$  and  $u_{N+1,Q}[\cdot]$  of lengths  $N$  and  $N+1$ , which maximize  $\Delta_{N,Q}^{(us)}$  and  $\Delta_{N+1,Q}^{(us)}$  for all such filters of the same lengths, yield  $\Delta_{N,Q}^{(us)} \leq \Delta_{N+1,Q}^{(us)}$ .  $\square$

*Proof:* The same proof applies as for the monotony property of  $p_N[\cdot]$ .  $\square$

As above, we derive bounds on the POP-ratio  $\Delta_{N,Q}^{(us)}$  of the optimum filter  $u_{N,Q}[\cdot]$ , which are based on the following lemmas:

*Lemma 5:* The POP-ratio  $\Delta_{N,Q}^{(us)}$  is lower bounded by  $\Delta_{N,Q}^{(us)} \geq \Delta_M^{(ps)}$ , where  $M = \min\{N, Q\}$ .  $\square$

*Proof:* If  $N \leq Q$ , then the  $Q$  levels of  $u_{N,Q}[\cdot]$  can be assigned to the tap values of  $p_N[\cdot]$  in order to obtain  $u_{N,Q}[n] = p_N[n]$  (all  $n$ ) such that  $\Delta_{N,Q}^{(us)} = \Delta_N^{(ps)} = \Delta_M^{(ps)}$ . For larger values of  $N$ , the monotony property completes the proof.  $\square$

*Lemma 6:* The POP-ratio  $\Delta_{N,Q}^{(us)}$  is upper bounded by  $\Delta_{N,Q}^{(us)} \leq \Delta_N^{(ps)}$ .  $\square$

*Proof:* If  $\Delta_{N,Q}^{(us)} > \Delta_N^{(ps)}$ , then  $u_{N,Q}[\cdot]$  would be a better choice for  $p_N[\cdot]$ , which contradicts with the assumption on  $p_N[\cdot]$ .  $\square$

We are now ready to state the

**POP-Ratio Bounds for Optimum Unequal-Step Tap Quantization:** The POP-ratio  $\Delta_{N,Q}^{(us)}$  of the optimum filter  $u_{N,Q}[\cdot]$  with  $Q$ -level unequal-step tap quantization is bounded by

$$u_{N,Q}^{(us)} \leq \Delta_{N,Q}^{(us)} \leq U_{N,Q}^{(us)},$$

where the lower bound is  $u_{N,Q}^{(es)} = \Delta_M^{(ps)}$  ( $M = \min\{N, Q\}$ ) and the upper bound is  $U_{N,Q}^{(us)} = \Delta_N^{(ps)}$ .  $\square$

As approximations  $\tilde{u}_{N,Q}[\cdot]$  on  $u_{N,Q}[\cdot]$ , we apply the modified gradient algorithm to the quantization levels of  $\tilde{e}_{N,Q}[\cdot]$  rather than to the tap values of  $p_N[\cdot]$  and use the initial values  $x_{up} \triangleq 1$  and  $x_{low} \triangleq 10^{-3}$ . Table 3 and Figures 4b and d show the obtained values  $\Delta_{N,Q}^{(\tilde{u}s)}$  for some sequences  $s[\cdot]$ . Because we can only determine  $\tilde{p}[\cdot]$  instead of  $p[\cdot]$ , we have only computed the lower bound  $u_{N,Q}^{(\tilde{u}s)}$  using  $\tilde{p}_N[\cdot]$  rather than  $p_N[\cdot]$ ; because, additionally, the upper bound for  $N \rightarrow \infty$  is  $U_{N,Q}^{(\tilde{u}s)} = \infty$ , Table 3 lists only  $u_{N,Q}^{(\tilde{u}s)}$ .

In Figures 4b and d, we again observe that the POP-ratio  $\Delta_{N,Q}^{(\tilde{u}s)}$  increases with increasing filter length  $N$  only up to a saturation length  $N_{sat}$ . Afterwards, for  $N > N_{sat}$ , the sidelobes due to quantization can no longer be decreased; in some cases, they even increase due to nonideal evaluation of  $\tilde{u}_{N,Q}[\cdot]$ . From Table 3, we conclude that, for values  $\frac{Q}{L} < 3.2$ , the modified gradient algorithm may output a filter with worse POP-ratio performance than the filter that was used to prove the lower bound. For larger values of  $\frac{Q}{L}$ , however, the outputs yield better performance. Nevertheless, we anticipate better solutions also for these parameters when appropriate algorithms are available.

| $L$ | (all values in [dB])          |                                  |                               |                                  |                                |                                   |                                |                                   |
|-----|-------------------------------|----------------------------------|-------------------------------|----------------------------------|--------------------------------|-----------------------------------|--------------------------------|-----------------------------------|
|     | $\nu_{\infty,3}^{(\bar{u}s)}$ | $\Delta_{\infty,3}^{(\bar{u}s)}$ | $\nu_{\infty,8}^{(\bar{u}s)}$ | $\Delta_{\infty,8}^{(\bar{u}s)}$ | $\nu_{\infty,16}^{(\bar{u}s)}$ | $\Delta_{\infty,16}^{(\bar{u}s)}$ | $\nu_{\infty,32}^{(\bar{u}s)}$ | $\Delta_{\infty,32}^{(\bar{u}s)}$ |
| 3   | 12.0                          | 9.5                              | 22.6                          | 21.6                             | 40.4                           | 26.9                              | 73.2                           | 32.0                              |
| 4   | —                             | 11.3                             | 17.4                          | 20.8                             | 30.1                           | 27.6                              | 52.4                           | 32.9                              |
| 5   | —                             | 15.6                             | 18.1                          | 25.1                             | 32.5                           | 28.3                              | 59.5                           | 35.3                              |
| 6   | —                             | 9.5                              | 17.0                          | 24.1                             | 26.4                           | 27.6                              | 42.7                           | 33.8                              |
| 7   | —                             | 15.6                             | 18.0                          | 22.8                             | 22.4                           | 27.1                              | 30.5                           | 33.0                              |
| 8   | —                             | 12.0                             | 18.4                          | 24.6                             | 27.6                           | 28.4                              | 40.3                           | 31.6                              |
| 9   | —                             | 10.9                             | —                             | 21.9                             | 19.6                           | 24.9                              | 21.6                           | 26.3                              |
| 10  | —                             | 9.5                              | —                             | 18.4                             | 21.3                           | 28.8                              | 28.3                           | 30.7                              |
| 11  | —                             | 13.1                             | —                             | 22.8                             | 23.3                           | 28.6                              | 32.4                           | 35.3                              |
| 12  | —                             | 18.1                             | —                             | 23.2                             | 24.8                           | 30.5                              | 34.9                           | 34.3                              |
| 13  | —                             | 23.5                             | —                             | 27.1                             | 25.8                           | 32.8                              | 37.2                           | 38.1                              |
| 14  | —                             | 15.6                             | —                             | 20.6                             | 23.8                           | 31.3                              | 31.1                           | 34.0                              |
| 15  | —                             | 17.4                             | —                             | 25.1                             | 22.6                           | 29.6                              | 34.2                           | 39.0                              |
| 16  | —                             | 16.5                             | —                             | 23.8                             | 20.9                           | 30.2                              | 26.6                           | 37.9                              |
| 17  | —                             | 12.1                             | —                             | 24.0                             | —                              | 30.9                              | 24.4                           | 36.2                              |
| 18  | —                             | 13.4                             | —                             | 21.8                             | —                              | 29.4                              | 23.0                           | 36.1                              |
| 19  | —                             | 17.5                             | —                             | 27.3                             | —                              | 33.8                              | 25.3                           | 41.1                              |
| 20  | —                             | 18.1                             | —                             | 22.5                             | —                              | 31.4                              | 25.3                           | 37.9                              |
| 21  | —                             | 14.6                             | —                             | 23.9                             | —                              | 33.4                              | 24.8                           | 34.9                              |
| 22  | —                             | 14.0                             | —                             | 24.4                             | —                              | 30.7                              | 24.0                           | 37.0                              |
| 23  | —                             | 7.4                              | —                             | 23.9                             | —                              | 29.8                              | 24.8                           | 35.8                              |
| 24  | —                             | 15.9                             | —                             | 25.8                             | —                              | 30.5                              | 25.8                           | 35.9                              |
| 25  | —                             | 11.5                             | —                             | 24.4                             | —                              | 30.1                              | 22.8                           | 37.5                              |
| 26  | —                             | 16.0                             | —                             | 26.0                             | —                              | 30.4                              | 25.4                           | 39.2                              |
| 27  | —                             | 16.9                             | —                             | 26.2                             | —                              | 32.8                              | 24.3                           | 38.0                              |
| 28  | —                             | 14.5                             | —                             | 24.7                             | —                              | 31.2                              | 24.2                           | 38.2                              |
| 29  | —                             | 19.0                             | —                             | 28.1                             | —                              | 35.2                              | 23.7                           | 41.1                              |
| 30  | —                             | 15.7                             | —                             | 24.2                             | —                              | 32.3                              | 24.9                           | 38.2                              |
| 31  | —                             | 18.8                             | —                             | 27.5                             | —                              | 32.4                              | 24.8                           | 35.4                              |
| 32  | —                             | 17.7                             | —                             | 24.9                             | —                              | 32.9                              | 23.7                           | 38.2                              |

Table 3: The POP-ratios  $\Delta_{\infty,Q}^{(\bar{u}s)}$  for the best binary sequences  $s[\cdot]$  of length  $L$  in the range  $3 \leq L \leq 32$  (cf. Table 1) together with their approximate lower bound  $\nu_{\infty,Q}^{(\bar{u}s)}$  for unequal-step tap quantization. The bound is omitted if  $Q < L$  because no meaningful value can be computed.

## VI Conclusion and Outlook

We have shown that, for multipath channel estimation, the performance of the truncated inverse filter can be improved by a couple of dB in POP-ratio.

As indicated by the derived bounds, our approach to determine inverse filter approximations with unbiased equal-step tap quantization yielded good results for small filter lengths  $N$  and small numbers of quantization levels  $Q$ . For larger products  $NQ$ , better methods have to be found. One such method could be the “Branch and Bound” method described in [7].

In the case of unequal-step tap quantizations, the presented algorithm may output for small values of  $N$  and  $Q$  a filter with worse performance than the filter that was used to prove the corresponding lower bound, so that the latter filter should be applied. For larger values of  $\frac{Q}{L}$ , however, the algorithm constructs filters with better performance than this lower bound indicates. Nevertheless, we anticipate better solutions also for these parameters when appropriate algorithms are available.

values of  $\frac{Q}{L}$ , however, the algorithm constructs filters with better performance than this lower bound indicates. Nevertheless, we anticipate better solutions also for these parameters when appropriate algorithms are available.

## References

- [1] J. Ruprecht, *Maximum-Likelihood Estimation of Multipath Channels*, ETH Ph.D. thesis supervised by Jim Massey, ISBN 3-89191-270-6, Hartung Gorre Verlag, Konstanz, Germany, 1989.
- [2] J.L. Massey, "Code time division multiple access", *unpublished*, 1989.
- [3] J.-P. de Weck, J. Ruprecht, "Real-time ML estimation of very frequency selective multipath channels", *IEEE Globecom*, 1990, San Diego, California, pp. 908.5.1-6.
- [4] J. Ruprecht, F.D. Neeser, M. Hufschmid, "Code time division multiple access: An indoor cellular system", in *Proceedings of the 42nd IEEE Vehicular Technology Conference*, Denver, 1992.
- [5] P.G. Schelbert, "SAW-correlator module for spread-spectrum DPSK asynchronous demodulation", *IEEE Ultrasonics Symposium*, pp. 145-149, 1990.
- [6] M. Ghermi, J. Ruprecht, "Exhaustive search for the best binary invertible sequences using a DSP", *Internal Report of Swiss PTT Telecom*, 1993.
- [7] B.A. Murtagh, *Advanced linear Programming: Computation and Practice*, McGraw-Hill, 1981.

# Inverses of Linear Sequential Circuits: On Beyond Poles and Zeros... \*

Michael K. Sain

Department of Electrical Engineering

University of Notre Dame

Notre Dame, Indiana 46556 USA

## Abstract

Thirteen years ago, polynomial module ideas were introduced into the systems literature by Wyman and Sain to address the issue of zero structures in inverse dynamics. Modules have since permitted a unified approach to the study of dynamical qualities for systems that are improper, nonminimal, implicit, and which display nontrivial kernels and cokernels. In this paper, we employ this tool to take a new look at the question of inverses for linear sequential circuits, studied early by Massey and Sain in 1968, with a view toward making the zero structure of an inverse linear sequential circuit as simple as possible. The issue turns out to be considerably more subtle than it would appear. We argue that the task of finding inverses with no more than the necessary zero structure does not have a natural solution for traditional cases of left and right inversion, unless the matrices involved are square. At first, this may seem quite counterintuitive. But, when we make clear just exactly how to account for all the zero structure of a system, then the true difficulty of the problem becomes strikingly clear.

## I Introductionour help

In 1965, it was my pleasure to explore some of the technical possibilities of modern algebra in a coding and system theory collaboration with James L. Massey at the University of Notre Dame. Massey eventually assumed Notre Dame's first endowed chair. It is with warm remembrances that I dedicate these extensions of some of our earliest work, on inverse systems, to JLM, from MKS, who is honored to hold that position here today.

Many interesting questions arise in systems with unequal numbers of inputs and outputs, or in systems with a singular relationship between equal numbers of inputs and outputs. Among such questions, none is more intriguing than the disparity between the total number of zeros and the total number of poles. Indeed, when one makes a tally of the total number of poles and zeros of a transfer function matrix, with due attention both to the finite plane

---

\*This work was supported in part by the Frank M. Freimann Chair in Electrical Engineering at the University of Notre Dame.

and the point at infinity, the number of zeros is not equal to the number of poles, as is so comfortably true for the single-input, single-output case.

It is natural to guess that nonzero kernels and cokernels have something to do with zero-like behavior. One fact in support of this view is that the total number of finite and infinite poles, regarded as points, is never less than the total number of finite and infinite zeros, regarded as points. But it is not an easy matter to envision a means to fold the kernel and cokernel information into an overall, integrated, pole-zero picture, which in the past has emphasized points, invariant factors, and so forth.

Nonetheless, there is a natural way to achieve the synthesis of the two types of information; and this is through the use of spaces to represent the poles and zeros. The reader will already be quite familiar with the usual state space idea and its associated operator  $A$ , which is traditionally associated with poles. Similar notions hold for zeros. From the notions of minimal polynomials, Cayley-Hamilton, and so forth, it is easy to surmise that polynomials in the operator  $A$  are quite natural scalars to employ in studying state spaces of poles and zeros. Thus, questions of poles and zeros may be quite naturally addressed in terms of spaces equipped with scalar-ring operations, namely modules. With regard to poles, such studies were initiated about thirty years ago [3]. For zeros, the corresponding steps are much more recent [7].

In this paper, we employ this tool to take a fresh look at the question of exactly how to make the zero structure of an inverse system as simple as possible. As will become clear in the sequel, this issue is much more subtle than is generally appreciated. Moreover, it can fail to have a natural resolution in most surprising cases. We begin with a review of terminology and basic properties.

## II Basic Pole and Zero Spaces

For the field  $k$ , let  $k[z]$  be the ring of polynomials in  $z$  with coefficients in  $k$ , and let  $k(z)$  be the induced quotient field. Let  $R$ ,  $U$ , and  $Y$  be finite-dimensional vector spaces over  $k$ . Observe that  $k[z]$  and  $k(z)$  are also  $k$ -vector spaces. Next use the  $k$ -bilinear tensor product to form  $k[z]$ -modules  $R[z] = k[z] \otimes_k R$ ,  $U[z] = k[z] \otimes_k U$ , and  $Y[z] = k[z] \otimes_k Y$ , and  $k(z)$ -vector spaces  $R(z) = k(z) \otimes_k R$ ,  $U(z) = k(z) \otimes_k U$ , and  $Y(z) = k(z) \otimes_k Y$ . These vector spaces serve as our spaces of signals in the discussion which follows. Note that this approach permits us to proceed without choosing bases. If  $M(z) : R(z) \rightarrow U(z)$  is a  $k(z)$ -linear map, we wish to introduce in a precise way the spaces which will be used to capture the description of its poles and zeros.

The pole module  $P(M(z))$  attached to a  $k(z)$ -linear map  $M(z)$  is defined to be the  $k[z]$ -factor module  $P(M(z)) = R[z]/\{R[z] \cap M^{-1}U[z]\}$ , where  $M^{-1}$  is the inverse image function of  $M(z)$ , defined on  $U[z]$  by  $M^{-1}U[z] = \{r(z) : r(z) \in R(z) \text{ and } M(z)r(z) \in U[z]\}$ .  $P(M(z))$  is finitely generated as a  $k[z]$ -module, because  $R[z]$  is finitely generated. Indeed, one can simply use a finite set of generators for the underlying vector space  $R$  for this purpose. It is a torsion module as well, because every element in  $R[z]$ , even though it is not itself in  $M^{-1}U[z]$ , can be scalar multiplied into one with an appropriately chosen polynomial  $p(z)$  in  $k[z]$ .

Finitely generated, torsion modules over  $k[z]$ , such as the pole module  $P(M(z))$ , have the character of traditional state spaces of finite dimension over the field  $k$ . If we therefore

consider it as a  $k$ -vector space, then  $P(M(z)) = X_P$ , which is the state space of a minimal realization of  $M(z)$ . The scalar multiplication in the module, given by the action  $z : P(M(z)) \rightarrow P(M(z))$ , defines a  $k$ -linear map  $A_P : X_P \rightarrow X_P$  in the realization. This map,  $A_P$ , determines the pole dynamics of the transfer function matrix  $M(z)$ .

The definition of the pole module of  $M(z)$  does not depend upon whether the kernel,  $\ker M(z)$ , of  $M(z)$ , or the cokernel,  $\text{coker } M(z)$ , of  $M(z)$  is nonzero. This is not in general the case for  $k[z]$ -zeros of  $M(z)$ . But there is an agreed-upon module containing those zeros of  $M(z)$  which are finitely generated and torsion, namely those zeros which in their character are essentially like poles. These zeros tend to be the most familiar, and so we introduce them first. The finitely generated, torsion zero module  $Z(M(z))$  attached to a  $k(z)$ -linear map  $M(z)$  was introduced by Wyman and Sain [7] as the  $k[z]$ -factor module  $Z(M(z)) = \{M^{-1}U[z] + R[z]\}/\{\ker M(z) + R[z]\}$ .

The zero module permits a state-space interpretation. As a  $k$ -vector space,  $Z(M(z)) = X_Z$ ; and it is not difficult to see that this space is of finite dimension. As in the case of the pole module, the scalar action  $z : Z(M(z)) \rightarrow Z(M(z))$  defines a  $k$ -linear map  $A_Z : X_Z \rightarrow X_Z$ .

### III Extended Zeros

In our discussion of extending the notion of zeros of transfer function matrices, it will be convenient to consider two cases. The first case is associated with the matrix having a nontrivial kernel. Modules used to describe this case are of a character similar to that used by Kalman [3] to discuss system outputs, and so we refer to these as zeros of output type. The second case is related to the matrix having a nontrivial cokernel. Modules used to describe the new zeros in this situation are of a character similar to that used in the Kalman theory for inputs, and so we refer to these as zeros of input type.

Consider first the zeros of output type. We construct the  $k[z]$ -factor module  $Z_\Gamma(M(z)) = M^{-1}U[z]/\{R[z] \cap M^{-1}U[z]\}$ , which is torsion. In the case in which  $\ker M(z)$  is not equal to zero, however, this module is not finitely generated. It will be called the  $\Gamma$ -zero module of  $M(z)$ . Immediately, we raise the question of the relation of this new module to that introduced in Section 2 for the traditional space of zeros. The answer to this question is quite intuitive and pleasing.

When  $\ker M(z)$  is not zero, we will define a new module  $\Gamma(M(z))$  in the manner  $\Gamma(M(z)) = \ker M(z)/\{R[z] \cap \ker M(z)\}$ . The technique used to relate this new module to the traditional space of zeros makes use of a short exact sequence. It will be easier to explain this terminology when we have an explicit example at hand, and toward this end we introduce just such a sequence:

$$0 \rightarrow \Gamma(M(z)) \rightarrow Z_\Gamma(M(z)) \longrightarrow Z(M(z)) \rightarrow 0. \quad (1)$$

In Equation 1, one needs to visualize above each arrow a linear mapping of modules. The picture is then one of a sequence of modules and linear mappings. Such a sequence is said to be exact at a particular module if the image of the incoming map is equal to the kernel of the outgoing map. The entire sequence is said to be exact if it is exact at each module in the sequence. Observe that exactness at the second module means that the

second mapping has zero kernel; and in an analogous way, exactness at the fourth module means that the third mapping has zero cokernel. Now we appeal to a technical point. The second module is both torsion and divisible, and so the center module is said to split, that is, it is isomorphic to a direct sum of the second and fourth modules, in the manner  $Z_\Gamma(M(z)) \approx \Gamma(M(z)) \oplus Z(M(z))$ .

This situation shows the strong intuitive character of the short exact sequence: If we wish to study the structure of a module, one way is to place it as the centerpiece in such a sequence. We will refer to  $\Gamma(M(z))$  as the divisible zero module of  $M(z)$ , and to  $Z_\Gamma(M(z))$ , the  $\Gamma$ -zero module of  $M(z)$ , as an extended zero module. We can construct two intersecting short exact sequences, as shown in Figure 1. The horizontal sequence displays the two components of the extended zero module, while the vertical sequence explains further the nature of the submodule building block. There is an intuitive way of speaking about factor modules, such as that in the second last position in the vertical sequence. We can say that  $\Gamma(M)$  consists of those vectors in the kernel of  $M(z)$  which are not polynomial vectors.

$$\begin{array}{c} 0 \\ \downarrow \\ \ker M(z) \cap R[z] \\ \downarrow \\ \ker M(z) \\ \downarrow \\ 0 \rightarrow \Gamma(M) \rightarrow Z_\Gamma(M) \rightarrow Z(M) \rightarrow 0 \\ \downarrow \\ 0 \end{array}$$

Figure 1.  $Z_\Gamma(M) \approx \Gamma(M) \oplus Z(M)$

$$\begin{array}{c} 0 \\ \downarrow \\ U[z] \cap \text{im } M(z) \\ \downarrow \\ Y[z] \\ \downarrow \\ 0 \rightarrow Z(M) \rightarrow Z_\Omega(M) \rightarrow \Omega(M) \rightarrow 0 \\ \downarrow \\ 0 \end{array}$$

Figure 2.  $Z_\Omega(G) \approx Z(G) \oplus \Omega(G)$

Now consider zeros of input type, by means of constructing the  $k[z]$ -factor module  $Z_\Omega(M(z)) = U[z]/\{U[z] \cap MR[z]\}$ . Although this module is finitely generated, it is not true that every equivalence class in it is torsion. We will refer to it as the  $\Omega$ -zero module of  $M(z)$ . When  $\text{coker } M(z)$  is nonzero, we can follow a program similar to that above, for the output zeros. If we define  $\Omega(M(z)) = U[z]/\{U[z] \cap MR(z)\}$ , then it may be established that there is a short exact sequence

$$0 \rightarrow Z(M(z)) \rightarrow Z_\Omega(M(z)) \rightarrow \Omega(M(z)) \rightarrow 0 \quad (2)$$

of  $k[z]$ -modules and  $k[z]$ -linear maps. The factor module in the sequence can be shown to be torsion-free. It then follows from the nature of the polynomial ring of scalars that  $\Omega(M(z))$  is a free module, and from that it may be shown that the sequence splits, in the manner  $Z_\Omega(M(z)) \approx \Omega(M(z)) \oplus Z(M(z))$ . We will call  $\Omega(M(z))$  the free zero module of  $M(z)$ . Like  $Z_\Gamma(M(z))$ , then,  $Z_\Omega(M(z))$  is an extended zero module, this time of  $\Omega$ -type. The same pictorial type of presentations can be made in this case as well, and we indicate one in Figure 2.

## IV Transfer Matrix Equations: Fixed Zeros

Two theorems are reviewed in this section. Theorem 1 defines and characterizes the module of fixed zeros in the transfer function matrix equation  $T(z) = P(z)M(z)$ . Two cases are

considered, depending upon whether  $M(z)$  is given or whether  $P(z)$  is given. The matrix  $T(z)$  is assumed given in both cases. The modules of fixed zeros are given the name of matching zero modules. The two cases take on the character of the two foregoing types of extended zero modules. Theorem 2 shows in a precise technical sense that these matching zero modules are contained in all solutions to the transfer function matrix problem, in the one case as a factor module, and in the other case as a submodule.

The matching  $\Gamma$ -zero module is denoted by  $Z_\Gamma$  and defined to be the  $k[z]$ -factor module  $Z_\Gamma = T^{-1}Y[z]/\{T^{-1}Y[z] \cap M^{-1}U[z]\}$ , where  $T(z) : R(z) \rightarrow Y(z)$  is a  $k(z)$ -linear map. The matching  $\Omega$ -zero module is denoted by  $Z_\Omega$  and defined to be the  $k[z]$ -factor module  $Z_\Omega = \{PU[z] + TR[z]\}/TR[z]$ , for  $P(z) : U(z) \rightarrow Y(z)$  a  $k(z)$ -linear map. The algebraic character of matching zero modules of  $\Gamma$ -type and of  $\Omega$ -type is settled by Theorem 1.

### **Theorem 1** (Matching Zeros, [5])

Let  $Z_\Gamma$  and  $Z_\Omega$  be the matching zero modules of  $\Gamma$ -type and  $\Omega$ -type, respectively, for  $k(z)$ -linear maps  $M(z) : R(z) \rightarrow U(z)$ ,  $P(z) : U(z) \rightarrow Y(z)$ , and  $T(z) : R(z) \rightarrow Y(z)$ . Then  $Z_\Gamma \approx \Gamma \oplus Z'_\Gamma$ , where  $\Gamma$  is torsion divisible and  $Z'_\Gamma$  is finitely generated and torsion; and  $Z_\Omega \approx \Omega \oplus Z'_\Omega$ , where  $\Omega$  is finitely generated and free while  $Z'_\Omega$  is finitely generated and torsion.

### **Remark**

It can be seen quite clearly from the theorem that the two matching zero modules have the nature of the two types of extended zeros. We shall show in the next section that these modules have a most interesting structure. Theorem 2, which follows, then provides the inclusion results desired.

### **Theorem 2** (Fixed Zeros, [5])

Suppose that  $Z_\Gamma$  and  $Z_\Omega$  are the matching zero modules of  $\Gamma$ -type and  $\Omega$ -type, respectively, and let  $T(z) : R(z) \rightarrow Y(z)$  be a  $k(z)$ -linear map.

1. If  $P(z) : U(z) \rightarrow Y(z)$  is a  $k(z)$ -linear map whose image contains that of  $T(z)$ , and if  $M(z) : R(z) \rightarrow U(z)$  is a  $k(z)$ -linear map which satisfies the equation  $T(z) = P(z)M(z)$ , then there exists an epic,  $k[z]$ -linear map

$$\beta_\Omega(z) : Z_\Omega(M(z)) \rightarrow Z_\Omega, \quad (3)$$

so that  $Z_\Omega$  is isomorphic to a factor module of the  $\Omega$ -zero module of  $M(z)$ .

2. If  $M(z) : R(z) \rightarrow U(z)$  is a  $k(z)$ -linear map whose kernel is contained in that of  $T(z)$ , and if  $P(z) : U(z) \rightarrow Y(z)$  is a  $k(z)$ -linear map which satisfies the equation  $T(z) = P(z)M(z)$ , then there exists a monic,  $k[z]$ -linear map

$$\beta_\Gamma(z) : Z_\Gamma \rightarrow Z_\Gamma(P(z)), \quad (4)$$

so that  $Z_\Gamma$  is isomorphic to a submodule of the  $\Gamma$ -zero module of  $P(z)$ .

### Remark

Equations 3 and 4 are the key assertions, the former an onto map, which makes the matching zero module a factor, and the latter a one-to-one map, which makes the matching zero module a submodule. These are just two basic ways of asserting inclusion in an algebraic sense.

## V Intuitive Meaning of Matching Zeros

In this section, we do two things. First, we present a technical result, Theorem 3, which breaks down the matching zero modules, in both cases, by means of triples of short exact sequences. Then, after the theorem, we provide a number of diagrams to help in its interpretation. These provide remarkable insights into the whole question of fixed zeros in particular, and more generally into the roles of kernels and cokernels in inverse dynamical system theory.

**Theorem 3** (Matching Structure, [5])

Let  $Z_\Gamma$  and  $Z_\Omega$  be the matching zero modules of  $\Gamma$ -type and  $\Omega$ -type, respectively, for  $k(z)$ -linear maps  $M(z) : R(z) \rightarrow U(z)$ ,  $P(z) : U(z) \rightarrow Y(z)$ , and  $T(z) : R(z) \rightarrow Y(z)$ .

1. If  $[T(z) \ P(z)] : R(z) \oplus U(z) \rightarrow Y(z)$  is the  $k(z)$ -linear map with action

$$[T(z) \ P(z)](r(z), u(z)) = T(z)r(z) + P(z)u(z), \quad (5)$$

then there exist  $k[z]$ -modules  $Z_1$  and  $P_1$ , together with appropriate  $k[z]$ -linear maps, such that the following three short sequences are exact:

$$0 \rightarrow P(T(z)) \rightarrow P([T(z) \ P(z)]) \rightarrow P_1 \rightarrow 0; \quad (6)$$

$$0 \rightarrow Z_1 \rightarrow Z_\Omega(T(z)) \rightarrow Z_\Omega([T(z) \ P(z)]) \rightarrow 0; \quad (7)$$

$$0 \rightarrow Z_1 \rightarrow Z_\Omega \rightarrow P_1 \rightarrow 0. \quad (8)$$

2. If

$$\begin{bmatrix} T(z) \\ M(z) \end{bmatrix} : R(z) \rightarrow Y(z) \oplus U(z) \quad (9)$$

is the  $k(z)$ -linear map having action

$$\begin{bmatrix} T(z) \\ M(z) \end{bmatrix}(r(z)) = (T(z)r(z), M(z)r(z)), \quad (10)$$

then there exist  $k[z]$ -modules  $Z_2$  and  $P_2$ , together with appropriate  $k[z]$ -linear maps, such that the following three short sequences are exact:

$$0 \rightarrow P_2 \rightarrow P \left( \begin{bmatrix} T(z) \\ M(z) \end{bmatrix} \right) \rightarrow P(T(z)) \rightarrow 0; \quad (11)$$

$$0 \rightarrow Z_\Gamma \left( \begin{bmatrix} T(z) \\ M(z) \end{bmatrix} \right) \rightarrow Z_\Gamma(T(z)) \rightarrow Z_2 \rightarrow 0; \quad (12)$$

$$0 \rightarrow P_2 \rightarrow Z_\Gamma \rightarrow Z_2 \rightarrow 0. \quad (13)$$

A diagrammatic interpretation for the second part of the representation theorem is given in the next three figures. The first part of the theorem can also receive such a treatment. For the sake of space, however, we omit the details. Figure 3 shows the basic short exact sequence for the fixed module  $Z_\Gamma$ , together with the further breakdown of the submodule  $P_2$ .

$$\begin{array}{ccccccc} & & 0 & & & & \\ & & \downarrow & & & & \\ 0 & \longrightarrow & P_2 & \longrightarrow & Z_\Gamma & \longrightarrow & Z_2 & \longrightarrow 0 \\ & & \downarrow & & & & \\ & & P \left( \begin{bmatrix} T \\ M \end{bmatrix} \right) & & & & \\ & & \downarrow & & & & \\ & & P(T) & & & & \\ & & \downarrow & & & & \\ & & 0 & & & & \end{array}$$

Figure 3.

$$\begin{array}{ccccccc} & & 0 & & & & \\ & & \downarrow & & & & \\ Z_\Gamma \left( \begin{bmatrix} T \\ M \end{bmatrix} \right) & & \downarrow & & & & \\ & & Z_\Gamma(T) & & & & \\ & & \downarrow & & & & \\ 0 & \longrightarrow & P_2 & \longrightarrow & Z_\Gamma & \longrightarrow & Z_2 & \longrightarrow 0 \\ & & & & & & \downarrow & \\ & & & & & & 0 & \end{array}$$

Figure 4.

The striking thing about Figure 3 is the appearance of the composite matrices built from  $T(z)$  and  $M(z)$ . This behavior, as well as the appearance of the extended zero spaces, is quite unpredicted from the single-input, single-output case. Figure 4 then supplies the corresponding information for the factor module  $Z_2$ . An intuitive interpretation of these two diagrams, appears in Figure 5.

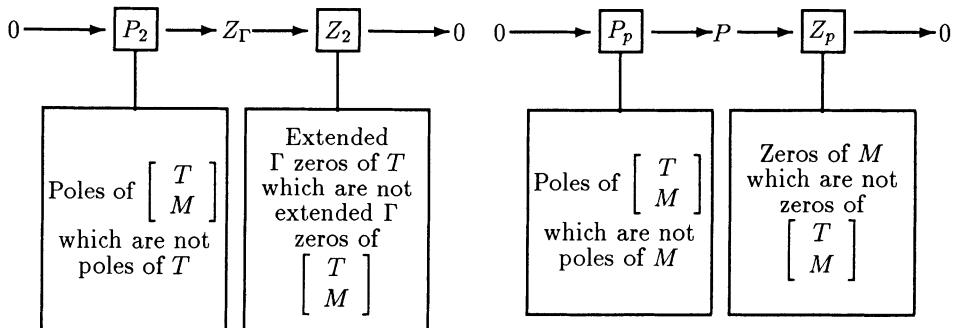


Figure 5.

Figure 6.

In the next section, we report briefly, so that the reader can compare, the original results of Conte, Perdon, and Wyman [1] on fixed poles.

## VI Conte, et al. [1]: Fixed Poles

We omit the technical theorem, and make use of the diagrammatic method to indicate the qualitative nature of the results. For purposes of consistency, the fixed poles are considered for the same case as that of Figures 3, 4, and 5. There is a module of fixed poles,  $P$ , which may be placed into the center of a short exact sequence as in Figure 7, wherein the nature of the submodule is also examined.

$$\begin{array}{ccccccc} & & 0 & & & & \\ & & \downarrow & & & & \\ 0 & \longrightarrow & P_p & \longrightarrow & P & \longrightarrow & Z_p \longrightarrow 0 \\ & & \downarrow & & & & \\ & & P \left( \begin{bmatrix} T \\ M \end{bmatrix} \right) & & & & \\ & & \downarrow & & & & \\ & & P(M) & & & & \\ & & \downarrow & & & & \\ & & 0 & & & & \end{array}$$

Figure 7.

$$\begin{array}{ccccccc} & & 0 & & & & \\ & & \downarrow & & & & \\ & & Z \left( \begin{bmatrix} T \\ M \end{bmatrix} \right) & & & & \\ & & \downarrow & & & & \\ & & Z(M) & & & & \\ & & \downarrow & & & & \\ 0 & \longrightarrow & P_p & \longrightarrow & P & \longrightarrow & Z_p \longrightarrow 0 \\ & & \downarrow & & & & \\ & & 0 & & & & \end{array}$$

Figure 8.

Figure 8 then further explores the character of the factor module in the horizontal sequence. Next compare Figures 3 and 7. At first, the vertical sequences may seem to be the same. But note that the pole modules at the bottom of the vertical sequences are different, one depending upon  $T(z)$  and the other depending upon  $M(z)$ . Figure 8 is to be compared with Figure 4. Again, there is the difference between  $T(z)$  and  $M(z)$ ; but note also the extension of the notion of zero. Finally, there is Figure 6, which is once more the intuitive counterpart of a joining together of Figures 7 and 8. We suspect that the reader can guess the result for fixed poles in the other case.

## VII The Point at Infinity

Up to this point, we have put the focus on the polynomial ring  $k[z]$ , so that as a result the zeros and poles of traditional type have been in the finite plane, so to speak—because we have not been too specific about the nature of  $k$ . The terminology comes, of course, from the complex field case. If we replace the subring  $k[z]$  of  $k(z)$  with any subring  $O$  which contains the base field  $k$ , which has quotient field  $k(z)$ , and which is a principal ideal domain, then a great number of the preceding definitions and properties follow through, with the need of course for some physical re-interpretation. Such rings include localizations of  $k[z]$  and discrete valuation rings. A key example of the latter is  $O_\infty$ , the subring of proper transfer functions. Other illustrations arise from forming the intersection of discrete valuation rings. An instance is the ring  $O_{ps}$  of transfer functions which are proper and stable, when  $k$  is the real numbers. When  $k[z]$  is replaced by  $O$ ,  $R[z]$ , and its counterparts for  $U$  and  $Y$ , must be

replaced by  $\Omega_O R = O \otimes_k R$ ,  $\Omega_O U = O \otimes_k U$ , and  $\Omega_O Y = O \otimes_k Y$ . Similarly, replace  $U[z]$  by  $\Omega_O U$ , and so forth for the other two cases.

Armed with these notions, we can return to the question of fixed poles and zeros in the transfer function matrix equation  $T(z) = P(z)M(z)$ . Once again the focus is on the situation in which  $M(z)$  is given. The other case is left as an exercise. The appropriate pictures, then, for the fixed poles and the fixed zeros are those of Figures 9 and 10. Notice that the superscript  $\infty$  has been added to distinguish these results. Recall that, if there are no fixed poles at infinity, then it will be possible to obtain a causal or proper solution.

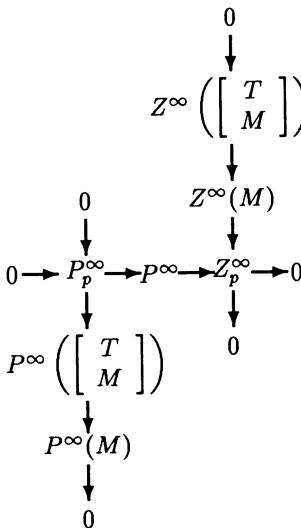


Figure 9.

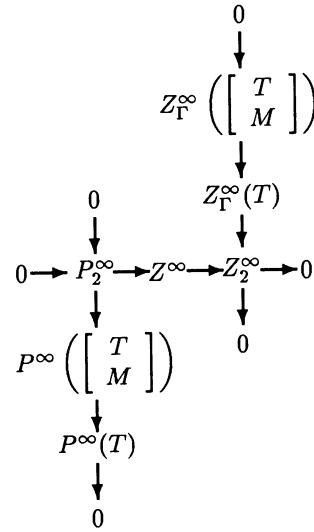


Figure 10.

## VIII The Extended Plane

We have observed in the preceding discussion that the number of finite poles and poles at infinity of a transfer function  $G(z)$  may be greater than the number of its finite zeros and zeros at infinity. This difference between the total number of poles and the total number of zeros may be calculated in terms of Kronecker indices or Wedderburn numbers [6], [2].

The Wedderburn-Forney construction introduced in [8] attaches a finite-dimensional vector space over  $k$ ,  $\mathcal{W}(\mathcal{C})$ , to a  $k(z)$ -vector space  $\mathcal{C}(z)$ . This space associated with  $\mathcal{C}(z) \subset V(z)$  is  $\mathcal{W}(\mathcal{C}) = \pi_-(\mathcal{C})/\{\mathcal{C} \cap z^{-1}\Omega_\infty V\}$ , where  $\pi_-$  is the  $k$ -linear projection which discards the polynomial part of a vector.

It is possible to combine the traditional pole space, associated with the finite plane, with its counterpart at infinity, by forming a direct sum. We call this the global pole space and denote it by  $\mathcal{P}(G(z))$ . Likewise, we can form a global zero space from the direct sum of the traditional zero space and its counterpart at infinity, and we denote the result by  $\mathcal{Z}(G(z))$ . The following theorem then holds.

**Theorem 4** (Global Structure, [8])

There is an exact sequence of finite dimensional vector spaces over  $k$ ,

$$0 \rightarrow \mathcal{Z}(G(z)) \rightarrow \frac{\mathcal{P}(G(z))}{\mathcal{W}(\ker G(z))} \rightarrow \mathcal{W}(\text{im } G(z)) \rightarrow 0, \quad (14)$$

where  $\mathcal{Z}(G(z))$  is the global space of zeros of  $G(z)$ ,  $\mathcal{P}(G(z))$  is the global space of poles of  $G(z)$ , and  $\mathcal{W}(\ )$  is the Wedderburn-Forney construction.

**Remark**

Using the traditional dimension relations for quotient spaces, we can see easily that the two Wedderburn-Forney spaces in this sequence account for the missing zeros, which then balance with the total number of poles, as desired. We have, then, another way of looking at kernels and cokernels in the pole-zero setting. When combined with the extended zero ideas preceding, this viewpoint produces some rather remarkable results with regard to inverse system structure.

## IX Invertibility: A Fresh Look

In this section, our main results involve a further and more detailed examination of the transfer function equation  $T(z) = P(z)M(z)$  under two conditions: (1)  $\ker T(z) = 0$  and (2)  $\text{coker } T(z) = 0$ . The big question is whether or not the fixed-zero constraints represented by 3 and 4, in Theorem 2, can be achieved with isomorphism. In other words, are there solutions  $M(z)$ , or  $P(z)$ , which have the matching zero modules  $Z_\Omega$ , or  $Z_\Gamma$ , as their extended zero modules of type  $\Omega$ , or  $\Gamma$ , respectively? We refer to such solutions as *essential solutions*. Because of space limitations, it will be necessary to omit the proofs.

Because essential solutions must produce the correct free or divisible extended zeros, the next theorem provides a basis to settle the question completely.

**Theorem 5** (Zeros: Free, Divisible, and Fixed)

In the equation  $T(z) = P(z)M(z)$  of  $k(z)$ -linear maps,  $\Omega(M(z)) \approx \Omega(Z_\Omega)$ , and  $\Gamma(P(z)) \approx \Gamma(Z_\Gamma)$ , if and only if  $M(\ker T(z)) = \ker P(z)$ .

**Remark**

If a system  $T(z)$  is invertible, then obviously  $\ker T(z) = 0$ . Thus, to achieve the constraints of Theorem 2 with isomorphism implies that  $\ker P(z) = 0$  as well, which in turn would require that  $P(z)$  is an isomorphism. Then  $M(z)$  is an isomorphism as well, so that there are no extended zeros of either type anywhere in the equation. Therefore, for the more general case of left inversion or right inversion, when the factors are not isomorphisms, there is no possibility of constructing solutions whose extended zero modules meet the bounds established in Theorem 2. We will thus be able to have a conclusive theorem. First, however, there are more remarks.

**Remark**

It may seem that the linking of the conditions  $\Omega(M(z)) \approx \Omega(Z_\Omega)$  and  $\Gamma(P(z)) \approx \Gamma(Z_\Gamma)$  in the theorem could be restrictive. In other words, we could envision a situation in which

the first condition would hold, but not the second, or conversely. But it may be shown that this is not the case. Indeed, if the first condition holds, then so does the second; and if the second condition holds, then so does the first. Accordingly, no loss of generality is incurred.

### **Remark**

Theorem 5 is not speaking of essential solutions, but only of a necessary condition for essential solutions. It is possible for the condition  $M(\ker T(z)) = \ker P(z)$  to occur without both  $P(z)$  and  $M(z)$  being essential solutions. Now the theorem.

### **Theorem 6 (Invertibility and Fixed Zeros)**

Consider the general invertibility equations  $1_{Y(z)} = P(z)M(z)$  or  $1_{R(z)} = P(z)M(z)$ . Then the factors  $M(z)$  or  $P(z)$  are essential solutions if and only if they are  $k(z)$ -linear isomorphisms, in which case both factors share this property.

In view of the stringent nature of Theorem 5, we may ask about the situation under slightly weaker conditions. So we shall consider  $k(z)$ -linear maps  $T(z)$  which have zero kernels or cokernels, but not necessarily both. For the former case, we refer to *extended left invertibility*. For the latter case, we speak of *extended right invertibility*. The next theorem addresses our question in these circumstances.

### **Theorem 7 (Extended Invertibility and Fixed Zeros)**

Consider the equation  $T(z) = P(z)M(z)$  of  $k(z)$ -linear maps. If a system  $P(z)$  is right invertible, and if  $\text{coker } T(z) = 0$ , then there exists a solution  $M(z)$  which is essential if and only if the dimension of  $R$  is no less than the dimension of  $U$ . If a system  $M(z)$  is left invertible, and if  $\ker T(z) = 0$ , then there exists a solution  $P(z)$  which is essential if and only if the dimension of  $Y$  is no less than the dimension of  $U$ .

### **Remark**

Notice that, in this theorem, if we add to the first part the assumption that  $\ker T(z)$  vanishes, then clearly the discussion reduces to that of Theorem 6. Thus Theorem 7 is a true generalization.

### **Remark**

A number of further generalizations are possible; but space limitations do not permit their elaboration. For instance, we can generalize Theorem 7 to the situations in which  $\text{im } T(z) = \text{im } P(z)$  on the one hand, and  $\ker T(z) = \ker M(z)$  on the other.

## **X Conclusions**

We have seen in Section VIII that an accounting of all the zeros in a multi-input, multi-output system—so as to have the total number of zeros equal to the total number of poles—leads naturally to the study of kernels and cokernels of the transfer functions involved. However, Theorem 4 is valid on the  $k$ -vector space level only, because the representation of poles and zeros at infinity, and their representation elsewhere, require different rings. If one wishes to involve dynamical action in the discussion, and thus employ module theory, it is necessary to develop the kernel, or cokernel, into a module. This means, in intuitive terms,

that we must find a larger set which includes the kernel, or cokernel, and which is closed under module scalar multiplication. It is in just this way that the notions of divisible and free zeros have been brought into the picture. When combined with the classical transmission zero modules, they lead to the extended zero modules of Section III. Then the study of fixed zeros in transfer function equations is possible and natural, as presented in Sections IV and V. Moreover, many questions which pertain to invertibility can be understood in this format. Remarkably, it turns out that the problem of constructing left inverses or right inverses with the simplest zero structure possible cannot be solved unless the systems are isomorphisms. This means that the design of zeros for inverse systems will be more of an *ad hoc* procedure, depending upon what structures are chosen to be combined with the fixed zeros. In view of the importance of inversion in communication and control, this is quite a thought-provoking situation.

## XI Special Interpretation

In view of the special occasion associated with this volume, as well as Professor Massey's well-known interest in technical lyrics, we have paraphrased the results and conclusions above with the following poem:

There was a code from Notre Dame,  
 And to another space it came!  
 To pursue a tomorrow,  
 Without undue zeroes,  
 It found that the secret was *more rows!*

Happy Birthday, Jim, and thank you for everything that you have done for me personally and professionally, and for my family. As a present, I have prepared your very own short exact sequence:

$$0 \rightarrow JLM \rightarrow MKS \rightarrow MKS/JLM \rightarrow 0$$

## References

- [1] G. Conte, A.M. Perdon, and B.F. Wyman, "Fixed Poles in Transfer Function Equations", *SIAM Journal on Control and Optimization*, Volume 26, pp. 356-368, 1988.
- [2] G.D. Forney, Jr., "Minimal Bases of Rational Vector Spaces with Applications to Multivariable Linear Systems", *SIAM Journal on Control and Optimization*, Volume 13, pp. 493-520, 1975.
- [3] R.E. Kalman, "Algebraic Structure of Linear Dynamical Systems. I. The Module of  $\Sigma$ ", *Proceedings National Academy of Sciences (USA)*, Volume 54, pp. 1503-1508, 1965.
- [4] J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits", in *IEEE Trans. Computers*, Volume C-17, pp. 330-337, 1968.

- [5] M.K. Sain, B.F. Wyman, and J.L. Peczkowski, “Extended Zeros and Model Matching”, *SIAM Journal on Control and Optimization*, Volume 29, pp. 562-593, 1991.
- [6] J.H.M. Wedderburn, *Lectures on Matrices*. American Mathematical Society Colloquium Publications, Volume 17, Chapter 4, 1934.
- [7] B.F. Wyman and M.K. Sain, “The Zero Module and Essential Inverse Systems”, *IEEE Transactions on Circuits and Systems*, Volume 27, pp. 112-126, 1981.
- [8] B.F. Wyman, M.K. Sain, G. Conte, and A.M. Perdon, “On the Zeros and Poles of a Transfer Function”, *Journal of Linear Algebra and Its Applications*, Vol. 122/123/124, pp. 123-144, 1989.

# Binary Sequences With Small Correlations

Gerald Seguin

Department of Electrical  
and Computer Engineering

Royal Military College of Canada  
Kingston, Ontario  
Canada K7K 5L0

Germain Drolet

Department of Electrical  
and Computer Engineering

Royal Military College of Canada  
Kingston, Ontario  
Canada K7K 5L0

## Abstract

Given  $\alpha$  and  $\beta$  as two affine transformations on the ring of integers modulo  $n$ , define a binary  $(\alpha, \beta)$  sequence as a sequence  $\underline{x} = (x_0, x_1, \dots, x_{n-1})$  that satisfies  $x_{\alpha(i)} = (-1)^i x_i$  and  $x_{\beta(i)} = (-1)^i x_i$ . The aperiodic autocorrelation function of an  $(\alpha, \beta)$  sequence is shown to satisfy a simple property. In particular, the Barker sequences of odd length are characterized as  $(\alpha, \beta)$  sequences. Further,  $(\alpha, \beta)$  sequences of various lengths with small correlation are given.

## I Introduction

Let  $\underline{x} = (x_0, x_1, x_2, \dots, x_{n-1})$  be a binary  $(\pm 1)$   $n$ -tuple. The aperiodic autocorrelation function of  $\underline{x}$  is defined as:

$$F_k = \sum_{j=0}^{n-k-1} x_j x_{j+k}, \quad k = 0, 1, \dots, n-1 \quad (1)$$

(In the sequel, the indices in  $x_j$  and  $F_k$  are always reduced modulo  $n$ .) The problem under study in this paper is the construction of  $n$ -tuples with “small values” of  $F$  where,

$$F = \max_{0 < k < n} |F_k| \quad (2)$$

Sequences (i.e.  $n$ -tuples) with small  $F$  find use in the design of radar signals and in synchronization problems [1,2]. A sequence for which  $F \leq 1$  is called a *Barker sequence* [2]. It has been shown [3] that Barker sequences of odd length  $n$  do not exist for  $n > 4$  [4].

We study binary  $n$ -tuples satisfying certain constraints. (Over the finite field  $\mathbf{F}_2$ , the sequences are the solutions of a system of linear equations.) This technique produces the Barker sequences of odd length; we also discover sequences of lengths 15, 19, and 21 with  $F = 3$ .

The work reported in this paper represents an extension of some of the results which appear in the Ph.D dissertation of the first author and which was done under the supervision of Jim Massey.

## II The Definition of $(\alpha, \beta)$ -Sequences

Following Massey and Uhran [5], we define the even and odd autocorrelation functions of  $\underline{x}$  by setting for  $k = 0, 1, \dots, n - 1$ :

$$\begin{aligned} E_k &= F_k + F_{n-k} = \sum_{j=0}^{n-1} x_j x_{j+k} \\ \theta_k &= F_k - F_{n-k} \end{aligned} \quad (3)$$

It now follows that

$$F_k = \frac{1}{2}(E_k + \theta_k) \quad (4)$$

and we also have that

$$E_{n-k} = E_k \quad \text{and} \quad \theta_{n-k} = -\theta_k \quad (5)$$

An *affine transformation* on the ring of integers modulo  $n$ ,  $\mathbf{Z}_n$ , is a function of the form:

$$\alpha(i) = ai + r, \quad i \in \mathbf{Z}_n \quad (6)$$

where the arithmetic is modulo  $n$ ;  $a, r \in \mathbf{Z}_n$  and  $a$  is relatively prime to  $n$ . The affine transformations on  $\mathbf{Z}_n$  form a group of order  $n\phi(n)$  under function composition where  $\phi(\cdot)$  is the Euler totient function.

We define  $\alpha(\underline{x})$  by setting:

$$\alpha(\underline{x})_i = x_{\alpha(i)}, \quad i \in \mathbf{Z}_n \quad (7)$$

Affine transformations preserve the even autocorrelation function in the sense that:

$$E_k(\alpha(\underline{x})) = E_{ak}(\underline{x}), \quad k \in \mathbf{Z}_n \quad (8)$$

where  $E_k(y)$  means the  $k$ th correlation coefficient of the  $n$ -tuple  $\underline{y}$ . From Equation (8), we see that if  $\underline{x}$  is invariant under  $\alpha$ ; i.e. if  $\alpha(\underline{x}) = \underline{x}$ , then

$$E_{ak}(\underline{x}) = E_k(\underline{x}) \quad (9)$$

We define two special sequences  $\underline{z}$  and  $\underline{z}'$  by setting,

$$z_i = (-1)^i, \quad z'_i = (-1)^{i+1}, \quad i \in \mathbf{Z}_n \quad (10)$$

where in Equation (10), the exponents are not reduced modulo  $n$ . Let  $\underline{xz}$  denote the componentwise product of  $\underline{x}$  and  $\underline{z}$ . The following lemma implies that  $\underline{x} \rightarrow \underline{xz}$  (or  $\underline{xz}'$ ) is a Barker preserving transformation [6].

**Lemma 1:**  $F_k(\underline{xz}) = F_k(\underline{xz}') = (-1)^k F_k(\underline{x}), \quad k \in \mathbf{Z}_n$

*Proof:*

$$F_k(\underline{xz}) = \sum_{j=0}^{n-k-1} x_j (-1)^j z_{j+k} (-1)^{j+k} = (-1)^k \sum_{j=0}^{n-k-1} x_j x_{j+k} = (-1)^k F_k(\underline{x})$$

similarly for  $\underline{xz}'$

We say that  $\underline{x}$  is an  $\alpha$  sequence or an  $\alpha'$  sequence if, respectively,

$$\begin{aligned}\alpha(\underline{x}) &= \underline{xz}, \quad i.e. \quad x_{\alpha(i)} = (-1)^i x_i, \quad i \in \mathbf{Z}_n \\ \text{or } \alpha(\underline{x}) &= \underline{xz}', \quad i.e. \quad x_{\alpha(i)} = (-1)^{i+1} x_i, \quad i \in \mathbf{Z}_n\end{aligned}\tag{11}$$

**Theorem 1:** If  $\underline{x}$  is an  $\alpha$  sequence or  $\alpha'$  sequence,  $\alpha(i) = ai + r$ , then

$$F_{ak}(\underline{x}) + F_{n-ak}(\underline{x}) = (-1)^k F_k(\underline{x}) + (-1)^{n-k} F_{n-k}(\underline{x}), \quad k \in \mathbf{Z}_n$$

*Proof:* Suppose that  $\alpha(\underline{x}) = \underline{xz}$  (The proof is the same for the case  $\alpha(\underline{x}) = \underline{xz}'$ ), then,

$$\begin{aligned}F_{ak}(\underline{x}) + F_{n-ak}(\underline{x}) &= E_{ak}(\underline{x}) = E_k(\alpha(\underline{x})) \\ &= F_k(\alpha(\underline{x})) + F_{n-k}(\alpha(\underline{x})) \\ &= F_k(\underline{xz}) + F_{n-k}(\underline{xz}) \\ &= (-1)^k F_k(\underline{x}) + (-1)^{n-k} F_{n-k}(\underline{x}).\end{aligned}$$

**Corollary:** (Golay [7], Séguin [8]): If  $\alpha(i) = -i - 1$ , then there are  $2^{n+1/2}$   $\alpha$  sequences if  $4|n-1$  and  $2^{n+1/2}$   $\alpha'$  sequences if  $4 \nmid n+1$ . If  $n$  is odd, and if  $\underline{x}$  is an  $\alpha$  or an  $\alpha'$  sequence, then

$$F_k(\underline{x}) = 0, \quad k \in \mathbf{Z}_n, \quad k \text{ odd}$$

*Proof:* Suppose  $\underline{x}$  is an  $\alpha$  sequence of odd length  $n$ , then  $x_{-i-1} = (-1)^i x_i$ . Setting  $i = \frac{n-1}{2}$ , we obtain,

$$x_{\frac{n-1}{2}} = x_{n-\frac{n-1}{2}-1} = (-1)^{\frac{n-1}{2}} x_{\frac{n-1}{2}}$$

and so  $\frac{n-1}{2}$  must be even, i.e.  $4|n-1$ . If  $4|n-1$ , we see that we may choose  $x_0, x_1, \dots, x_{\frac{n-1}{2}}$  arbitrarily and the condition  $x_{\alpha(i)} = (-1)^i x_i$  determines  $x_{\frac{n+1}{2}}, \dots, x_{n-1}$ . Similar arguments obtain the result for  $\alpha'$  sequences.

If  $\underline{x}$  is an  $\alpha$  or  $\alpha'$  sequence, then by Theorem 1, with  $a = -1$ , we have that;

$$F_{n-k} + F_k = (-1)^k F_k + (-1)^{n-k} F_{n-k}$$

Hence, if  $k$  is odd, then  $n - k$  is even, so that:

$$F_{n-k} + F_k = -F_k + F_{n-k}$$

Therefore  $2F_k = 0$  which means that  $F_k = 0$ .

If  $\alpha$  is an affine transformation on  $\mathbf{Z}_n$ , then the  $\alpha$ -orbit of  $i \in \mathbf{Z}_n$  is:

$$i^\alpha = \{\alpha^j(i) \mid j = 0, 1, \dots\} \tag{12}$$

The  $\alpha$ -orbits constitute a partition of  $\mathbf{Z}_n$  and the cardinality of  $i^\alpha$  is the least positive integer  $m$  such that  $\alpha^m(i) = i$ .

The simple proof of the following theorem may be found in [8].

**Theorem 2:** There exists an  $\alpha$  sequence ( $\alpha'$  sequence) of length  $n$  if and only if each  $\alpha$  orbit of  $\mathbf{Z}_n$  contains an even number of odd (even) integers. When this is the case, the number of  $\alpha$  sequences ( $\alpha'$  sequences) is  $2^{\eta(\alpha)}$ , where  $\eta(\alpha)$  is the number of  $\alpha$  orbits.

**Corollary:** If there exists an  $\alpha$  sequence ( $\alpha'$  sequence) of length  $n$ , then

$$\begin{aligned} & 4|n \text{ if } n \text{ is even} \\ & 4|n - 1(4 | n + 1) \text{ if } n \text{ is odd} \end{aligned}$$

If  $\underline{x}$  is an  $\alpha$  sequence ( $\alpha'$  sequence) then we can sometimes say a lot about its aperiodic autocorrelation function; e.g. the corollary to Theorem 1. In order to say something more precise about the aperiodic autocorrelation function of  $\underline{x}$  we impose further conditions on it. We therefore say that  $\underline{x}$  is an  $(\alpha, \beta)$  sequence or an  $(\alpha, \beta)'$  sequence if, respectively,

$$\begin{aligned} \alpha(\underline{x}) &= \underline{xz} \text{ and } \beta(\underline{x}) = \underline{xz} \\ \text{or } \alpha(\underline{x}) &= \underline{xz}' \text{ and } \beta(\underline{x}) = \underline{xz}', \end{aligned}$$

where  $\alpha$  and  $\beta$  are affine transformations on  $\mathbf{Z}_n$

We now have the following improvement to Theorem 1:

**Theorem 3:** If  $n$  is odd,  $\underline{x}$  an  $(\alpha, \beta)$  or  $(\alpha, \beta)'$  sequence,  $\alpha(i) = ai + r$ ,  $\beta(i) = bi + s$ , then,

$$\begin{aligned} F_{ab^{-1}i} &= \frac{1}{2}\{(1 + (-1)^{[i] + [ab^{-1}i]})F_i + (1 - (-1)^{[i] + [ab^{-1}i]})F_{n-i}\} \\ &= F_i \text{ or } F_{n-1} \end{aligned} \tag{13}$$

where in Equation (13),  $[m]$  denotes the integer  $m$  reduced modulo  $n$  and  $b^{-1}$  is the inverse of  $b$  in  $\mathbf{Z}_n$ .

*Proof* By Theorem 1:

$$\begin{aligned} E_{ak} &= F_{ak} + F_{n-ak} = (-1)^k(F_k - F_{n-k}) = (-1)^k\Theta_k \\ E_{bk} &= (-1)^k\theta_k \end{aligned}$$

where use was made of the fact that  $n$  is odd. Set  $k = [ab^{-1}j]$  in the second equation

$$E_{aj} = (-1)^{[ab^{-1}j]}\theta_{ab^{-1}j}$$

Then use the first of the above two equations to obtain:

$$\theta_{[ab^{-1}j]} = (-1)^{[j] + [ab^{-1}j]}\theta_j$$

Now  $\alpha(\underline{x}) = \beta(\underline{x})$  implies that  $\beta^{-1}\alpha(\underline{x}) = \underline{x}$ ; i.e.  $\underline{x}$  is invariant under the transformation:

$$\beta^{-1}\alpha(i) = ab^{-1}i + b^{-1}(r - s)$$

and so  $E_{ab^{-1}j} = E_j$ . Consequently,

$$\begin{aligned}
 F_{ab^{-1}j} &= \frac{1}{2}\{E_{ab^{-1}j} + \theta_{ab^{-1}j}\} \\
 &= \frac{1}{2}\{E_j + (-1)^{[j]+[ab^{-1}j]}\theta_j\} \\
 &= \frac{1}{2}\{(1 + (-1)^{[j]+[ab^{-1}j]})F_j + (1 - (-1)^{[j]+[ab^{-1}j]})F_{n-j}\} \\
 &= F_j \text{ or } F_{n-j}
 \end{aligned}$$

**Corollary:** If  $\underline{x}$  is an  $(\alpha, \beta)$  or  $(\alpha, \beta)'$  sequence of odd length  $n$ , then  $F_k(\underline{x})$ ,  $k = 1, 2, \dots, n-1$ , can assume at most  $2\eta(ab^{-1})$  values, where  $\eta(ab^{-1})$  is the number of orbits induced by  $i \rightarrow ab^{-1}i$ .

Theorem 3 is quite strong. For example, if  $i \rightarrow ab^{-1}i$  has only two orbits (the smallest possible number) then the corresponding  $(\alpha, \beta)$  sequence (or  $(\alpha, \beta)'$  sequence) will automatically be a Barker sequence!

*Example 1:*  $n = 7$ ;  $\alpha(i) = 3i+4$ ;  $\beta(i) = -i-1$ ; then  $ab^{-1} = -3 = 4$ . The orbits of  $i \rightarrow ab^{-1}i$  are  $\{0\}, \{1, 4, 2\}, \{3, 5, 6\}$  so that if  $\underline{x}$  is an  $(\alpha, \beta)'$  sequence. (It cannot be an  $(\alpha, \beta)$  sequence because  $4|n+1$ .) Then  $F_1 = F_3 = F_5$  and  $F_2 = F_4 = F_6$ . The only  $(\alpha, \beta)'$  sequences are  $++--+-$  and  $--++-+$ , both of which are Barker sequences.

If  $\underline{x}$  is an  $(\alpha, \beta)$  or  $(\alpha, \beta)'$  sequence then so is  $-\underline{x}$ . If these are the only two solutions, then we say that  $(\alpha, \beta)$  characterizes  $\underline{x}$ . It turns out that for  $n = 3, 5, 7, 11$ , and 13, there is a Barker sequence for which there exists a pair  $(\alpha, \beta)$  that completely characterizes it as given in Table 1. Table 2 lists some  $(\alpha, \beta)$  sequences which we found through computer search.

The weakness of Theorem 3 is that it says nothing about how to choose  $\alpha, \beta$  in order to guarantee that there exists a corresponding  $(\alpha, \beta)$  or  $(\alpha, \beta)'$  sequence. We look at this problem in the next section.

Table 1

| n  | Barker Sequence           | $\alpha(i)$ | $\beta(i)$ |
|----|---------------------------|-------------|------------|
| 3  | - + +                     | $i + 1$     | $-i - 1$   |
|    | + + -                     | $i + 2$     | $-i - 1$   |
| 5  | + + + - +                 | $3i$        | $-i - 1$   |
|    | + - + + +                 | $3i + 2$    | $-i - 1$   |
| 7  | + + + - - + -             | $3i + 4$    | $-i - 1$   |
|    | - + - - + + +             | $3i + 5$    | $-i - 1$   |
| 11 | - - - + + + - + + - +     | $2i + 4$    | $-i - 1$   |
|    | + - + + - + + + - - -     | $2i + 8$    | $-i - 1$   |
| 13 | + + + + + - - + + - + - + | $10i + 1$   | $-i - 1$   |
|    | + - + - + + - - + + + + + | $10i + 8$   | $-i - 1$   |

### III On the Existence of $(\alpha, \beta)$ Sequences

We can give a necessary and sufficient condition for the existence of an  $(\alpha, \beta)$  sequence or  $(\alpha, \beta)'$  sequence by introducing an appropriate graph.

**Definition 1:** The directed  $(\alpha, \beta)$  graph is defined as follows:

1. The set of vertices is  $V = \mathbf{Z}_n$
2. There is a branch with tail ‘i’ and head ‘j’ if and only if  $j = \alpha(i)$  or  $j = \beta(i)$ . Hence, the set of arcs is  $A = \{(i, \alpha(i)) \mid i \in \mathbf{Z}_n\} \cup \{(i, \beta(i)) \mid i \in \mathbf{Z}_n\}$

Recall that for  $n$  odd, if there exists an  $(\alpha, \beta)$  sequence (or an  $(\alpha, \beta)'$  sequence) then  $4|n - 1$  (or  $4|n + 1$ ). This motivates the following definition.

Table 2

| <b>n</b> | $\alpha(i)$ | $\beta(i)$ | <b>Sequence</b>   | <b>F</b> |
|----------|-------------|------------|---|----------|
| 9        | $i + 4$     | $-i - 1$   | ++++++--+-+   | 5        |
| 15       | $11i + 2$   | $-i - 1$   | +-+-+--+-----++-  | 3        |
| 17       | $4i$        | $13i + 6$  | +--+++-+++-+---+-++   | 4        |
| 19       | $8i + 4$    | $-i - 1$   | --+-+-+++-+----+----+   | 3        |
| 21       | $8i + 17$   | $-i - 1$   | ++-----+----+---+---+   | 3        |
| 33       | $10i + 10$  | $-i - 1$   | +--+--++++-+----+---+---+--++<br>++++   | 7        |
| 35       | $6i + 27$   | $-i - 1$   | --+-----+----+---+----+---+----+----+<br>-+---++                                  | 7        |
| 39       | $25i + 25$  | $-i - 1$   | -+-----+----+---+----+---+----+----+<br>+--+-++-+++                               | 7        |
| 45       | $26i + 26$  | $-i - 1$   | ++-----+----+---+----+---+----+----+<br>+----+----+---+----+                      | 7        |
| 57       | $37i + 37$  | $-i - 1$   | +----+----+---+----+---+----+---+----+<br>-+----+----+---+----+---+----+---+----+ | 7        |

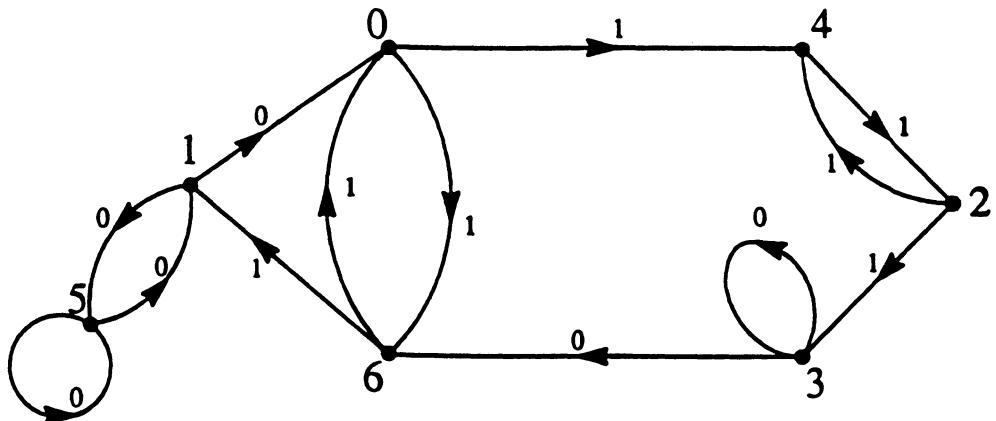
**Definition 2:** The weighted directed  $(\alpha, \beta)$  graph is obtained by assigning to the arc  $(i, j)$  of the directed  $(\alpha, \beta)$  graph the weight:

$$\left( i + \frac{n-1}{2} \right) \bmod 2$$

The weight is to be considered as an element in the finite field  $\mathbf{F}_2$ .

**Theorem 4:** There exists an  $(\alpha, \beta)$  or  $(\alpha, \beta)'$  sequence of odd length  $n$  if and only if the sum (in  $\mathbf{F}_2$ ) of the weights of the arcs making up any cycle in the corresponding weighted directed  $(\alpha, \beta)$  graph is 0. When this is the case, the number of  $(\alpha, \beta)$  or  $(\alpha, \beta)'$  sequences is  $2^g$  where  $g$  is the number of components of the graph.

**Example 2:** Let  $n = 7$ ;  $\alpha(i) = 3i+4$ ;  $\beta(i) = -i-1$ ; then the corresponding weighted directed  $(\alpha, \beta)$  graph is:



**Figure 1:  $(\alpha, \beta)$ -Graph of Example 2**

It may now be verified that every cycle in this graph has  $\mathbf{F}_2$ -weight 0. There are only two  $(\alpha, \beta)'$  sequences which are the Barker sequences of length 7 reported in Table 1.

**Example 3:** Let  $n = 15$ ;  $\alpha(i) = 11i + 2$ ;  $\beta(i) = -i - 1$ , then the corresponding graph is presented in Figure 2. Again, it may be verified that all the cycles have  $\mathbf{F}_2$ -weight 0. There are four  $(\alpha, \beta)$  sequences, one of which is reported in Table 2.

We can obtain an equivalent form of Theorem 4 in terms of matrices as follows: we consider our sequences as being over  $\mathbf{F}_2$  by replacing  $-1$  by 1 and 1 by 0. Then  $\underline{X}$  is an  $(\alpha, \beta)$  sequence if

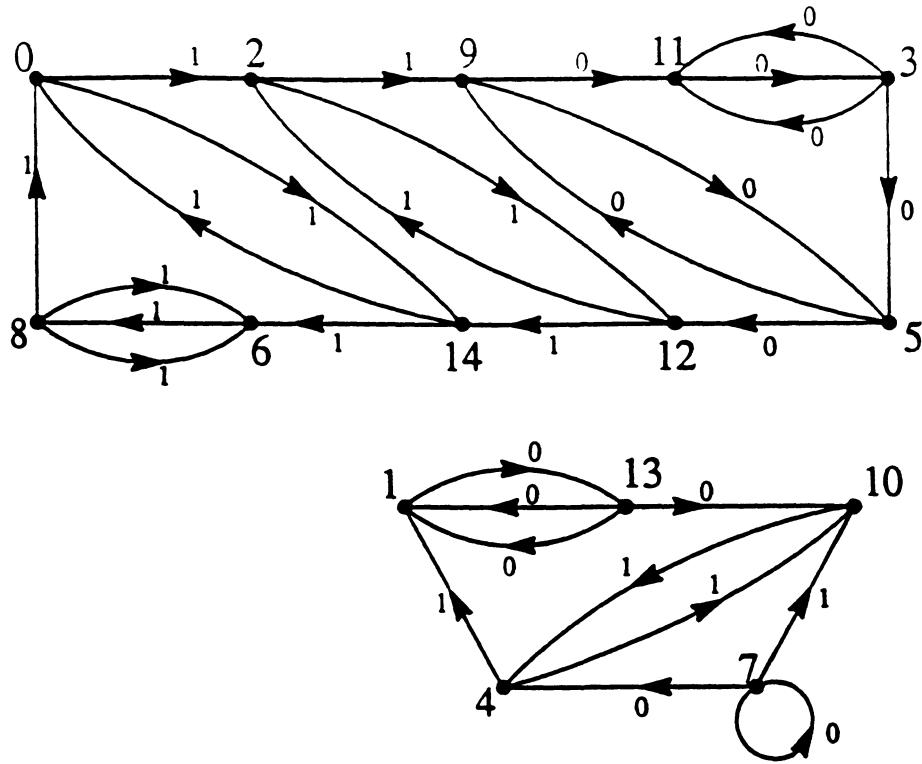
$$\alpha(\underline{X}) = \underline{X} + \underline{Z} \text{ and } \beta(\underline{X}) = \underline{X} + \underline{Z}$$

where now

$$Z_i = i \bmod 2, \quad i \in \mathbf{Z}_n.$$

If  $A$  is the matrix representation of  $\alpha$ , i.e.  $A$  has a 1 in position  $(\alpha(i), i)$ ,  $i \in \mathbf{Z}_n$ , and 0's elsewhere, and if  $B$  is the matrix representation of  $\beta$ , then the above conditions become

$$\underline{X}(I + A) = \underline{Z} \text{ and } \underline{X}(I + B) = \underline{Z}$$



**Figure 2:  $(\alpha, \beta)$ -Graph of Example 3**

or equivalently:

$$X[I + A, I + B] = [\underline{Z}, \underline{Z}] \quad (14)$$

where  $I$  is the  $n \times n$  identity matrix. It now follows that there exists an  $(\alpha, \beta)$  sequence if and only if

$$\text{rank} \begin{bmatrix} \underline{Z} & \underline{Z} \\ I + A & I + B \end{bmatrix} = \text{rank} [I + A, I + B] \quad (15)$$

For  $(\alpha, \beta)'$  sequences, we simply replace  $\underline{Z}$  by  $\underline{Z}'$  in Equation (15) where

$$Z'_i = (i + 1) \bmod 2, i \in \mathbf{Z}_n.$$

We collect these results as:

**Theorem 5:** If  $A, B, \underline{Z}$  and  $\underline{Z}'$  are as defined above then,

1. If  $4|n - 1$ , then there exists an  $(\alpha, \beta)$  sequence if and only if:

$$\text{rank} \begin{bmatrix} \underline{Z} & \underline{Z} \\ I + A & I + B \end{bmatrix} = \text{rank} [I + A, I + B]$$

and

2. If  $4|n + 1$ , then there exists an  $(\alpha, \beta)'$  sequence if and only if

$$\text{rank} \begin{bmatrix} \underline{Z}' & \underline{Z}' \\ I+A & I+B \end{bmatrix} = \text{rank} [I+A, I+B]$$

Those familiar with graph theory will be able to see that Theorems 4 and 5 are equivalent. We immediately have the following corollary:

**Corollary:** If there exists an  $(\alpha, \beta)$  or  $(\alpha, \beta)'$  sequence of odd length  $n$ , then the total number of such sequences is  $2^n - \mu$  where  $\mu = \text{rank} [I+A, I+B]$ .

*Example 4:* Let  $n = 5$ ;  $\alpha(i) = 3i$ ;  $\beta(i) = -i - 1$ , then the matrix  $[I+A, I+B]$  is easily computed to be,

$$[I+A, I+B] = \begin{bmatrix} 0000010001 \\ 0110001010 \\ 0010100000 \\ 0101001010 \\ 0001110001 \end{bmatrix}$$

The rank of this matrix is four. Since  $[\underline{Z}, \underline{Z}] = [0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0]$  is a row of  $[I+A, I+B]$ , then clearly Equation (15) is satisfied and so there exists an  $(\alpha, \beta)'$  sequence of length 5. The  $(\alpha, \beta)$  sequences over  $\mathbf{F}_2$  are 0 0 0 1 0, 1 1 1 0 1; both of which are Barker sequences.

## IV Conclusion

In this paper, we introduced the notion of an  $(\alpha, \beta)$  sequence of length  $n$  where  $\alpha, \beta$  are affine transformations on  $\mathbf{Z}_n$ . Theorem 3 shows that we can say a lot about the aperiodic correlations of an  $(\alpha, \beta)$  sequence. In some cases, an  $(\alpha, \beta)$  sequence is automatically a Barker sequence. In particular we were able to characterize the binary Barker sequences as  $(\alpha, \beta)$  sequences. By computer search we were able to find other  $(\alpha, \beta)$  sequences with good aperiodic correlation. (These are reported in Table 2.)

In Section III, we obtained necessary and sufficient conditions on  $\alpha$  and  $\beta$  to ensure the existence of an  $(\alpha, \beta)$  sequence. Unfortunately, those results are difficult to apply, and so we are not able to draw any precise conclusions about  $(\alpha, \beta)$  sequences for large values of  $n$ .

## References

- [1] C.E. Cook and M. Bernfield, *Radar Signals*, New York: Academic Press, 1967.
- [2] R.H. Barker, "Group synchronizing of binary digital systems", in *Communication Theory*, W. Jackson (ed.), pp 273-287, 1953.
- [3] J. Storer and R. Turyn, "On binary sequences" in *Proceedings of the American Mathematical Society*, Vol. 12, pp 394-399, 1961.
- [4] R. Turyn, "Sequences with small correlations" in *Error Correcting Codes*, H. B. Mann (ed.), John Wiley and Sons, 1968.

- [5] J.L. Massey and J.J. Uhran, Jr., *Final Report for Multi-path Study*, published by the Department of Electrical Engineering of the University of Notre Dame, 1969.
- [6] S.W. Golomb and R.A. Scholtz, “Generalized Barker sequences”, *IEEE Trans. on Information Theory*, Vol. IT-11, No. 4, pp 533-537, 1965.
- [7] M.J.E. Golay, “A class of finite binary sequences with alternate auto-correlation values equal to zero”, *IEEE Trans. on Information Theory*, Vol. IT-18, No. 3, p 449, 1972.
- [8] G.E. Séguin, *Binary Sequences with Specified Correlation Properties*, Ph.D. Dissertation, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, 1971.

# Fast Bounded-Distance Decoding of the Nordstrom-Robinson Code

Feng-Wen Sun

Department of Mathematics  
and Computing Science,  
Eindhoven University of Technology,  
5600 MB Eindhoven,  
The Netherlands

Henk C.A. van Tilborg

\*Department of Mathematics  
and Computing Science,  
Eindhoven University of Technology,  
5600 MB Eindhoven,  
The Netherlands.

## Abstract

Based on the two-level squaring construction of the Reed-Muller code, a bounded-distance decoding algorithm for the Nordstrom-Robinson code is given. This algorithm involves 199 real operations, which is less than one half of the computational complexity of the known maximum-likelihood decoding algorithms for this code. The algorithm also has exactly the same effective error coefficient as the maximum-likelihood decoding, so that its performance is only degraded by a negligible amount.

## I Introduction

The  $(16, 256, 6)$  Nordstrom-Robinson code is a practical nonlinear double-error-correcting code. It can also be used as a vector quantizer for encoding random waveforms (see [1]). The Nordstrom-Robinson code consists of the  $[16, 5, 8]$  first-order Reed-Muller code and seven of its cosets. The  $[16, 5, 8]$  Reed-Muller code can be decoded by the (fast) Hadamard transform (FHT) [14]. Based on the general supercode decoding method of [6], the Nordstrom-Robinson code can be decoded by performing the FHT eight times.

Recently, the Kerdock codes, and thus in particular the Nordstrom-Robinson code, were found [10, 11] to be the images of linear codes over  $\mathbf{Z}_4$ , the set of integers mod 4. This new perspective leads to a soft-decision decoding algorithm of the Kerdock codes by performing *complex*-valued FHT's of smaller size than that of the *real*-valued FHT's of the supercode approach. However, the two approaches essentially have the same decoding complexity.

By observing that during the computation of the various FHT's some intermediate results can be shared, Adoul [1] was able to reduce the computational complexity of the decoding procedure to a total 432 real operations (304 additions and 128 comparisons).

While the problem of maximum-likelihood decoding of the Nordstrom-Robinson code is interesting in its own right, it may be advantageous for practical applications to use

---

\*This work is partially supported by the Dutch organization for Scientific Research.

a slightly suboptimal but much more efficient decoding algorithm. In the sequel such an algorithm will be described.

The proposed algorithm utilizes the two-level squaring construction of the Reed-Muller code [7], which is contained in the Nordstrom-Robinson code. We shall see that the two-level squaring construction can be viewed as a two-level ‘multilevel construction’ [17]. Thus, multistage decoding can be applied to decode up to half the minimum distance of the code.

The proposed method is a bounded-distance decoding algorithm, so that it has the same ‘error exponent’ as maximum-likelihood decoding. It also has exactly the same *effective error coefficient* as maximum-likelihood decoding, so its performance will be degraded only marginally. The decoding complexity of the algorithm is only 199 real operations.

When discussing the complexity of (decoding) algorithms only the total number of real additions and comparisons will be counted. As is quite usual in the literature, operations such as memory addressing, negation, taking the absolute value, multiplication by 2, as well as the checking of logical conditions and modulo 2 additions will be ignored [7, 8, 16].

While writing down words like “effective error coefficient” and “supercode”, memories come back to me (the second author) of the first time that I listened to Jim Massey’s lecture. This was almost 25 years ago in Belgium and Jim started his talk with the coding theorist’s pledge. The text of this pledge implied that coding theorists should keep the coding community small by using inaccessible jargon for well-known notions (“code” instead of “subset”, “word” instead of “vector”, etc.). Of course, the subsequent talk turned out to be as well-presented and enthusing as all later talks that I heard by Jim. When congratulating Jim on his sixtieth birthday, I would like him to know that his teaching ability, together with his warm personality and his sense of duty, have been a constant inspiration to me.

## II Some Properties of Supercode Decoding

Let  $C$  be an arbitrary binary code of blocklength  $n$ . In this section, the 0 and 1 bits in a codeword will be regarded as the real numbers 1 and  $-1$  respectively. Thus, a codeword of  $C$  will be a vector in  $n$ -dimensional Euclidean space. This translation is appropriate for the Gaussian channel and for vector quantization with a squared-error distortion measure. In the later sections, we shall use the same letter to denote a binary vector or its  $\pm$  representation and we shall switch from one representation to the other depending on the context.

The *Voronoi region*  $V(\mathbf{c})$  of a codeword  $\mathbf{c}$  is the set of all  $\mathbf{x} \in \mathbf{R}^n$  that are at least as close to  $\mathbf{c}$  as to any other codeword of  $C$ . Given a decoding algorithm for  $C$  (not necessarily a maximum-likelihood decoding algorithm), define as in [8] the *decision region*  $D(\mathbf{c})$  of  $\mathbf{c}$  as the set of all  $\mathbf{x} \in \mathbf{R}^n$  that are decoded to  $\mathbf{c}$ . The error-correcting radius of  $\mathbf{c}$ , denoted  $\epsilon(\mathbf{c})$ , is the radius of largest sphere that can be inscribed into the decision region of  $\mathbf{c}$ . The error-correcting radius of the code, denoted by  $\epsilon$ , is the minimum value of  $\epsilon(\mathbf{c})$  among all codewords. Evidently  $D(\mathbf{c})$ ,  $\epsilon(\mathbf{c})$  and  $\epsilon$  depend on the decoding algorithm. For a maximum-likelihood decoding algorithm the decision regions and the error-correcting radius of the code are Voronoi regions and  $d/2$  respectively, where  $d$  is the minimum distance of the code. If the radius  $\epsilon$  of a suboptimal decoding algorithm is equal to  $d/2$ , one often calls it a *bounded-distance decoding algorithm*.

If a point lies both on the sphere of radius  $\epsilon$  centered at  $\mathbf{c}$  and on the boundary of  $D(\mathbf{c})$ , it will be called an *effective boundary point* of  $\mathbf{c}$ . The number of effective boundary points of  $\mathbf{c}$  is called the *effective error coefficient* of  $\mathbf{c}$ , denoted  $\text{eff}(\mathbf{c})$ . Note that for at least one codeword  $\mathbf{c}$   $\text{eff}(\mathbf{c})$  must be nonzero [8]. The effective error coefficient of the code, denoted  $\text{eff}(C)$ , is the average value of  $\text{eff}(\mathbf{c})$ , i.e.,

$$\text{eff}(C) = \frac{1}{|C|} \sum_{\mathbf{c} \in C} \text{eff}(\mathbf{c}).$$

In [9], Forney introduced the concept of *geometrically uniform signal sets*. For a geometrically uniform signal set, the Voronoi regions of all points have exactly the same shape. In this situation different points have equal error probability when a maximum-likelihood decoding algorithm is applied. In a similar way one may call a decoding algorithm *geometrically uniform* if all decision regions  $D(\mathbf{c})$  are congruent to each other. Although this is an interesting concept, we shall not discuss it in this paper. Instead, we will give a theorem about the effective error coefficient of the supercode decoding algorithm.

Let the binary codes  $C_i$ ,  $1 \leq i \leq m$ , have minimum distance  $d_i$ , and let the union  $C$  of these codes have minimum distance  $d$ . Further, let  $\mathbf{A}_i$  denote a decoding algorithm for  $C_i$ ,  $1 \leq i \leq m$ . Then one can apply the general supercode decoding method of Conway and Sloane [6] to decode  $C$ : first decode the received word with respect to each code  $C_i$  by means of decoding algorithm  $\mathbf{A}_i$ ; then compare the  $m$  survivors and choose the one closest to the received word as the most likely transmitted word in  $C$ . This supercode decoding procedure will be referred to as Algorithm  $\mathbf{A}$ .

**Theorem 1** *If each  $\mathbf{A}_i$ ,  $1 \leq i \leq m$ , is a bounded-distance decoding algorithm (for  $C_i$ ), then Algorithm  $\mathbf{A}$  is also a bounded-distance decoding algorithm (for  $C$ ).*

*Further, if the minimum distance  $d$  of  $C$  is strictly smaller than every  $d_i$ , algorithm  $\mathbf{A}$  will have an effective error coefficient that is equal to that of a maximum-likelihood decoding algorithm.*

**Proof:** Let  $\mathbf{x}$  be a point in  $\mathbf{R}^n$  at distance less than  $d/2$  to one of the codewords in  $C$ , say to  $\mathbf{c}_1$ . Without loss of generality, assume that  $\mathbf{c}_1$  is in  $C_1$ . Then

$$d(\mathbf{c}_1, \mathbf{x}) < d/2 \leq d_1/2,$$

Algorithm  $\mathbf{A}_1$  will choose  $\mathbf{c}_1$  as output. Assume that algorithm  $\mathbf{A}_i$ ,  $i > 1$ , chooses  $\mathbf{c}_i$  as output. Then either  $\mathbf{c}_i = \mathbf{c}$  or (by the triangle inequality),

$$d(\mathbf{c}_i, \mathbf{x}) \geq d/2.$$

This proves that there can be at most one codeword  $\mathbf{c}_i$  at distance less than  $d/2$  from  $\mathbf{x}$ , which shows that the error-correcting radius  $\epsilon$  is equal to  $d/2$  and that algorithm  $\mathbf{A}$  is a bounded-distance decoding algorithm.

Now assume that the minimum distance  $d$  of  $C$  is strictly smaller than every  $d_i$ . Let  $\mathbf{b}$  be a vector on the boundary of  $D(\mathbf{c}_1)$  and also on the sphere of radius of  $d/2$  centered around  $\mathbf{c}_1$ . Because  $d_1$  is strictly larger than  $d$ , algorithm  $\mathbf{A}_1$  will choose  $\mathbf{c}_1$  as output without ambiguity. Since  $\mathbf{b}$  is on the boundary of  $D(\mathbf{c}_1)$ , one of the algorithms  $\mathbf{A}_i$ ,  $i \geq 2$ ,

will output a vector that is as close to  $\mathbf{b}$  as  $\mathbf{c}_1$  is. Clearly, each  $\mathbf{A}_i$ ,  $i \geq 2$ , only outputs codewords in  $C$ . So, the above implies that  $\mathbf{b}$  has distance  $d/2$  to at least two codewords of  $C$ . Thus,  $\mathbf{b}$  must be on the boundary of  $V(\mathbf{c}_1)$  and must also be counted when calculating the effective error coefficient for maximum-likelihood decoding. The opposite inclusion is trivially true.  $\square$

### III Decoding the Nordstrom-Robinson Code

Let  $V_4$  denote the binary vector space of dimension four and let  $V_4^e$  denote the  $[4, 3, 2]$  even weight code in  $V_4$ . Further, let  $A$  be the  $[4, 2, 2]$  binary code with generator matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

and let  $\mathbf{g} = (1, 1, 1, 1)$ . Consider the following binary code of blocklength 16.

$$C_1 = \{(\mathbf{a} + b_1\mathbf{g}, \mathbf{a} + b_2\mathbf{g}, \mathbf{a} + b_3\mathbf{g}, \mathbf{a} + b_4\mathbf{g}) \mid \mathbf{a} \in A, \mathbf{b} \in V_4^e\}. \quad (1)$$

Clearly this code is linear with cardinality  $4 \times 2^3 = 2^5$ . Further, if  $\mathbf{a}$  is nonzero it must have weight 2 and the corresponding codeword will have weight  $4 \times 2 = 8$ , independent of the choice of  $\mathbf{b}$ . When  $\mathbf{a} = \mathbf{0}$  then one gets the all-zero and the all-one vectors or codewords of weight  $2 \times 4 = 8$ . This shows that  $C_1$  is in fact the  $[16, 5, 8]$  first-order Reed-Muller code.

The construction above is in fact equivalent to the two-level squaring construction of the Reed-Muller codes described in [7]. It is also a special case of the generalized concatenated codes construction by Blokh and Zyablov [3] and by Zinov'ev [18]. In some recent literature, the generalized concatenated codes are referred to as *multilevel codes* [4, 13, 17]. Multilevel codes can be decoded by the *multistage decoding algorithm* [4, 13, 17].

Instead of decoding directly with respect to  $C_1$ , we shall first decode with respect to the code

$$C'_1 = \{(\mathbf{a} + b_1\mathbf{g}, \mathbf{a} + b_2\mathbf{g}, \mathbf{a} + b_3\mathbf{g}, \mathbf{a} + b_4\mathbf{g}) \mid \mathbf{a} \in A, \mathbf{b} \in V_4\}. \quad (2)$$

Notice that  $C'_1$  consists of the union of  $C_1$  and a coset of  $C_1$ . Assume that a decoding algorithm for  $C'_1$  applied to a vector  $\mathbf{x}$  yields the vector

$$(\hat{\mathbf{a}} + \hat{b}_1\mathbf{g}, \hat{\mathbf{a}} + \hat{b}_2\mathbf{g}, \hat{\mathbf{a}} + \hat{b}_3\mathbf{g}, \hat{\mathbf{a}} + \hat{b}_4\mathbf{g}) \quad (3)$$

We shall accept the above estimate for  $\hat{\mathbf{a}}$ . The next step is to find the closest codeword  $\hat{\mathbf{c}} \in C_1$  to  $\mathbf{x}$  among the eight candidates

$$(\hat{\mathbf{a}} + b_1\mathbf{g}, \hat{\mathbf{a}} + b_2\mathbf{g}, \hat{\mathbf{a}} + b_3\mathbf{g}, \hat{\mathbf{a}} + b_4\mathbf{g}), \quad (4)$$

with  $\mathbf{b} \in V_4^e$ . The resulting codeword will be taken as the final estimate. Of course, if  $(\hat{b}_1, \hat{b}_2, \hat{b}_3, \hat{b}_4)$  itself is already in  $V_4^e$  one gets  $\hat{\mathbf{b}} = \mathbf{b}$ . Otherwise, one obtains  $\mathbf{b}$  from  $\hat{\mathbf{b}}$  by inverting the least reliable coordinate  $\hat{b}_i$ . This is in fact the idea of Wagner decoding [15] applied to  $\hat{\mathbf{b}}$ .

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |

Table 1: The coset leaders of the Nordstrom-Robinson code

As follows from the general theory of multilevel decoding [17], the above decoding procedure is a bounded-distance decoding algorithm, which means that the algorithm always outputs the closest codeword whenever the received point is within the error-correcting radius of the code.

The Nordstrom-Robinson code consists of the union of the  $[16, 5, 8]$  Reed-Muller code with seven of its cosets. The coset leaders are given as rows in Table 1.

The strategy for decoding the Nordstrom-Robinson code is to apply the decoding algorithm of the Reed-Muller code that is described above to the eight cosets of the Reed-Muller code. Then the metrics of the eight survivors are compared to find the closest codeword in the Nordstrom-Robinson code. The theorem in Section II guarantees that the bounded-distance property of the Reed-Muller decoding algorithm still holds for this Nordstrom-Robinson decoding algorithm.

Let us look at the complexity of the decoding algorithm, assuming that binary phase-shift keying (BPSK) is the modulation scheme being used. The algorithm is divided into three steps.

### Step 1: the precomputation

For an additive white Gaussian channel, finding the codeword closest to a received point  $\mathbf{x} = (x_1, x_2, \dots, x_{16}) \in \mathbf{R}^{16}$  is equivalent to finding the codeword with the largest inner product with  $\mathbf{x}$ , because the codewords are represented in the  $\pm 1$  notation. At the precomputation stage, we calculate the inner product between the segments  $(x_{4i+1}, x_{4i+2}, x_{4i+3}, x_{4i+4})$  and every vector in  $V_4$  for  $i = 0, 1, 2, 3$  as the *confidence values* of these combinations. Note that a binary four-tuple and its complement have confidence values with the same magnitude but opposite sign. So we need to calculate only  $x_{4i+1} \pm x_{4i+2} \pm x_{4i+3} \pm x_{4i+4}$ . Three additions find  $x_{4i+1} + x_{4i+2} + x_{4i+3} + x_{4i+4}$ . The remaining confidence values can be recursively calculated with seven additions by using the appropriate Gray code array as depicted in Table 2.

Each row of the Gray code array differs from its adjacent rows only in one position. Thus the confidence value of the corresponding vector can be obtained by adding or subtracting  $2x_j$  for some  $j$ . Because the multiplication by 2 is not counted as an operation, in agreement with the convention at the end of Section I, the confidence values of the binary four-tuples can be obtained with 10 additions. So at the precomputation stage  $4 \times 10$  real additions are needed. This accounts for Step 1 in Table 3.

|   |   |   |
|---|---|---|
| + | + | + |
| + | + | - |
| + | - | - |
| + | - | + |
| - | - | + |
| - | - | - |
| - | + | - |
| - | + | + |

Table 2: The eight-point Gray code array

### Step 2: decoding the subcodes

We shall now calculate the complexity of decoding  $\mathbf{x}$  with respect to each of the cosets of  $C_1$ . Each  $\mathbf{a} \in A$  defines 16 possible codewords in  $C'_1$  (see (2)). When a  $b_i$  changes in value, the confidence value of  $\mathbf{a} + b_i\mathbf{g}$  just changes in sign. Thus, among the 16 possible codewords in  $C'_1$ , the one that has the largest inner product with  $\mathbf{x}$  is the one with the  $b_i$ 's such that each  $\mathbf{a} + b_i\mathbf{g}$  has a positive confidence value. Then, in three additions, we add the positive confidence values  $\mathbf{a} + b_i\mathbf{g}$ ,  $i = 1, 2, 3, 4$ , and call the outcome the *confidence value* of  $\mathbf{a}$ . Let  $\mathbf{a}$  run over all four vectors in  $A$ , compare the confidence value of each  $\mathbf{a}$  and choose the one with largest confidence value, say

$$(\hat{\mathbf{a}} + \hat{b}_1\mathbf{g}, \hat{\mathbf{a}} + \hat{b}_2\mathbf{g}, \hat{\mathbf{a}} + \hat{b}_3\mathbf{g}, \hat{\mathbf{a}} + \hat{b}_4\mathbf{g}), \quad (5)$$

as the survivor. This step concludes the decoding with respect to the [16, 5, 8] Reed-Muller code. It involves  $3 \times 4 = 12$  additions and three comparisons (see Steps 2 i and 2 ii in Table 3).

If  $(\hat{b}_1, \hat{b}_2, \hat{b}_3, \hat{b}_4)$  is in  $V_4^c$ , then the corresponding codeword will be the survivor of the coset. Otherwise, one needs to find, as discussed before, the value of  $i$ , say  $i^*$ , for which  $\hat{\mathbf{a}} + \hat{b}_i\mathbf{g}$ ,  $1 \leq i \leq 4$ , has the smallest confidence value and invert  $\hat{b}_{i^*}$ . The scalar product between the resulting vector and  $\mathbf{x}$  can be obtained by subtracting two times the confidence values of  $\hat{\mathbf{a}} + \hat{b}_{i^*}\mathbf{g}$  from the confidence value of the survivor. Thus the second step of decoding [16, 5, 8] involves three comparisons and one addition (see Step 2 iii in Table 3).

In total, the decoding of  $\mathbf{x}$  with respect to the [16, 5, 8] Reed-Muller code  $C_1$  involves  $12 + 1 = 13$  additions and  $3 + 3 = 6$  comparisons. When decoding  $\mathbf{x}$  with respect to one of the other seven cosets  $C_i$  of  $C_1$  in the Nordstrom-Robinson code one first inverts the confidence values of the coordinates where the coset leader (see Table 1) of that coset has coordinates equal to 1 and then applies the above decoding method. Therefore decoding with respect to the eight cosets involves  $13 \times 8 = 104$  additions and  $6 \times 8 = 48$  comparisons.

### Step 3: comparing the survivors

Seven more comparisons (Step 3) will yield the best estimate among the eight survivors. Altogether the decoding of the Nordstrom-Robinson code involves 199 real operations (144 additions and 55 comparisons).

The complete decoding algorithm described in this section is summarized in Table 3.

Let  $\mathbf{x}$  be a received word.

- |               |   |                    |
|---------------|---|--------------------|
| <b>Step 1</b> | Compute the confidence value of all vectors in $V_4$ for each segment $(x_{4i+1}, x_{4i+2}, x_{4i+3}, x_{4i+4})$ , $1 \leq i \leq 4$ , using the Gray code array.   | $4 \times 10 = 40$ |
| <b>Step 2</b> | For each of the cosets $C_i$ of $C_1$ do:   | 8 ×                |
| i)            | for each of the four $\mathbf{a}$ 's in $A$ determine its confidence value and the corresponding vector $\mathbf{b}$ ;  | $(3 \times 4 +$    |
| ii)           | choose the survivor (see (5)); this is the closest codeword in $C'_i$ to $\mathbf{x}$ ;   | 3 +                |
| iii)          | if $\mathbf{b} \notin V_4^c$ , find $i^*$ which minimizes the confidence value of $\hat{\mathbf{a}} + \hat{b}_i \mathbf{g}$ , invert $\hat{b}_i$ and subtract twice this confidence value from that of $\hat{\mathbf{a}} + \hat{b}_{i^*} \mathbf{g}$ ; this gives the closest codeword in $C_i$ to $\mathbf{x}$ . | $3 + 1) = 152$     |
| <b>Step 3</b> | Compare the eight decoding results to find the closest codeword to $\mathbf{x}$ in the Nordstrom-Robinson code.   | 7                  |
- 

199

Table 3: Decoding the Nordstrom-Robinson code

## IV Remarks

The most efficient bounded-distance decoding of the extended Golay code involves 455 operations [2], i.e.  $455/12 \approx 40$  per information bit. The Nordstrom-Robinson code is also a rate- $1/2$  code, but the computational complexity per information bit is  $199/8 \approx 25$  operations, which is about 60% of the decoding complexity of the Golay code. For the vector quantization of an independent, identically distributed Gaussian process, the Nordstrom-Robinson code in case of maximum-likelihood decoding has a performance about only 0.1 dB away from the performance of the extended Golay code [1]. How well the suboptimal decoding algorithms presented here will perform if the code is used for vector quantization remains a topic for further research.

## References

- [1] J. P. Adoul, “Fast ML decoding algorithm for the Nordstrom-Robinson code,” *IEEE Trans. on Inform. Theory*, vol. 33, pp. 931-933, 1987.
- [2] O. Amrani, Y. Be’ery, A. Vardy , F. W. Sun, and H. C. A. van Tilborg, “The Leech lattice and the Golay code: bounded-distance decoding and multilevel constructions,” accepted by *IEEE Trans. on Inform. Theory*.
- [3] E. L. Blokh and V. V. Zyablov, “Coding of generalized concatenated codes,” *Prob. Perekhodch. Inform.*, vol. 10, no. 3, pp. 218-222, 1974.

- [4] A. R. Calderbank, "Multilevel trellis codes and multistage decoding," *IEEE Trans. Commun.*, vol. 37, pp. 222-229, 1989.
- [5] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. on Inform. Theory*, vol. 28, pp. 227-232, 1982.
- [6] J. H. Conway and N. J. A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. on Inform. Theory*, vol. 32, pp. 41-50, 1986.
- [7] G. D. Forney Jr. "Coset codes-Part II: Binary lattices and related codes," *IEEE Trans. on Inform. Theory*, vol. 34, pp. 1152-1187, 1988.
- [8] G. D. Forney, Jr. "A bounded-distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Trans. on Inform. Theory*, vol. 35, pp. 906-909, July 1989.
- [9] G. D. Forney, Jr., "Geometrically uniform codes," *IEEE Trans. on Inform. Theory*, vol. 37, pp. 1241-1260, Sept. 1991.
- [10] G. D. Forney, Jr., N. J. A. Sloane, and M. D. Trott, "The Nordstrom-Robinson code is the binary image of the octacode," *Proceedings 1992 DIMACS/IEEE Workshop on Coding and Quantization*, 1993.
- [11] A. R. Hammons, Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane and P. Solé, "The  $Z_4$ -Linearity of Kerdock, Preparata, Goethals and related codes," to appear in *IEEE Trans. on Inform. Theory*.
- [12] J. Hong and M. Vetterli, "Computing  $m$  DFT's over  $GF(q)$  with one DFT over  $GF(q^m)$ ," *IEEE Trans. on Inform. Theory*, vol. 39, pp. 271-274, 1993.
- [13] H. Imai and S. Hirakawa, "Multilevel coding method using error-correcting codes," *IEEE Trans. on Inform. Theory*, vol. IT-23, pp. 371-377, 1977.
- [14] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, Amsterdam: North-Holland, 1977.
- [15] R. A. Silverman and M. Balser, "Coding for a constant data rate source," *IRE Trans. Inform. Theory*, vol. PGIT-4, pp. 50-63, 1954.
- [16] J. Snyders and Yair Be'ery, "Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. on Inform. Theory*, vol. 35, pp. 963-975, 1989.
- [17] J. Wu and D. J. Costello, Jr. "New multilevel codes over  $GF(q)$ ," *IEEE Trans. on Inform. Theory*, vol. 38, pp. 933-935, 1992.
- [18] V. A. Zinov'ev, "Generalized cascade codes," *Prob. Peredach. Inform.*, vol. 12, No. 1, pp. 2-9, 1976.
- [19] V. A. Zinov'ev and V. V. Zyablov, "Decoding of nonlinear generalized cascade codes," *Prob. Peredach. Inform.*, vol. 14, No. 2, pp. 110-114, 1978.

# Binary Convolutional Codes Revisited

Gottfried Ungerboeck  
IBM Research Division, Zurich Laboratory  
CH-8803 Rueschlikon, Switzerland

## Abstract

No general algebraic method is known for the construction of convolutional codes with optimum distance properties. Good binary rate- $k/n$  convolutional codes have been found for small values of  $k$  and  $n$  by various computer search methods, where *good* means that the free Hamming distances of these codes closely approach or are equal to established upper bounds. In this paper, we report on another attempt to find convolutional codes by computer search. The considered codes are produced by encoders that resemble closely those employed for trellis-coded modulation, i.e.,  $k$ -tuples of information bits are first expanded into binary  $(k+1)$ -tuples by a rate- $\tilde{k}/(k+1)$  convolutional encoder, where  $1 \leq \tilde{k} \leq k$ ; the  $(k+1)$ -tuples are then encoded into binary  $n$ -tuples by a memoryless mapper (= block encoder), whose mapping rule is based on set-partitioning of binary  $n$ -tuples with respect to Hamming distance. Code searches have been performed for rates  $k/n$  in the range  $1 \leq k < n \leq 8$ , and for code memories in the range  $2 \leq v \leq 10$ . New codes with larger free Hamming distance than known codes were found for the rates  $4/5$ ,  $5/6$ ,  $6/7$ , and  $7/8$ .

## I Introduction

In Jim Massey's exemplary scientific career, early work on convolutional codes has played a significant role. Convolutional codes were first proposed by Elias (1954). The concept was then further developed by Wozencraft (1957), and by Wyner and Ash (1963). In 1963, Massey developed a class of multiple-error-correcting convolutional codes and devised a simple-to-implement decoding method for them called threshold decoding [1]. The advent of the Viterbi decoding algorithm in 1967 [2] marked another important milestone in the development and practical application of convolutional codes. The usefulness of trellis diagrams for describing convolutional codes became known and the algebraic structure of convolutional codes was explored. Massey and Sain [3] determined the conditions under which the operation of a convolutional encoder can be inverted without feedback, thus avoiding *catastrophic* error propagation. The algebraic structure of encoders for convolutional codes over finite fields could subsequently be completely clarified by Forney [4].

A general algebraic method for the construction of convolutional codes has not yet been discovered. The codes now in common use have been obtained by various methods of computer

search. Bussgang (1965), Costello (1969), Odenwalder (1970), and Bahl and Jelinek (1971) have pioneered the search for good convolutional codes. We will mainly refer in this paper to the binary rate- $k/n$  convolutional codes found by Larsen [5] for rates  $1/2, 1/3, 1/4$ , Paaske [6] for rates  $2/3, 3/4$ , and Daut et al. [7] for rates  $2/3, 3/4, 1/5, 2/5, 3/5, 4/5, 1/6, 5/6, 1/7-6/7, 1/8, 3/8, 5/8, 7/8$ . The free Hamming distances of these codes approach closely and for rates where  $k < n - 1$  almost always achieve the established upper bounds. This essentially limits the chance for finding better codes to the set of codes with rate  $(n - 1)/n$ . The codes mentioned above have been found by direct code search for given rates. Other useful codes, usually with higher rates, have been obtained by puncturing lower-rate codes [8],[9]. The punctured codes exhibit generally lower free Hamming distances than the codes found by direct search, but offer other advantages for their application.

In this paper, we report on one further attempt to find good binary convolutional codes by computer search. We will present a few new codes with rates  $4/5, 5/6, 6/7$ , and  $7/8$ , which exhibit larger free Hamming distances than known codes for these rates. The considered codes are generated by encoders that resemble closely those employed for trellis-coded modulation [10]. In general, this restricts the search for codes to a smaller class of codes among which the best possible codes cannot always be found.

(It has been brought to the attention of this author that Ø. Ytrehus has recently also reported on the results of new code searches for high-rate convolutional codes [15]. Unfortunately, the paper of Ytrehus was not available to this author when the final version of this paper had to be completed.)

## II Upper Bounds on Free Hamming Distance of Convolutional Codes

Upper bounds on the free distance achievable with given code parameters provide a useful guideline during code search. For convolutional codes, upper distance bounds are obtained from block-code bounds by considering finite-length segments of convolutional code sequences originating and ending (*terminated*) in the all-zero state as block codes, determining the block-code bounds for all possible segment lengths and associated rates, and taking the minimum thereof. For this study, we have used the Heller bound [11] and the improved version by Odenwalder [12], which both are based on the Plotkin bound for linear block codes. These bounds are surprisingly tight for low to moderately high convolutional code rates, as can be expected from the fact that the Plotkin bound is tight for low block-code rates. For higher convolutional-code rates, such as  $5/6$  and larger, we found also cases, where convolutional-code bounds based on the Hamming (sphere packing) block-code bound were smaller than the Heller and Odenwalder bounds. A version of the Elias block-code bound adapted from [13] for finite block-code parameters was also employed. However, for the considered convolutional-code parameters no case was found, where a tighter bound than the Plotkin- and Hamming-type bounds was achieved.

### III Convolutional Codes Viewed as Trellis Codes Based on Set Partitioning

Figure 1 shows the encoder structure considered for this study. This structure should be familiar from trellis-coded modulation. First, the  $k$ -tuples of information input bits are expanded to binary  $(k+1)$ -tuples by a rate- $\tilde{k}/(\tilde{k}+1)$  convolutional encoder. Then, instead of mapping the  $(k+1)$ -tuples into modulation signals, these bits are encoded into binary  $n$ -tuples by a linear block encoder (mapper) with a generator matrix  $G_{k+1/n}$  chosen according to the set-partitioning concept.

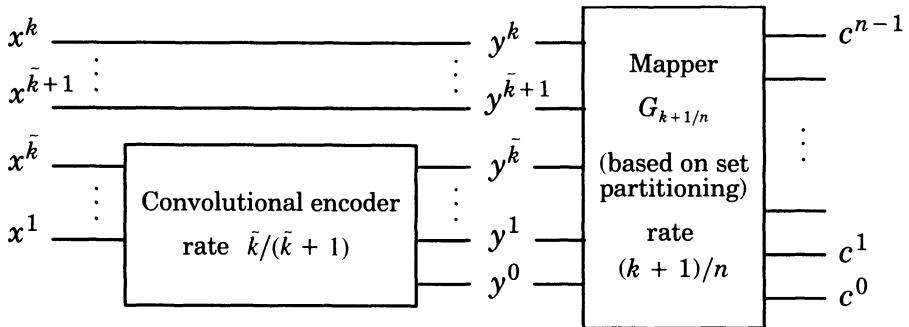


Figure 1. Set-partitioning encoder for convolutional rate- $k/n$  code.

### IV Set Partitioning of Binary $n$ -Tuples

Several methods could be considered to set-partition binary  $n$ -tuples. We adopt the simplest method, which follows directly from standard Reed-Muller codes, and begin with explaining the partitioning of the set of all possible binary 8-tuples. The generator matrix of the trivial ( $N=8$ ,  $K=8$ ,  $d=1$ ) Reed-Muller code is given by

$$G_{8/8} = \left[ \begin{array}{l|c|c} & & \text{row weight} \\ \hline g_7 : & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 8 \\ g_6 : & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 4 \\ g_5 : & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 4 \\ g_4 : & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 4 \\ g_3 : & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 2 \\ g_2 : & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 \\ g_1 : & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ g_0 : & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \right]$$

## The linear mapping

$$[y^7, y^6, \dots, y^0] G_{8/8} \Rightarrow [c^7, c^6, \dots, c^0]$$

exhibits the desired set-partitioning structure. In the language of trellis coded modulation, we obtain the tree of subsets (=cosets), with maximally increasing minimum intra-subset distances  $\Delta_0 \leq \Delta_1 \leq \dots$  illustrated in Figure 2, where  $\mathcal{G}_{8-\ell/8}$  for  $0 \leq \ell \leq 7$  denotes the set of 8-tuples (=block code) generated by  $G_{8/8}$  with  $y^{\ell-1} = \dots = y^0 = 0$ .

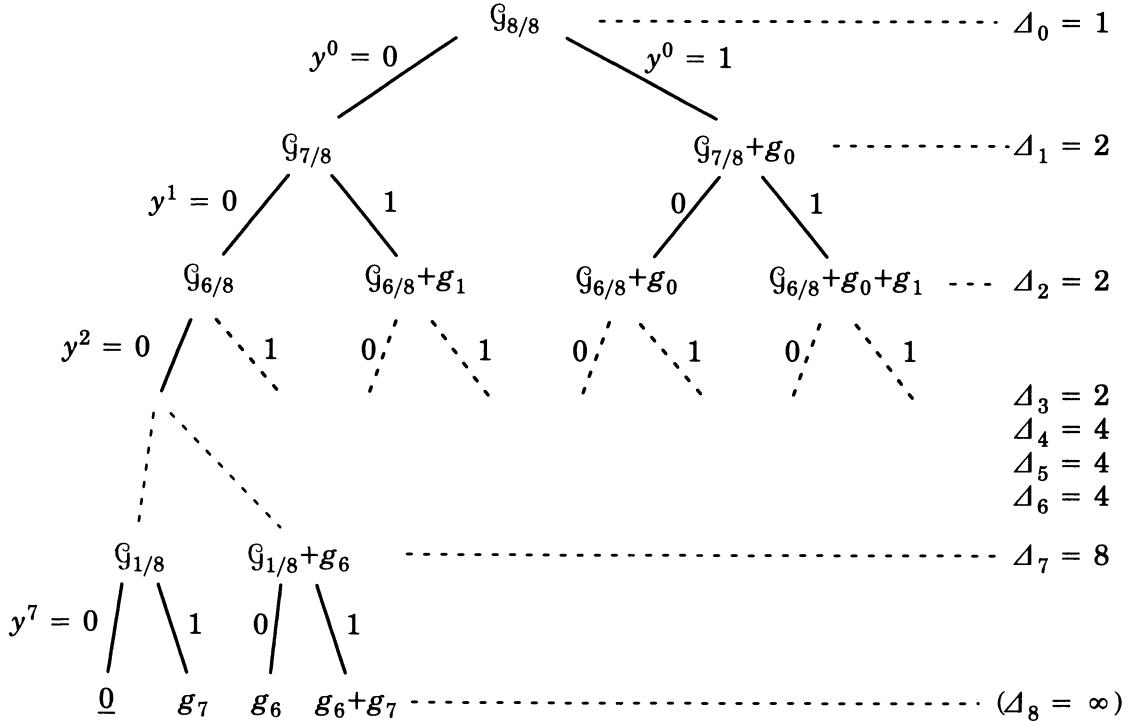


Figure 2. Set partitioning of binary 8-tuples.

The generator matrix  $G_{8/8}$  will be used for rate  $k/n = 7/8$  codes. For smaller values of  $n$  in the range  $5 \leq n \leq 7$  and  $k = n - 1$  we delete from  $G_{8/8}$  the first  $8 - n$  rows and columns, to obtain  $G_{n/n}$ . For values of  $n$  in the range  $2 \leq n \leq 4$  we proceed in a similar manner using the matrix

$$G_{4/4} = \begin{bmatrix} g_3 : & 1 & 1 & 1 & 1 \\ g_2 : & 0 & 1 & 0 & 1 \\ g_1 : & 0 & 0 & 1 & 1 \\ g_0 : & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{row weight}$$

Thus, we obtain the following matrices and corresponding sequences of minimum intra-subset distances:

|          |  |
|----------|--|
| rate 1/2 | $G_{2/2} : \Delta_0 = 1, \Delta_1 = 2$   |
| rate 2/3 | $G_{3/3} : \Delta_0 = 1, \Delta_1 = \Delta_2 = 2$  |
| rate 3/4 | $G_{4/4} : \Delta_0 = 1, \Delta_1 = \Delta_2 = 2, \Delta_3 = 4$  |
| rate 4/5 | $G_{5/5} : \Delta_0 = 1, \Delta_1 = \Delta_3 = \Delta_4 = 2, \Delta_2 = 4$                                     |
| rate 5/6 | $G_{6/6} : \Delta_0 = 1, \Delta_1 = \Delta_2 = \Delta_3 = 2, \Delta_4 = \Delta_5 = 4$                          |
| rate 6/7 | $G_{7/7} : \Delta_0 = 1, \Delta_1 = \Delta_2 = \Delta_3 = 2, \Delta_4 = \Delta_5 = \Delta_6 = 4$               |
| rate 7/8 | $G_{8/8} : \Delta_0 = 1, \Delta_1 = \Delta_3 = \Delta_4 = 2, \Delta_2 = \Delta_5 = \Delta_6 = 4, \Delta_7 = 8$ |

With these generator matrices good rate- $(n - 1)/n$  codes could be found, which in some cases are better than known codes. For codes with  $k < n - 1$ , we have used matrices  $G_{k+1/n}$  obtained from  $G_{n/n}$  by deleting the last  $n - k - 1$  rows. For example, this leads to

$$\text{rate } 3/7 \quad G_{4/7} : \Delta_0 = 2, \Delta_1 = \Delta_2 = \Delta_3 = 4 .$$

Although some new interesting codes could also be found for these rates, the results of code searches were generally less satisfying than those for rate- $(n - 1)/n$  codes and will not be reported in this paper.

It is nevertheless simpler to continue the following discussion for the general case of rate- $k/n$  codes. Let  $w_c(y^k, \dots, y^0)$  be the Hamming weight of the  $n$ -tuple  $[c^{n-1}, \dots, c^0] = [y^k, \dots, y^0]G_{k+1/n}$ . We note that

$$\Delta_\ell = \underset{\text{all } [y^k, \dots, y^\ell] \neq \underline{0}}{\text{Min}} w_c(y^k, \dots, y^\ell, 0, \dots, 0) , \quad 0 \leq \ell \leq k ,$$

and define  $\Delta_{k+1} = \infty$ .

## V Free Hamming Distance and Code Search

As for trellis coded modulation, we assume for the rate- $\tilde{k}/(\tilde{k} + 1)$  convolutional encoders the systematic form with feedback shown in Figure 3, where  $v$  is the number of binary storage elements corresponding to the code memory. The usual polynomial notation will be employed where appropriate.

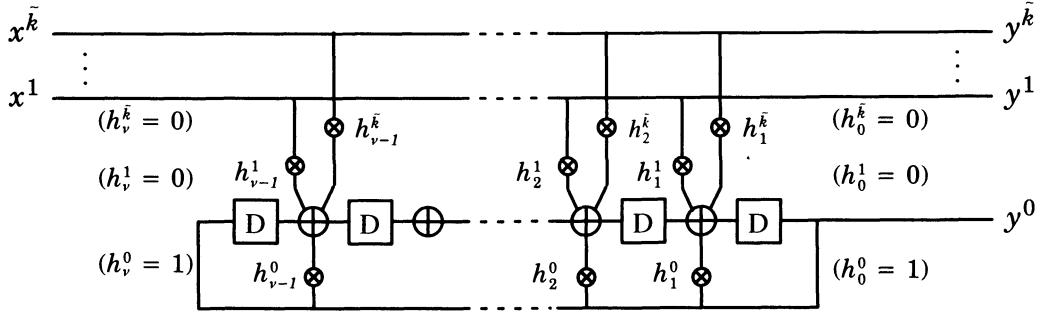


Figure 3. Systematic rate- $\tilde{k}/(\tilde{k} + 1)$  convolutional encoder with feedback ( $\nu \geq 2$ ).

Let  $\mathcal{Y}^{\tilde{k}}$  be the set of code sequences  $\underline{y}^{\tilde{k}}(D) = [y^{\tilde{k}}(D), \dots, y^0(D)]$  produced by the rate- $\tilde{k}/(\tilde{k} + 1)$  encoder. These sequences satisfy the parity-check equation

$$H^{\tilde{k}}(D) y^{\tilde{k}}(D) + \dots + H^1(D) y^1(D) + H^0(D) y^0(D) = 0(D) ,$$

where the parity-check polynomials for  $\nu \geq 2$  take the form

$$H^i(D) = 1 + h_{\nu-1}^i D^{\nu-1} + \dots + h_1^i D + 0 , \quad 1 \leq i \leq \tilde{k} ,$$

$$H^0(D) = 1 + h_{\nu-1}^0 D^{\nu-1} + \dots + h_1^0 D + 1 .$$

Let  $\mathcal{C}^n$  be the set of code sequences  $\underline{c}(D) = [c^{n-1}(D), \dots, c^0(D)]$  obtained at the output of the mapper with constrained inputs  $\underline{y}^{\tilde{k}}(D) \in \mathcal{Y}^{\tilde{k}}$ , and unconstrained inputs  $y^k(D), \dots, y^{\tilde{k}+1}(D)$  for  $\tilde{k} < k$ . The free Hamming distance between the code sequences of  $\mathcal{C}^n$  is given by

$$d_f = \text{Min}[\Delta_{\tilde{k}+1}, d_{f,\tilde{k}}] ,$$

where  $\Delta_{\tilde{k}+1}$  is the minimum *parallel transition distance* between multiple  $n$ -tuples associated with each transition in the trellis diagram of  $\mathcal{Y}^{\tilde{k}}$ , and  $d_{f,\tilde{k}}$  represents the minimum *non-parallel transition distance* between  $n$ -tuples associated with paths in the trellis diagram that diverge to different states and later remerge, and hence differ during more than one transition. With

$$w_c^*(y^{\tilde{k}}, \dots, y^0) = \underset{\text{all } \underline{y}^{\tilde{k}}, \dots, \underline{y}^{\tilde{k}+1}}{\text{Min}} w_c(y^{\tilde{k}}, \dots, y^{\tilde{k}+1}, y^{\tilde{k}}, \dots, y^0) ,$$

$d_{f,\tilde{k}}$  becomes

$$d_{f,\tilde{k}} = \underset{\text{all } \underline{y}^{\tilde{k}}(D) \neq \underline{0}(D) \in \mathcal{Y}^{\tilde{k}}}{\text{Min}} \sum_j w_c^*(y_j^{\tilde{k}}, \dots, y_j^0) .$$

We also observe that  $w_c^*(y^{\tilde{k}}, \dots, y^q \neq 0, 0, \dots, 0) \geq \Delta_q$  for  $0 \leq q < \tilde{k}$ , and that hence the special form of the parity-check polynomials ( $h_v^0 = h_0^0 = 0$ ;  $h_v^i = h_0^i = 0$ ,  $1 \leq i \leq \tilde{k}$ ) ensures  $d_{f,\tilde{k}} \geq 2\Delta_1$  [10], provided the polynomials  $H^{\tilde{k}}(D), \dots, H^1(D)$  are linearly independent.

The search for codes was performed in a manner similar as described in [10]. First for given values  $k/n$  and  $v \geq 2$ , the upper distance bound  $d_f^{UB}$  was computed. Then for several values of  $\tilde{k}$  up to the smallest value, for which  $\Delta_{\tilde{k}+1} \geq d_f^{UB}$ , codes with the largest parallel transition distance  $d_{f,\tilde{k}}$  were determined by a code search program. Finally, among the codes achieving the same largest free Hamming distance the code obtained with the smallest value of  $\tilde{k}$  was selected. No attempt was made to evaluate error coefficients. A detailed description of the code search program would exceed the scope of this paper. In principle, the program explores the entire space of possible parity-check polynomials. A set of rules is employed for the early rejection of codes or sets of codes, which cannot achieve good distance or whose distance could not be larger than the distance of the best code among the codes already tested. One of these rules is to reject all codes with parity-check polynomials  $H^\ell(D), \dots, H^0(D)$ , and untested polynomials  $H^{\tilde{k}}(D), \dots, H^{\ell+1}(D)$ , if the *level*- $\ell$  subcode obtained with  $y^{\tilde{k}}(D) = \dots, y^{\ell+1}(D) = 0(D)$  does not achieve larger distance than the largest value of  $d_{f,\tilde{k}}$  found for codes examined earlier. The distances  $d_{f,\tilde{k}}$  are determined by the bidirectional search algorithm in the form described by Larsen [14].

## VI Results of Search for Rate- $(n-1)/n$ Codes

In the code tables given below parity-check polynomials are represented in octal notation, for example,  $H^1(D) = D^6 + D^5 + D^3 + D$  is specified as  $oh^1 = 0152$ . Values of  $d_f$  that are equal to  $d_f^{UB}$  are highlighted with bold numbers.

Rate 1/2 codes

| $v$ | $\tilde{k}$ | $oh^1$ | $oh^0$ | $\Delta_{\tilde{k}+1}$ | $d_{f,\tilde{k}}$ | $d_f (d_f^{UB})$ |
|-----|-------------|--------|--------|------------------------|-------------------|------------------|
| 2   | 1           | 0002   | 0005   | $\infty$               | 5                 | <b>5</b> (5)     |
| 3   | 1           | 0004   | 0013   | $\infty$               | 6                 | <b>6</b> (6)     |
| 4   | 1           | 0004   | 0023   | $\infty$               | 7                 | 7 (8)            |
| 5   | 1           | 0014   | 0043   | $\infty$               | 8                 | <b>8</b> (8)     |
| 6   | 1           | 0042   | 0117   | $\infty$               | 10                | <b>10</b> (10)   |
| 7   | 1           | 0152   | 0205   | $\infty$               | 10                | 10 (11)          |
| 8   | 1           | 0152   | 0435   | $\infty$               | 12                | <b>12</b> (12)   |
| 9   | 1           | 0524   | 1013   | $\infty$               | 12                | 12 (13)          |
| 10  | 1           | 0644   | 2123   | $\infty$               | 14                | <b>14</b> (14)   |

equ. to Larsen [5]

### Rate 2/3 codes

| $\nu$ | $\tilde{k}$ | $oh^2$ | $oh^1$ | $oh^0$ | $\Delta_{\tilde{k}+1}$ | $d_{f,\tilde{k}}$ | $d_f (d_f^{UB})$ |
|-------|-------------|--------|--------|--------|------------------------|-------------------|------------------|
| 2     | 1           | —      | 0002   | 0005   | 2                      | 5                 | 2 (4)            |
| 3     | 2           | 0004   | 0002   | 0011   | $\infty$               | 4                 | <b>4</b> (4)     |
| 4     | 2           | 0014   | 0002   | 0021   | $\infty$               | 5                 | 5 (6)            |
| 5     | 2           | 0022   | 0006   | 0043   | $\infty$               | 6                 | <b>6</b> (6)     |
| 6     | 2           | 0066   | 0040   | 0105   | $\infty$               | 7                 | 7 (8)            |
| 7     | 2           | 0154   | 0036   | 0203   | $\infty$               | 8                 | <b>8</b> (8)     |
| 8     | 2           | 0316   | 0076   | 0401   | $\infty$               | 8                 | 8 (9)            |

Paaske [6]:  $d_f = 3$

equ. to Paaske [6]

### Rate 3/4 codes

| $\nu$ | $\tilde{k}$ | $oh^3$ | $oh^2$ | $oh^1$ | $oh^0$ | $\Delta_{\tilde{k}+1}$ | $d_{f,\tilde{k}}$ | $d_f (d_f^{UB})$ |
|-------|-------------|--------|--------|--------|--------|------------------------|-------------------|------------------|
| 2     | 1           | —      | —      | 0002   | 0005   | 2                      | 5                 | 2 (4)            |
| 3     | 2           | —      | 0004   | 0002   | 0011   | 4                      | 4                 | <b>4</b> (4)     |
| 4     | 2           | —      | 0004   | 0002   | 0021   | 4                      | 5                 | 4 (5)            |
| 5     | 3           | 0030   | 0014   | 0002   | 0041   | $\infty$               | 5                 | 5 (6)            |
| 6     | 3           | 0050   | 0022   | 0006   | 0103   | $\infty$               | 6                 | <b>6</b> (6)     |
| 7     | 3           | 0060   | 0032   | 0006   | 0201   | $\infty$               | 6                 | 6 (8)            |

(Daut [7],  $\nu = 1 : d_f = 2$ )

Daut [7]:  $d_f = 3$

equ. to Paaske [6]

### Rate 4/5 codes

| $\nu$ | $\tilde{k}$ | $oh^4$ | $oh^3$ | $oh^2$ | $oh^1$ | $oh^0$ | $\Delta_{\tilde{k}+1}$ | $d_{f,\tilde{k}}$ | $d_f (d_f^{UB})$ |
|-------|-------------|--------|--------|--------|--------|--------|------------------------|-------------------|------------------|
| 2     | 1           | —      | —      | —      | 0002   | 0005   | 2                      | 5                 | 2 (3)            |
| 3     | 1           | —      | —      | —      | 0004   | 0013   | 2                      | 6                 | 2 (4)            |
| 4     | 3           | —      | 0010   | 0004   | 0002   | 0021   | 4                      | 4                 | <b>4</b> (4)     |
| 5     | 3           | —      | 0010   | 0004   | 0002   | 0041   | 4                      | 4                 | 4 (5)            |
| 6     | 4           | 0060   | 0044   | 0014   | 0002   | 0101   | $\infty$               | 5                 | 5 (6)            |
| 7     | 4           | 0152   | 0104   | 0022   | 0006   | 0203   | $\infty$               | 6                 | 6 (7)            |

(Daut [7],  $\nu = 1 : d_f = 2$ )

equ. to Daut [7]

Daut [7]:  $d_f = 3$

better than Daut [7]

new

Rate 5/6 codes

| $\nu$ | $\tilde{k}$ | $oh^4$ | $oh^3$ | $oh^2$ | $oh^1$ | $oh^0$ | $\Delta_{\tilde{k}+1}$ | $d_{f,\tilde{k}}$ | $d_f(d_f^{UB})$ | (Daut [7], $\nu = 1 : d_f = 2$ ) |
|-------|-------------|--------|--------|--------|--------|--------|------------------------|-------------------|-----------------|----------------------------------|
| 2     | 1           | —      | —      | —      | 0002   | 0005   | 2                      | 5                 | 2 (3)           | equ. to Daut [7]                 |
| 3     | 1           | —      | —      | —      | 0004   | 0013   | 2                      | 6                 | 2 (4)           | Daut [7]: $d_f = 3$              |
| 4     | 3           | —      | 0010   | 0004   | 0002   | 0021   | 4                      | 4                 | 4 (4)           |                                  |
| 5     | 3           | —      | 0010   | 0004   | 0002   | 0041   | 4                      | 4                 | 4 (6)           |                                  |
| 6     | 3           | —      | 0044   | 0014   | 0002   | 0101   | 4                      | 5                 | 4 (6)           | new                              |

Rate 6/7 codes

| $\nu$ | $\tilde{k}$ | $oh^4$ | $oh^3$ | $oh^2$ | $oh^1$ | $oh^0$ | $\Delta_{\tilde{k}+1}$ | $d_{f,\tilde{k}}$ | $d_f(d_f^{UB})$ | (Daut [7], $\nu = 1 : d_f = 2$ ) |
|-------|-------------|--------|--------|--------|--------|--------|------------------------|-------------------|-----------------|----------------------------------|
| 2     | 1           | —      | —      | —      | 0002   | 0005   | 2                      | 5                 | 2 (3)           | equ. to Daut [7]                 |
| 3     | 1           | —      | —      | —      | 0004   | 0013   | 2                      | 6                 | 2 (4)           |                                  |
| 4     | 3           | —      | 0010   | 0004   | 0002   | 0021   | 4                      | 4                 | 4 (4)           |                                  |
| 5     | 3           | —      | 0010   | 0004   | 0002   | 0041   | 4                      | 4                 | 4 (6)           |                                  |
| 6     | 3           | —      | 0044   | 0014   | 0002   | 0101   | 4                      | 5                 | 4 (6)           | new                              |

Rate 7/8 codes

| $\nu$ | $\tilde{k}$ | $oh^4$ | $oh^3$ | $oh^2$ | $oh^1$ | $oh^0$ | $\Delta_{\tilde{k}+1}$ | $d_{f,\tilde{k}}$ | $d_f(d_f^{UB})$ | (Daut [7], $\nu = 1 : d_f = 2$ ) |
|-------|-------------|--------|--------|--------|--------|--------|------------------------|-------------------|-----------------|----------------------------------|
| 2     | 1           | —      | —      | —      | 0002   | 0005   | 2                      | 5                 | 2 (2)           |                                  |
| 3     | 1           | —      | —      | —      | 0004   | 0013   | 2                      | 6                 | 2 (4)           |                                  |
| 4     | 3           | —      | 0010   | 0004   | 0002   | 0021   | 4                      | 4                 | 4 (4)           |                                  |
| 5     | 3           | —      | 0010   | 0004   | 0002   | 0041   | 4                      | 4                 | 4 (4)           |                                  |
| 6     | 3           | —      | 0044   | 0014   | 0002   | 0101   | 4                      | 5                 | 4 (6)           | new                              |

## VII Discussion of the Obtained Codes

It had to be expected that better codes than the known ones can hardly be found. The codes obtained for the rates 1/2, 2/3 (with one exception), and 3/4 (with only exception) are equivalent to the known codes. Surprisingly, with the assumed encoder structure it was not possible to achieve the distance of the best known codes for the following rates and code memories: (2/3,  $\nu=2$ ), (3/4,  $\nu=2$ ), (4/5,  $\nu=3$ ), and (5/6,  $\nu=3$ ). It appears that in these cases the larger distances of the known codes can be achieved with generator matrices based on different set partitioning. For the rates 4/5 to 7/8, new codes with distances, which in several cases achieve the upper bound, have been obtained. This represents currently our main result of "revisiting convolutional codes". Further investigations may be worthwhile. Many of the new codes exhibit parallel

transitions in their trellis diagrams. They may offer advantages in terms of decoding complexity versus coding gain in a comparison with known codes. This could be another topic for further study.

## References

- [1] J.L. Massey, *Threshold Decoding*, MIT Press, Cambridge, Massachusetts, 1963.
- [2] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, 1967.
- [3] J.L. Massey and M.K. Sain, "Inverses of linear circuits", *IEEE Trans. Comp.*, vol. C-17, pp. 330–337, 1968.
- [4] G.D. Forney, Jr., "Convolutional codes I: algebraic structure", *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, 1970.
- [5] K.J. Larsen, "Short convolutional codes with maximal free distance for rates 1/2, 1/3, and 1/4", *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 371–372, 1973.
- [6] E. Paaske, "Short binary convolutional codes with maximal free distances for rates 2/3 and 3/4", *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 683–689, 1974.
- [7] D.G. Daut, J.W. Modestino, and L.D. Wismer, "New short constraint length convolutional code constructions for selected rational rates", *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 794–800, 1982.
- [8] Y. Yasuda, K. Kashiki, and Y. Hirata, "High-rate punctured convolutional codes for soft decision Viterbi decoding", *IEEE Trans. Commun.*, vol. COM-32, pp. 315–318, 1984.
- [9] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications", *IEEE Trans. Commun.*, vol. COM-36, pp. 389–399, 1988.
- [10] G. Ungerboeck, "Channel coding with multilevel/phase signals", *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, 1982.
- [11] J.A. Heller, "Sequential decoding: Short constraint length convolutional codes", Jet Propulsion Lab., Calif. Inst. Techn., Pasadena, Space Program Summary 37–54, vol. 3, pp. 171–174, 1968.
- [12] J.P. Odenwalder, "Optimal decoding of convolutional codes", Ph.D. dissertation, Dept. Syst. Sci., Sch. Eng. Appl. Sci., Univ. California, 1970.
- [13] R.E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley Publ. Company, 1983.
- [14] K.J. Larsen, "Comments on 'An efficient algorithm for computing free distance' ", *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 437–439, 1972.
- [15] Ø. Ytrehus, "Binary convolutional codes of high rate", submitted to *IEEE Trans. Inform. Theory*.

# Some Reflections On The Interference Channel

Edward C. van der Meulen  
Department of Mathematics  
Katholieke Universiteit Leuven  
3001 Leuven, Belgium

## Abstract

This paper provides a short overview of the advances on the interference channel, which is a well-known channel in multiuser information theory, that have appeared in the literature since 1976. Several open problems are defined.

## I Introduction

In 1977 this author had a survey article [23] published in the *IEEE Transactions on Information Theory* on multiway channels in information theory. It described results in this field in the period up to 1976. The editor of the Transactions at the time was Jim Massey. It was he who invited me to publish that article in the Transactions and who handled my paper. Through his initiative and action, the field of multiuser information theory could get quite an exposure in the journal. At later occasions, at information and communication theory meetings, I interacted often with Jim on questions related to multiuser information theory. Several times Jim suggested to me to write another survey paper. I wrote follow-up papers on the broadcast channel and the multiaccess channel, but never yet on the interference channel. At a meeting in Essen in 1991, Jim showed particular interest in the connections between various achievable rate regions for the interference channel, when I gave a short expository lecture on the topic then. It is great pleasure for me to write this comprehensive survey on the interference channel at the occasion of the sixtieth birthday of James L. Massey, whose enthusiasm has stimulated me much during my career.

## II Preliminaries

An interference channel models the situation where two unrelated senders try to communicate separate information to two different receivers via a common channel. A channel of this type was first considered by Shannon [22] in the context of his analysis of the two-way channel. Not only may there be actual noise in the common channel, but the crosstalk in one direction represents by itself interference for the other sender-receiver pair. Such situations arise e.g. in personal and mobile communication. We still do not know to what extent crosstalk interference limits communication. We do know that coding can eliminate most of the effects of noise-interference. Therefore, we would like to understand the ultimate performance limits of an interference channel, both for practical and intellectual reasons. However, as we shall see below, despite some very hard work, this problem is still not solved.

In this survey we describe results on achievable performance rates for various kinds of interference channels. We consider both the discrete alphabet memoryless interference channel and the discrete-time additive white Gaussian noise interference channel.

A discrete memoryless interference channel with two senders and two receivers, denoted by  $\{\mathcal{X}_1 \times \mathcal{X}_2, w(y_1, y_2|x_1, x_2), \mathcal{Y}_1 \times \mathcal{Y}_2\}$ , consists of two finite input alphabets  $\mathcal{X}_1$  and  $\mathcal{X}_2$ , two finite output alphabets  $\mathcal{Y}_1$  and  $\mathcal{Y}_2$ , and a transition probability matrix  $\{w(y_1, y_2|x_1, x_2)\}$ . The transmission and reception points are all at different places. The goal is for sender 1 to communicate information to receiver 1 and for sender 2 to communicate information to receiver 2. Here,  $w(y_1, y_2|x_1, x_2)$  is the probability that the outputs  $y_1$  and  $y_2$  are received, given that the inputs  $x_1$  and  $x_2$  are transmitted. It gives rise to the marginal conditional probabilities  $w_1(y_1|x_1, x_2)$  and  $w_2(y_2|x_1, x_2)$ , which are the only important channel parameters from the viewpoint of communication performance, since the two receivers are isolated and cannot collaborate in the decoding of their outputs. The memoryless condition means that for sequences  $\mathbf{x}_a = (x_a^{(1)}, \dots, x_a^{(n)}) \in \mathcal{X}_a^n$  and  $\mathbf{y}_a = (y_a^{(1)}, \dots, y_a^{(n)}) \in \mathcal{Y}_a^n$ ,  $a = 1, 2$ ,

$$w^n(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}_1, \mathbf{x}_2) = \prod_{t=1}^n w(y_1^{(t)}, y_2^{(t)}|x_1^{(t)}, x_2^{(t)}).$$

Let  $\mathcal{M}_a = \{1, 2, \dots, M_a\}$ ,  $a = 1, 2$ , be message sets for senders 1 and 2, respectively. A code  $(n, M_1, M_2, \lambda)$  for an interference channel  $\mathcal{K}$  is a collection of  $M_1$  codewords  $\mathbf{x}_{1i} \in \mathcal{X}_1^n$ ,  $i \in \mathcal{M}_1$ ;  $M_2$  codewords  $\mathbf{x}_{2j} \in \mathcal{X}_2^n$ ,  $j \in \mathcal{M}_2$ ;  $M_1$  disjoint decoding sets  $B_{1i} \subset \mathcal{Y}_1^n$ ,  $i \in \mathcal{M}_1$ ; and  $M_2$  disjoint decoding sets  $B_{2j} \subset \mathcal{Y}_2^n$ ,  $j \in \mathcal{M}_2$ , such that

$$P_{e1} \triangleq \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} w_1^n(B_{1i}^c | \mathbf{x}_{1i}, \mathbf{x}_{2j}) \leq \lambda$$

$$P_{e2} \triangleq \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} w_2^n(B_{2j}^c | \mathbf{x}_{1i}, \mathbf{x}_{2j}) \leq \lambda,$$

where  $c$  denotes set-complementation.  $P_{e1}$  and  $P_{e2}$  are called the average error probabilities of the code  $(n, M_1, M_2, \lambda)$ .

A pair  $(R_1, R_2)$  of nonnegative real numbers is called an achievable rate pair for an interference channel  $\mathcal{K}$  if for any  $\eta > 0$ ,  $0 < \lambda < 1$ , and any sufficiently large  $n$ , there exists a code  $(n, M_1, M_2, \lambda)$  such that

$$\frac{1}{n} \log M_a \geq R_a - \eta, \quad a = 1, 2.$$

The capacity region of  $\mathcal{K}$ , denoted by  $\mathcal{C}$ , is the set of all achievable rate pairs. An achievable rate region (also called inner bound) is any convex subset of the capacity region. An outer bound on the capacity region is a subset of the first quadrant of the  $(R_1, R_2)$ -plane that contains the capacity region. Throughout the paper we assume not only that the interference channel is memoryless, but also that it is frame (or block)-synchronous, i.e., that the beginnings of the codewords sent by the transmitters coincide.

The main aim of the research is to find a computable expression (i.e., a single-letter characterization) for the capacity region of the interference channel. For the general discrete memoryless interference channel and the Gaussian interference channel such an expression has not yet been found, and only inner and outer bounds on the capacity region are available. These bounds are typically computable expressions themselves, based on single-letter mutual

information functions. Throughout this paper the notation for entropy  $H(X)$ , conditional entropy  $H(Y|X)$ , mutual information  $I(X;Y)$ , and conditional mutual information  $I(X;Y|Z)$  will follow that of the book by Gallager [13].

### III Early Results for the Discrete Memoryless Interference Channel

Ahlswede [1], Sato [17] and Carleial [6] obtained inner bounds on the capacity region of the general discrete memoryless interference channel. Sato [17] obtained a useful outer bound to this capacity region. Ahlswede [2] found the capacity region of the so-called compound discrete memoryless interference channel and Sato [17] and Carleial [6] independently established the capacity region of the discrete memoryless interference channel with statistically equivalent outputs. These results were reported in [23] and will not be restated here. After this, Benzel [3] established the capacity region of a class of discrete degraded interference channels. His result can be summarized as follows. Let  $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{Y}_1 = \mathcal{Y}_2 = \mathcal{S} = \{0, 1, 2, \dots, s-1\}$ , where  $s$  is an arbitrary integer. Let the marginal transition probability distributions  $\{w_1(y_1|x_1, x_2)\}$  and  $\{w_2(y_2|x_1, x_2)\}$  be determined by the equations

$$Y_1 = X_1 + X_2 + V_1 \quad (1.1)$$

$$Y_2 = X_1 + X_2 + V_1 + V_2, \quad (1.2)$$

where  $+$  denotes addition modulo  $s$  and  $V_1, V_2$  are independent noise random variables defined over  $\mathcal{S}$  with certain distributions. Let  $S$  be a random variable defined on  $\mathcal{S}$ , independent of  $(V_1, V_2)$  with probability distribution  $P_S$ . Let

$$\begin{aligned} \mathcal{G}(S) = \{(R_1, R_2) : & \quad 0 \leq R_1 \leq H(S + V_1) - H(V_1) \\ & \quad 0 \leq R_2 \leq \log s - H(S + V_1 + V_2)\}. \end{aligned} \quad (2)$$

Then Benzel [3] proved that the capacity region of the discrete degraded interference channel described by (1.1) and (1.2) is given by

$$\mathcal{C} = \text{convex closure of } \cup_{P_S} \mathcal{G}(S). \quad (3)$$

In 1981, Han and Kobayashi [14] put forward a new achievable rate region for the general discrete memoryless interference channel, which includes the achievable rate regions established in [1], [17], and [6] and still is the best one to date. We now proceed to describe this region.

### IV The Han–Kobayashi Region

Consider a discrete memoryless interference channel  $\{\mathcal{X}_1 \times \mathcal{X}_2, w(y_1, y_2|x_1, x_2), \mathcal{Y}_1 \times \mathcal{Y}_2\}$ , denoted by  $\mathcal{K}$ , and introduce a modified interference channel  $\mathcal{K}_m$ , which differs from  $\mathcal{K}$  only in the way the quintuple  $\{\mathcal{X}_1 \times \mathcal{X}_2, w(y_1, y_2|x_1, x_2), \mathcal{Y}_1 \times \mathcal{Y}_2\}$  is used. Instead of two message sets  $\mathcal{M}_1$  and  $\mathcal{M}_2$ , consider now four message sets  $\mathcal{L}_a = \{1, \dots, L_a\}, \mathcal{N}_a = \{1, \dots, N_a\}, a = 1, 2$ , and a code  $(n, L_1, N_1, L_2, N_2, \lambda)$ , which is so defined that it enables sender 1 to send  $L_1$  private messages to receiver 1 and  $N_1$  common messages to both receivers, and sender 2 to send  $L_2$  private messages to receiver 2 and  $N_2$  common messages to both receivers, such that the average probability of a decoding error is at most  $\lambda$  for each receiver.

If the interference channel is used this way it is denoted by  $\mathcal{K}_m$ . A quadruple  $(S_1, T_1, S_2, T_2)$  of nonnegative real values is called an achievable rate vector for  $\mathcal{K}_m$  if for arbitrary  $\eta > 0$ ,  $0 < \lambda < 1$ , and any large  $n$ , there exists a code  $(n, L_1, N_1, L_2, N_2, \lambda)$  such that

$$\frac{1}{n} \log L_a \geq S_a - \eta \quad a = 1, 2$$

$$\frac{1}{n} \log N_a \geq T_a - \eta \quad a = 1, 2.$$

Clearly, if there is a code  $(n, L_1, N_1, L_2, N_2, \lambda)$  for  $\mathcal{K}_m$ , then there is a code  $(n, L_1 N_1, L_2 N_2, \lambda)$  for  $\mathcal{K}$ . Consequently, if  $(S_1, T_1, S_2, T_2)$  is an achievable rate vector for  $\mathcal{K}_m$ , then  $(S_1 + T_1, S_2 + T_2)$  is an achievable rate pair for  $\mathcal{K}$ .

Consider now auxiliary random variables  $Q, U_1, W_1, U_2, W_2$ , defined on arbitrary finite sets  $\mathcal{Q}, \mathcal{U}_1, \mathcal{W}_1, \mathcal{U}_2, \mathcal{W}_2$ , respectively, and let  $X_1, X_2, Y_1$ , and  $Y_2$  be random variables defined on  $\mathcal{X}_1, \mathcal{X}_2, \mathcal{Y}_1$ , and  $\mathcal{Y}_2$ , respectively. Let  $\mathcal{P}^*$  be the set of all  $Z = Q U_1 W_1 U_2 W_2 X_1 X_2 Y_1 Y_2$  such that

- i)  $U_1, W_1, U_2$ , and  $W_2$  are conditionally independent given  $Q$ ;
- ii)  $X_1 = f_1(U_1, W_1 \mid Q), X_2 = f_2(U_2, W_2 \mid Q)$ , where, for each  $q \in \mathcal{Q}$ ,  $f_1(\cdot \mid q) : \mathcal{U}_1 \times \mathcal{W}_1 \rightarrow \mathcal{X}_1$ , and  $f_2(\cdot \mid q) : \mathcal{U}_2 \times \mathcal{W}_2 \rightarrow \mathcal{X}_2$  are arbitrary deterministic functions, and  $f_1, f_2, \mathcal{Q}, \mathcal{U}_1, \mathcal{W}_1, \mathcal{U}_2, \mathcal{W}_2$  range over all possible choices;
- iii)  $\Pr \{Y_1 = y_1, Y_2 = y_2 \mid X_1 = x_1, X_2 = x_2\} = w(y_1, y_2 \mid x_1, x_2)$ .

Then  $Z = Q U_1 W_1 U_2 W_2 X_1 X_2 Y_1 Y_2 \in \mathcal{P}^*$  implies that  $X_1, X_2, Y_1, Y_2$  are random variables induced on  $\mathcal{X}_1, \mathcal{X}_2, \mathcal{Y}_1, \mathcal{Y}_2$  from  $U_1, W_1, U_2, W_2, Q$  via the test channel  $(f_1, f_2, w)$ .

For any  $Z \in \mathcal{P}^*$ , let  $\mathcal{S}(Z)$  be the set of all quadruples  $(S_1, T_1, S_2, T_2)$  of nonnegative real numbers such that

$$S_1 \leq I(U_1; Y_1 \mid W_1 W_1 Q), \quad (4.1)$$

$$T_1 \leq I(W_1; Y_1 \mid U_1 W_2 Q), \quad (4.2)$$

$$T_2 \leq I(W_2; Y_1 \mid U_1 W_1 Q), \quad (4.3)$$

$$S_1 + T_1 \leq I(U_1 W_1; Y_1 \mid W_2 Q), \quad (4.4)$$

$$S_1 + T_2 \leq I(U_1 W_2; Y_1 \mid W_1 Q), \quad (4.5)$$

$$T_1 + T_2 \leq I(W_1 W_2; Y_1 \mid U_1 Q), \quad (4.6)$$

$$S_1 + T_1 + T_2 \leq I(U_1 W_1 W_2; Y_1 \mid Q), \quad (4.7)$$

$$S_2 \leq I(U_2; Y_2 \mid W_1 W_2 Q), \quad (4.8)$$

$$T_1 \leq I(W_1; Y_2 \mid U_2 W_2 Q), \quad (4.9)$$

$$T_2 \leq I(W_2; Y_2 \mid U_2 W_1 Q), \quad (4.10)$$

$$S_2 + T_1 \leq I(U_2 W_1; Y_2 \mid W_2 Q), \quad (4.11)$$

$$S_2 + T_2 \leq I(U_2 W_2; Y_2 \mid W_1 Q), \quad (4.12)$$

$$T_1 + T_2 \leq I(W_1 W_2; Y_2 \mid U_2 Q), \quad (4.13)$$

$$S_2 + T_1 + T_2 \leq I(U_2 W_1 W_2; Y_2 \mid Q). \quad (4.14)$$

Let

$$\mathcal{S} = \text{closure of } \cup_{Z \in \mathcal{P}^*} \mathcal{S}(Z). \quad (5)$$

**Theorem 1** (Han and Kobayashi [14]) : Any element of  $\mathcal{S}$  is an achievable rate vector for the modified interference channel  $\mathcal{K}_m$ . Moreover,  $\mathcal{S}$  is convex.

$\mathcal{S}(Z)$  is a closed convex polyhedron in the four-dimensional Euclidean space. The seven mutual information functions on the right side of the inequalities (4.1) – (4.7) give rise to a polymatroid (in the terminology of combinatorics), and so do those on the right side of the inequalities (4.8) – (4.14). The set  $\mathcal{S}(Z)$  is the intersection of the independence polyhedra associated with these two polymatroids. As a consequence, an achievable rate region for the interference channel  $\mathcal{K}$  can be derived as follows. For any  $Z \in \mathcal{P}^*$ , let

$$\mathcal{R}(Z) = \{(R_1, R_2) : R_1 = S_1 + T_1, R_2 = S_2 + T_2 \text{ for some } (S_1, T_1, S_2, T_2) \in \mathcal{S}(Z)\}, \quad (6)$$

and define

$$\mathcal{R}^* = \text{closure of } \cup_{Z \in \mathcal{P}^*} \mathcal{R}(Z). \quad (7)$$

Moreover, let  $\mathcal{P}$  denote the set of all  $Z = Q U_1 W_1 U_2 W_2 X_1 X_2 Y_1 Y_2 \in \mathcal{P}^*$  such that  $Q = \phi$  ( $\phi$  is a constant), and define

$$\mathcal{R} = \text{convex closure of } \cup_{Z \in \mathcal{P}} \mathcal{R}(Z). \quad (8)$$

**Theorem 2** (Han and Kobayashi ([14])) : Any element of  $\mathcal{R}^*$  is achievable for the interference channel  $\mathcal{K}$ .  $\mathcal{R}^*$  is convex. Moreover,  $\mathcal{R}^*$  remains invariant if one imposes the following constraints on the cardinalities of the auxiliary sets :

$$\begin{aligned} |\mathcal{U}_1| &\leq |\mathcal{X}_1| + 2, & |\mathcal{W}_1| &\leq |\mathcal{X}_1| + 7, \\ |\mathcal{U}_2| &\leq |\mathcal{X}_2| + 2, & |\mathcal{W}_2| &\leq |\mathcal{X}_2| + 7, & |\mathcal{Q}| &\leq 11. \end{aligned}$$

Clearly, since  $\mathcal{R}^*$  is convex,  $\mathcal{R} \subset \mathcal{R}^*$ . Hence,  $\mathcal{R}$  is also an achievable rate region for  $\mathcal{K}$ .

Notice that the formulation of region  $\mathcal{R}$  involves a convex-hull operation whereas that of region  $\mathcal{R}^*$  uses a time-sharing parameter  $Q$ . In other parts of the literature on multiway channels most achievable rate regions are in terms of a convex-hull operation. Cover [11] used a time-sharing parameter in his formulation of an achievable rate region for the broadcast channel. Han and Kobayashi [14] prefer to use  $Q$  because the inverse inclusion  $\mathcal{R} \supset \mathcal{R}^*$  may not hold in general. They emphasize that the parameter  $Q$  is introduced by them to allow for simultaneous superposition coding rather than sequential superposition coding as e.g. used by Carleial [6]. Simultaneous superposition coding turns out to be more powerful than sequential coding for more complex systems such as the interference channel. For simpler systems such as the degraded broadcast channel the difference between the two disappears. A numerical example given in [14] for the Gaussian interference channel (see Section V) suggests that  $\mathcal{R}^*$  strictly extends  $\mathcal{R}$  for the general discrete memoryless interference channel. In [14] the claim is made that for multiway channels with more than one receiver it does make a difference whether the achievable rate region is formulated in terms of a time-sharing parameter rather than a convex hull operation and that for such multiway channels the former formulation may lead to a larger region than the latter one.

In [14, Theorem 4.1], Han and Kobayashi present a simple explicit expression for  $\mathcal{R}(Z)$ , which is easier to compute than deriving (6) from (4.1) – (4.14). That expression also reveals

the geometrical shape of  $\mathcal{R}(Z)$ . The formal description of it is rather involved, though. Therefore, for brevity's sake we only indicate the main inequalities here, and refer to [14] for specific formulas.

**Theorem 3** (Han and Kobayashi [14]) : For any  $Z \in \mathcal{P}^*$ , the region  $\mathcal{R}(Z)$  is equal to a polyhedron, consisting of all pairs  $(R_1, R_2)$  of nonnegative real numbers such that

$$\begin{aligned} R_1 &\leq \rho_1 \quad , \quad R_2 \leq \rho_2 \quad , \quad R_1 + R_2 \leq \rho_{12}, \\ 2R_1 + R_2 &\leq \rho_{10} \quad , \quad \text{and} \quad R_1 + 2R_2 \leq \rho_{20}, \end{aligned} \quad (9)$$

where  $\rho_1, \rho_2, \rho_{12}, \rho_{10}$ , and  $\rho_{20}$  are single-letter information-theoretic expressions given by (4.2) – (4.6) in [14].

Han and Kobayashi [14] also showed that the region  $\mathcal{R}^*$  contains the achievable rate regions previously established by Carleial [6] and Sato [17]. They first defined a simple subset  $\mathcal{R}_0^*$  of  $\mathcal{R}^*$ . For each  $Z \in \mathcal{P}^*$ , let  $\mathcal{R}_0(Z)$  be the set of all  $(R_1, R_2)$  such that

$$R_1 \leq \sigma_1 + I(U_1; Y_1 | W_1 W_2 Q) \quad (10.1)$$

$$R_2 \leq \sigma_2 + I(U_2; Y_2 | W_1 W_2 Q) \quad (10.2)$$

$$R_1 + R_2 \leq \sigma_{12} + I(U_1; Y_1 | W_1 W_2 Q) + I(U_2; Y_2 | W_1 W_2 Q), \quad (10.3)$$

where  $\sigma_1, \sigma_2$ , and  $\sigma_{12}$  are single-letter information-theoretic expressions given by (3.33) in [14]. Next define

$$\mathcal{R}_0^* = \text{closure of } \cup_{Z \in \mathcal{P}^*} \mathcal{R}_0(Z) \quad (11)$$

and

$$\mathcal{R}_0 = \text{convex closure of } \cup_{Z \in \mathcal{P}} \mathcal{R}_0(Z). \quad (12)$$

**Theorem 4** (Han and Kobayashi [14]) : For any  $Z \in \mathcal{P}^*, \mathcal{R}_0(Z) \subset \mathcal{R}(Z)$ . Hence, the inclusion relations  $\mathcal{R}_0 \subset \mathcal{R}_0^* \subset \mathcal{R}^*$  hold and  $\mathcal{R}_0$  and  $\mathcal{R}_0^*$  are achievable rate regions for  $\mathcal{K}$ .

Han and Kobayashi [14] showed that the regions established by Carleial [6] and Sato [17], denoted by  $\mathcal{R}_C$  and  $\mathcal{R}_S$ , respectively, are contained in  $\mathcal{R}_0$ . Concerning  $\mathcal{R}_C$ , they demonstrated that

$$\mathcal{R}_C = \text{convex closure of } \bigcup_{Z \in \mathcal{P}} \mathcal{R}_C(Z), \quad (13)$$

where for each  $Z, \mathcal{R}_C(Z)$  is a set of rate pairs  $(R_1, R_2)$  satisfying certain inequalities, similar to but more restrictive than (10), so that  $\mathcal{R}_C(Z) \subset \mathcal{R}_0(Z)$ . Hence  $\mathcal{R}_C \subset \mathcal{R}_0$ . In [14, Fig. 4] a graphical comparison is given of the regions  $\mathcal{R}_C(Z)$  and  $\mathcal{R}_0(Z)$ , showing that for an arbitrary choice of  $Z, \mathcal{R}_C(Z)$  is strictly contained in  $\mathcal{R}_0(Z)$ .

*Open problem 1 :* It is unknown whether  $\mathcal{R}^*$  is maximal, i.e., is also the capacity region of the discrete memoryless interference channel. Apart from several interesting special cases, no general converse theorem has been proved. The evaluation of the region  $\mathcal{R}^*$  remains prohibitively difficult. It is unknown whether in general  $\mathcal{R}^*$  strictly extends  $\mathcal{R}$ , although this is conjectured by Han and Kobayashi [14]. They also conjectured  $\mathcal{R} \neq \mathcal{R}_0$  and  $\mathcal{R}^* \neq \mathcal{R}_0^*$ ; these conjectures remain open.

## V The Gaussian Interference Channel

A discrete-time additive white Gaussian interference channel with noise variances  $N_1$  and  $N_2$  and input power constraints  $P_1$  and  $P_2$  has alphabets  $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{Y}_1 = \mathcal{Y}_2 = \mathcal{R}$ . Its channel operation is specified by

$$Y_1 = a_{11}X_1 + a_{21}X_2 + Z_1, \quad (14.1)$$

$$Y_2 = a_{12}X_1 + a_{22}X_2 + Z_2, \quad (14.2)$$

where  $Z_1, Z_2$  are independent zero-mean Gaussian additive noises with variances  $N_1$  and  $N_2$ , respectively, and the transmitted input sequences  $\mathbf{x}_{1i} \in \mathcal{X}_1^n$  and  $\mathbf{x}_{2j} \in \mathcal{X}_2^n, i \in \mathcal{M}_1, j \in \mathcal{M}_2$ , satisfy the average power constraints

$$\frac{1}{n} \sum_{t=1}^n (x_{ai}^{(t)})^2 \leq P_a, \quad a = 1, 2.$$

It is well-known that through a scaling transformation a Gaussian interference channel with arbitrary transmission coefficients  $a_{11}, \dots, a_{22}$  can be transformed into a channel which is equivalent to the original channel from the point of view of achievable rates and has the following standard form:

$$Y_1 = X_1 + bX_2 + Z_1, \quad (15.1)$$

$$Y_2 = aX_1 + X_2 + Z_2, \quad (15.2)$$

where  $Z_1, Z_2$  are independent zero-mean Gaussian additive noises with variance 1. Therefore we restrict ourselves to interference channels of this form.

Early work on the Gaussian interference channel includes that by Carleial [5],[6], and Sato [19]. Carleial [5] studied the Gaussian interference channel with very strong interference and Carleial [6] established an achievable rate region for the general Gaussian interference channel. These results were, to a large extent, reported in [23]. Bergmans [4] distinguished four types of interference for the Gaussian interference channel and pointed out the difficulties when the interference is medium/medium. These investigations were continued by Han and Kobayashi [14], Sato [20], Carleial [7], and Costa [9].

With some obvious modifications the results of Section IV can be carried over to the Gaussian case. The class  $\mathcal{P}^*$  of distributions is defined as before but now  $w$  is the Gaussian interference channel of the form (15.1)–(15.2) with power constraints  $P_1$  and  $P_2$ . Given the power constraints  $P_1, P_2$  define the subclass  $\mathcal{P}^*(P_1, P_2)$  of  $\mathcal{P}^*$  to consist of all  $Z = QU_1W_1U_2W_2X_1X_2Y_1Y_2 \in \mathcal{P}^*$  such that  $\sigma^2(X_1) \leq P_1, \sigma^2(X_2) \leq P_2$ . Furthermore, define the subclass  $\mathcal{P}(P_1, P_2)$  by requiring  $Z \in \mathcal{P}(P_1, P_2)$  if and only if  $Z \in \mathcal{P}^*(P_1, P_2)$  and  $Q = \phi$ . Finally, for practical purposes, the subclass  $\mathcal{P}'(P_1, P_2)$  of  $\mathcal{P}(P_1, P_2)$  is defined by  $Z \in \mathcal{P}'(P_1, P_2)$  if and only if  $Z \in \mathcal{P}(P_1, P_2), U_1, W_1, U_2, W_2$  are Gaussian, and  $X_1 = U_1 + W_1, X_2 = U_2 + W_2$ . Now, define for the Gaussian interference channel in standard form the regions

$$\mathcal{G}^* = \text{closure of } \bigcup_{Z \in \mathcal{P}^*(P_1, P_2)} \mathcal{R}(Z) \quad (16)$$

$$\mathcal{G} = \text{convex closure of } \bigcup_{Z \in \mathcal{P}(P_1, P_2)} \mathcal{R}(Z) \quad (17)$$

$$\mathcal{G}' = \text{convex closure of } \bigcup_{Z \in \mathcal{P}'(P_1, P_2)} \mathcal{R}(Z) \quad (18)$$

$$\mathcal{G}'_0 = \text{convex closure of } \bigcup_{Z \in \mathcal{P}'(P_1, P_2)} \mathcal{R}_0(Z) \quad (19)$$

$$\mathcal{G}'_C = \text{convex closure of } \bigcup_{Z \in \mathcal{P}'(P_1, P_2)} \mathcal{R}_C(Z), \quad (20)$$

where  $\mathcal{R}(Z)$ ,  $\mathcal{R}_0(Z)$ , and  $\mathcal{R}_C(Z)$  are described in Section IV.

**Theorem 5** (Han and Kobayashi [14]) :  $\mathcal{G}'_0$ ,  $\mathcal{G}'$ ,  $\mathcal{G}$ , and  $\mathcal{G}^*$  are all achievable rate regions for the Gaussian interference channel with power constraints  $P_1, P_2$ . Moreover,  $\mathcal{G}'_C \subset \mathcal{G}'_0 \subset \mathcal{G}' \subset \mathcal{G} \subset \mathcal{G}^*$ , where  $\mathcal{G}'_C$  is the region of Carleial [6].

The computation of the whole region  $\mathcal{G}^*$  directly in terms of  $a, b, P_1$  and  $P_2$  turns out to be extremely complicated and has not yet been carried out. Also, the computation of the subregion  $\mathcal{G}$  seems impractical. Han and Kobayashi [14] did compute  $\mathcal{G}'$  and  $\mathcal{G}'_0$  numerically for various values of  $a, b, P_1$  and  $P_2$  using the characterization (9) for  $\mathcal{R}(Z)$ . In four examples  $\mathcal{G}' = \mathcal{G}'_0 = \mathcal{G}'_C$ . In two other ones  $\mathcal{G}'$  strictly extends  $\mathcal{G}'_0 (= \mathcal{G}'_C)$ . In yet another example  $\mathcal{G}'_0 (= \mathcal{G}')$  extends  $\mathcal{G}'_C$ . They finally use an example due to Carleial [6] to show the important fact that  $\mathcal{G}^*$  can strictly extend on  $\mathcal{G}' (= \mathcal{G}'_C)$ , thereby motivating their conjecture that  $\mathcal{R}^* \neq \mathcal{R}$  for the general discrete memoryless interference channel. The reason for this is that, for this example, the time-division multiplex/frequency-division multiplex (TDM/FDM) curve is not contained in  $\mathcal{G}' = \mathcal{G}'_C$ , but  $\mathcal{G}^*$  always contains this curve because of the time-sharing parameter figuring in the definition of  $\mathcal{G}^*$ . The TDM/FDM curve is described by the points  $(R_1, R_2)$  such that

$$R_1 = \frac{\lambda}{2} \log\left(1 + \frac{P_1}{\lambda}\right), \quad R_2 = \frac{1-\lambda}{2} \log\left(1 + \frac{P_2}{1-\lambda}\right), \quad (21)$$

where  $0 < \lambda < 1$ .

Carleial [5] showed that the capacity region of the Gaussian interference channel with very strong interference, defined by the conditions  $a^2 \geq 1 + P_2, b^2 \geq 1 + P_1$  is the full rectangular region given by

$$0 \leq R_1 \leq C_1 \stackrel{\Delta}{=} (1/2) \log(1 + P_1), \quad (22.1)$$

$$0 \leq R_2 \leq C_2 \stackrel{\Delta}{=} (1/2) \log(1 + P_2). \quad (22.2)$$

Han and Kobayashi [14] and Sato [20] independently extended Carleial's result and found the capacity region of the Gaussian interference channel with strong interference, defined by the conditions  $a \geq 1, b \geq 1$ . They showed that this capacity region is given by the set of rates  $(R_1, R_2)$  satisfying (22.1)–(22.2) and the additional constraint

$$R_1 + R_2 \leq C_3 \stackrel{\Delta}{=} \min\left\{\frac{1}{2} \log(1 + a^2 P_1 + P_2), \frac{1}{2} \log(1 + P_1 + b^2 P_2)\right\}. \quad (23)$$

The condition of strong interference can also be expressed in terms of mutual informations by the condition (26.1)–(26.2) below.

Costa [9] continued these investigations and defined the notions of moderate and weak interference for the case  $a = b$  and  $P_1 = P_2 = P$ . Interference is said to be weak if  $0 < a = b \leq a^*$ , with  $a^* = \sqrt{d_0}$ , where  $d_0$  is the positive solution of the equation  $2d(1 + dP) = 1$ , and interference is said to be moderate if  $a^* < a = b < 1$ .

For symmetric Gaussian interference channels with moderate interference, the largest known achievable rate sum  $R_1 + R_2$  is obtained from the TDM/FDM curve, which yields  $R_1 + R_2 = (1/2) \log(1 + 2P)$ .

For symmetric Gaussian interference channels with weak interference, the largest known achievable rate sum  $R_1 + R_2$  is derived from the procedure which treats the interference signal as noise, and equals  $R_1 + R_2 = \log(1 + P/(1 + a^2 P))$ .

Whereas for the Gaussian interference channel with strong and very strong interference the capacity region is known, for the general Gaussian interference channel with interference coefficients  $0 < a < 1$ , or  $0 < b < 1$ , the capacity region remains unknown. The largest achievable rate region for the general Gaussian interference channel established so far is the one put forth by Han and Kobayashi [14].

Costa [9] established the optimality of two extreme points in the Han and Kobayashi region, viz. the points  $(C_1, C_{21})$  and  $(C_{12}, C_2)$ , where

$$C_{12} \triangleq \frac{1}{2} \log\left(1 + \frac{a^2 P_1}{1 + P_2}\right) \quad (24)$$

$$C_{21} \triangleq \frac{1}{2} \log\left(1 + \frac{b^2 P_2}{1 + P_1}\right), \quad (25)$$

and  $C_1$  and  $C_2$  are given by (22.1) and (22.2), respectively.

**Theorem 6** (Costa [9]) : For rate pairs  $(R_1, R_2)$  in the capacity region of a Gaussian interference channel with arbitrary positive interference parameters  $a$  and  $b$  and arbitrary power constraints  $P_1$  and  $P_2$  the following statements hold, for  $\epsilon > 0$  :

- (i) If  $R_2 \geq C_2 - \epsilon$ , then  $R_1 \leq C_{12} + \delta_1(\epsilon)$ , where  $\delta_1(\epsilon) \rightarrow 0$  as  $\epsilon \rightarrow 0$ .
- (ii) If  $R_1 \geq C_1 - \epsilon$ , then  $R_2 \leq C_{21} + \delta_2(\epsilon)$ , where  $\delta_2(\epsilon) \rightarrow 0$  as  $\epsilon \rightarrow 0$ .

Based on the above theorem, Costa [9] found the  $R_1$ -coordinate of the extreme point of the capacity region of the general Gaussian interference channel along the line  $R_2 = C_2$  for all values of  $a \geq 0, b \geq 0$ . Similarly, for the roles of  $R_1$  and  $R_2$  interchanged. In particular he showed that the  $R_1$ -coordinate of the extreme point of the capacity region of the general Gaussian interference channel along the line  $R_2 = C_2$  equals  $C_{12}$  when  $0 < a \leq 1$  for all values of  $b \geq 0$ . In the course of his proof, Costa [9] examined the Gaussian Z-interference channel, which is defined by the condition that  $b = 0$ . He showed that every Gaussian Z-interference channel with  $0 < a < 1$  is equivalent to a degraded Gaussian interference channel.

*Open problem 2* : Whereas for the Gaussian interference channel with strong and very strong interference the capacity region is known, for the general Gaussian interference channel with arbitrary positive interference parameters the capacity region is unknown, apart from the two extreme points found by Costa [9]. Also, the capacity region of the degraded Gaussian interference channel is as yet unknown.

## VI Miscellaneous Specific Results

As pointed out above, the capacity region of the general interference channel is unknown, both in the discrete memoryless case and in the Gaussian case. However, in several specific

situations the capacity region was obtained, by showing that a general inner bound and outer bound coincide in such a case. We will now briefly summarize these results, and mention some other progress as well.

The capacity region of a discrete memoryless interference channel with statistically equivalent outputs was obtained in [1], [6], and [17]. Sato [18] and Sato and Tanabe [21] obtained the capacity region of the class of discrete memoryless channels with very strong interference, i.e., those for which  $I(X_1; Y_1|X_2) \leq I(X_1; Y_2)$  and  $I(X_2; Y_2|X_1) \leq I(X_2; Y_1)$  for all product probability distributions on  $\mathcal{X}_1 \times \mathcal{X}_2$ . As discussed in Section III, Benzel [3] obtained the capacity region for a class of discrete additive degraded interference channels. This class may be regarded as a discrete analog of the family of degraded Gaussian interference channels considered in [6] and [19]. Sato [19] obtained several outer bounds on the capacity region of the degraded Gaussian interference channel. As mentioned in Section V, the capacity region of the Gaussian interference channel with strong interference was found in [14] and [20]. Carleial [7] found new outer bounds for the capacity regions of the discrete memoryless interference channel and the Gaussian interference channel. He showed that for discrete channels this bound coincides with the capacity region in special cases. His outer bound for the Gaussian interference channel is shown to be tighter than previously known outer bounds for an intermediate range of interference strength.

El Gamal and Costa [12] established the capacity region of a class of deterministic discrete memoryless interference channels. In this class the outputs  $Y_1$  and  $Y_2$  are deterministic functions of the inputs  $X_1$  and  $X_2$  and interferences  $V_1$  and  $V_2$  such that  $H(Y_1|X_1) = H(V_2)$  and  $H(Y_2|X_2) = H(V_1)$  for all product probability distributions on  $\mathcal{X}_1 \times \mathcal{X}_2$ , where  $V_1$  is a function of  $X_1$  and  $V_2$  is a function of  $X_2$ . The capacity region found in [12] for this situation is equal to the union of the sets  $\mathcal{R}(Z)$  described in Theorem 3 by the constraints (9), but then over those  $Z$  which reduce to product probability distributions on  $\mathcal{X}_1 \times \mathcal{X}_2$ .

Costa and El Gamal [10] investigated the discrete memoryless interference channel with strong interference, which, in analogy with the Gaussian case, is defined by the condition that

$$I(X_1; Y_1|X_2) \leq I(X_1; Y_2|X_2) \quad (26.1)$$

and

$$I(X_2; Y_2|X_1) \leq I(X_2; Y_1|X_1) \quad (26.2)$$

for all product probability distributions on  $\mathcal{X}_1 \times \mathcal{X}_2$ . Following a conjecture by Sato [20], they showed that  $\mathcal{C}$  in this case can be expressed as the union of rate pairs  $(R_1, R_2)$  satisfying

$$0 \leq R_1 \leq I(X_1; Y_1|X_2, Q) \quad (27.1)$$

$$0 \leq R_2 \leq I(X_2; Y_2|X_1, Q) \quad (27.2)$$

$$R_1 + R_2 \leq \min\{I(X_1, X_2; Y_1|Q), I(X_1, X_2; Y_2|Q)\}, \quad (27.3)$$

where  $Q$  is a time-sharing parameter of cardinality four, and the union is over all probability distributions of the form  $p(q)p(x_1|q)p(x_2|q)w(y_1, y_2|x_1, x_2)$ .

Prelov and van der Meulen [16] investigated the compound interference channel with additive almost Gaussian noise. Here, both senders send simultaneously to both receivers, as in [2], and the channel model is given by

$$Y_1 = X_1 + bX_2 + W_1 + Z_1(\epsilon_1) \quad (28.1)$$

$$Y_2 = aX_1 + X_2 + W_2 + Z_2(\epsilon_2) \quad (28.2)$$

where  $W_1$  and  $W_2$  are Gaussian noise random variables, but  $Z_1(\epsilon_1)$  and  $Z_2(\epsilon_2)$  are small additional non-Gaussian noises satisfying  $EZ_i^2(\epsilon_i) = \epsilon_i^2, i = 1, 2$ . In [16] an asymptotic expression is found for  $\mathcal{C}$  of this channel when  $\epsilon_i \rightarrow 0, i = 1, 2$ , if certain moment conditions on  $Z_1(\epsilon_1)$  and  $Z_2(\epsilon_2)$  are satisfied.

Cheng and Verdú [8] showed that multivariate Gaussian input distributions do not achieve the limiting characterization of the capacity region of the Gaussian interference channel. This limiting characterization is the expression obtained in [1] for the discrete memoryless interference channel evaluated for the Gaussian interference channel.

Vanroose and van der Meulen [24] made a classification of all binary deterministic interference channels and found uniquely decodable codes for the deterministic binary erasure interference channel with rate pairs well above the time-sharing line. This interference channel is defined by the channel functions  $Y_1 = X_1 \cdot X_2$  and  $Y_2 = X_2$ . In [24] an inner bound is derived for the zero-error capacity region of the deterministic binary erasure interference channel. In particular, it is shown that the code pair  $(C_1, C_2)$  with  $C_1 = \{00, 11\}, C_2 = \{01, 10, 11\}$  is uniquely decodable for this interference channel with rate sum 1.29248.

## VII Conclusions

Although many intricate and delicate results in the theory of achievable rate regions for the interference channel were established during the past 20 years, there are still many open problems left, in particular the determination of the capacity region of the general interference channel. These are obviously hard problems, since their solution connects with the solution of other outstanding problems in multiuser information theory, such as finding the capacity region of the general broadcast channel. Nevertheless, it would be important to continue to pursue these investigations so as not to lose perspective of where the difficulties lie for solving these problems. The number of situations where the capacity region is known is for the interference channel much smaller than for the multiple access channel. The key paper [15] by Massey and Mathys clearly belongs to the area of the multiple access channel. It would be interesting to investigate whether certain aspects of the collision channel can be carried over to the interference channel as well. Coding for interference channels also needs to be further developed, after the initial investigations of [24].

## Acknowledgements

I should like to thank the authors of the various articles mentioned in this survey for their clear expositions and the abundance of their results, of which I have made extensive use when writing. I have particularly drawn from the landmark paper by Han and Kobayashi [14]. I also should like to thank Max Costa for the thorough lecture series on the interference channel he gave in Leuven in 1986, which greatly helped me in my understanding of the interference channel, and for stimulating discussions and exchanges on this channel afterwards. I also thank Dick Blahut, Peter Vanroose, and the referees for their helpful comments.

## References

- [1] R. Ahlswede, “Multi-way communication channels”, in *Proc. 2nd Int. Symp. Information Theory*, Tsahkadsor, Armenia, U.S.S.R., Sept. 1971. Publishing House of the Hungarian Academy of Sciences, pp. 23–52, 1973.
- [2] R. Ahlswede, “The capacity region of a channel with two senders and two receivers,” *Ann. Prob.*, vol. 2, pp. 805–814, 1974.
- [3] R. Benzel, “The capacity region of a class of discrete additive degraded interference channels,” *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 228–231, 1979.
- [4] P.P. Bergmans, “The Gaussian network”, in : *The Information Theory Approach to Communications*, G. Longo, Ed. Vienna : Springer, pp. 233–261, CISM Courses and Lectures, No. 229, 1978.
- [5] A.B. Carleial, “A case where interference does not reduce capacity,” *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 569–570, 1975.
- [6] A.B. Carleial, “Interference channels,” *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 60–70, 1978.
- [7] A.B. Carleial, “Outer bounds on the capacity of interference channels,” *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 602–606, 1983.
- [8] R.S. Cheng and S. Verdú, “On limiting characterizations of memoryless multiuser capacity regions”, *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 609–612, 1993.
- [9] M.H.M. Costa, “On the Gaussian interference channel,” *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 607–615, 1985.
- [10] M.H.M. Costa and A. El Gamal, “The capacity region of the discrete memoryless interference channel with strong interference,” *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 710–711, 1987.
- [11] T.M. Cover, “An achievable rate region for the broadcast channel,” *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 399–404, 1975.
- [12] A. El Gamal and M.H.M. Costa, “The capacity region of a class of deterministic interference channels,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 343–346, 1982.
- [13] R.G. Gallager, *Information Theory and Reliable Communication*. New York : Wiley, 1968.
- [14] T.S. Han and K. Kobayashi, “A new achievable rate region for the interference channel,” *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 49–60, 1981.
- [15] J.L. Massey and P. Mathys, “The collision channel without feedback”, *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 192–204, 1985.

- [16] V.V. Prelov and E.C. van der Meulen, “The capacity region of the compound interference channel with additive almost Gaussian noise”, in *Proceedings of the Twelfth Symposium on Information Theory in the Benelux*, F.M.J. Willems and Tj. J. Tjalkens, Editors, May 23–24, 1991, Veldhoven, The Netherlands, pp. 103–106.
- [17] H. Sato, “Two-user communication channels,” *IEEE Trans. Inform. Theory*, vol. IT–23, pp. 295–304, 1977.
- [18] H. Sato, “On the capacity region of a discrete two-user channel for strong interference”, *IEEE Trans. Inform. Theory*, vol. IT–24, pp. 377–379, 1978.
- [19] H. Sato, “On degraded Gaussian two-user channels”, *IEEE Trans. Inform. Theory*, vol. IT–24, pp. 637–640, 1978.
- [20] H. Sato, “The capacity of the Gaussian interference channel under strong interference”, *IEEE Trans. Inform. Theory*, vol. IT–27, pp. 786–788, 1981.
- [21] H. Sato and M. Tanabe, “A discrete two-user channel with strong interference”, *Trans. Inst. Electron. Commun. Eng. Japan*, vol. 61, pp. 880–884, 1978.
- [22] C.E. Shannon, “Two-way communication channels”, in *Proc. 4th Berkeley Symp. on Mathematical Statistics and Probability*, Vol. 1, Berkeley, CA : Univ. California Press, 1961, pp. 611–644.
- [23] E.C. van der Meulen, “A survey of multiway channels in information theory : 1961–1976”, *IEEE Trans. Inform. Theory*, vol. IT–23, pp. 1–37, 1977.
- [24] P. Vanroose and E.C. van der Meulen : “Code constructions for the binary erasure interference channel”, in : *Proceedings Fifth Joint Soviet-Swedish International Workshop on Information Theory*, Jan. 1991, Moscow, USSR, pp. 156–159.

# Capacity of a Simple Stable Protocol for Short Message Service Over a CDMA Network

Andrew J. Viterbi  
QUALCOMM Incorporated

## Abstract

A simple protocol is described for transmission of short transaction messages on a CDMA network. The protocol employs flow control on the input based on a measurement of total signal energy received at the base stations. For bulk-Poisson arrivals of slotted message, with chi-squared distribution of received powers, the Erlang capacity is upper bounded.

## I Introduction

In a CDMA network supporting a number of uncoordinated, generally mobile, users transmitting spread spectrum modulated digital voice or stream data, capacity is maximized by power controlling all users such that their received energies at the base station are approximately equal; moreover, stability is maintained by limiting the arrival rate so as to maintain the ratio of total received power-to-background noise below a fixed level for all but a small percentage of time [1]. In a two-way system, with a base station transmitting pilot and synchronization signals, in addition to data, each user's transmitter power can be controlled accurately by a combination of open loop and closed loop methods. The open loop approach is for each user to measure the power it receives from the base station and to control its transmitted power to be inversely proportional to that received (in terms of dB-watts, the sum of the received and transmitted power is kept constant). This generally accounts for most of the variability among users, where wide distance variations and blockage profiles, by large buildings or terrain impediments, can cause the dynamic range of attenuations to be as large as 100 dB. However, because the transmit and receive center frequencies are usually separated by tens of megahertz, even after open loop power control there will usually remain a moderate discrepancy due to the differing propagation conditions in the two frequency bands. This will manifest itself as a combination of slow and fast fading, usually taken to be Rayleigh distributed. Closed loop power control, where the base station performs frequent measurements of the energy received from each user and sends up-down change commands, can accurately adjust for slowly varying fading (whose decorrelation period is on the order of one frame or packet, or more) but not for fast fading, which is, however, mitigated by interleaved error-correcting coding. The effect of the combination of open loop and closed loop control is that the received average (or slow) power variation in dB is maintained to within a standard deviation of less than 2 dB. The user capacity in Erlangs is then determined by the constraint that the total received power at the base station not exceed the background noise by more than a fixed ratio for all but a small fraction of the time.

For short message service where each user message consists of only a few tens or hundreds of bits, closed-loop power control is not possible, since the measurement time plus the transmission and processing delays exceed the message duration. We assume, therefore, that

after open loop power control, the received power level during each message will be a chi-squared distributed random variable (the square of a Rayleigh variable) with a known mean. This assumption of constant fading over the message duration represents the worst case, since a faster fading rate will permit coding and interleaving to improve performance by exploiting diversity.

The principal mechanism used to ensure stability and to enhance capacity is to control the arrival rate such that the ratio of total received power to background noise is kept below a tolerable level. Throughout we assume Poisson arrivals and a fixed message length with slotted transmission, for computational simplicity. It is expected, however, that the results will not differ significantly with unslotted transmission of variable length messages.

## II Transmission Protocol

The base station measures its total received power once per message slot. If the ratio of the measured power to the background noise exceeds a fixed threshold, the accepted message arrival rate is diminished. This is accomplished by the base station transmitting a "persistence state",  $j$ , which is a non-negative integer. Then any user desiring to transmit a message is allowed to access the network with a probability  $\pi^j$  ( $\pi < 1$ ) and otherwise refrain and depart the system. All correctly received messages are recognized and acknowledged by the base station, so that incorrect messages are retransmitted.

Let the expected number of newly accessing users per slot be  $\rho$ . Then accounting for retransmitted, erroneous, messages the average arrival rate is  $\rho/(1 - P_E)$  messages/slot. Now if the system is in persistence state  $j$ , the average rate of admitted users becomes

$$\rho_j = \frac{\rho}{1 - P_E} \pi^j, \quad j = 0, 1, 2, \dots \quad (1)$$

We take the number of admitted users in a given slot to be Poisson distributed\*, so that

$$\Pr(k \text{ users}) = \rho_j^k e^{-\rho_j} / k!, \quad k \geq 0. \quad (2)$$

The protocol consists of the base station raising the persistence state by unity whenever its received power-to-background noise power exceeds a threshold  $1/\eta$ , and lowering the persistence state by unity whenever it does not. Letting the bandwidth be  $W$  Hz, the data bit rate per user be  $R$  and the background noise density be  $N_0$ , each user's power is a chi-squared variable with mean  $\bar{E}_b R$  where  $\bar{E}_b$  is the average bit energy, the background noise power is  $N_0 W$ , and the total received power is denoted  $I_0 W$ .

Then

$$N_0 W = I_0 W - \bar{E}_b R \sum_k z_k \quad (3)$$

---

\* This is justified if the new arrivals are Poisson distributed and the admission event and error event, both binary random variables, are independent of the new arrivals.

where  $\bar{E}_b R$  is the average received power for each user and  $z_k$  is the normalized received power of the  $k$ th user.

The threshold condition is

$$I_0 / N_0 < 1 / \eta, \quad (4)$$

or equivalently using (3),

$$\sum_k z_k < \frac{(1-\eta)W/R}{\bar{E}_b / I_0} \triangleq A. \quad (5)$$

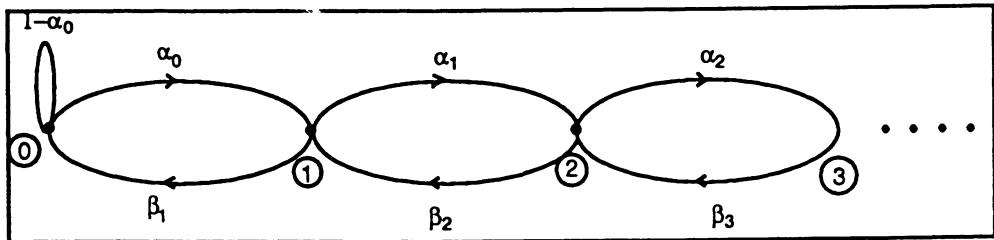
When condition (4) or (5) is violated, the persistence state is raised by unity. Thus the transition probability from state  $j$  to state  $j+1$  is given by

$$\alpha_j = \Pr\left(\sum_k z_k > A \mid \text{arrival rate } \rho_j\right), \quad j = 0, 1, 2, \dots \quad (6)$$

while the transition probability from state  $j+1$  to state  $j$  is given by

Thus the network persistence state is a "birth-death" Markov chain as described by Figure 1. Since the persistence state can not be negative, the self loop for state 0 has transition probability  $1-\alpha_0$ .

$$\beta_{j+1} = \Pr\left(\sum_k z_k < A \mid \text{arrival rate } \rho_{j+1}\right) = 1 - \alpha_{j+1}, \quad j = 0, 1, 2, \dots \quad (7)$$



**Figure 1: Markov Persistence-State Diagram**  
 (Persistence State  $j$  has State Probability  $P_j$  and admits Arrival Rate  $\rho_j$ )

### III Throughput and Blocking Probability

It is shown in Appendix I that the transition probability,  $\alpha_j$ , from any state is upper bounded by

$$\alpha_j < \begin{cases} \exp\left[-A\left(1 - \sqrt{\rho_j / A}\right)^2\right], & \rho_j < A \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

and we may obtain a pessimistic\* estimate of the persistence steady-state probabilities  $P_j$  by solving the linear equilibrium equations obtained from Figure 1, which lead to the recurrence relations

$$P_{k+1} = (P_k - \alpha_{k-1} P_{k-1}) / (1 - \alpha_{k+1}), \quad k = 1, 2, \dots$$

$$\text{where } P_1 = \alpha_0 P_0 / (1 - \alpha_1)$$

$$\text{and } \sum_{k=0}^{\infty} P_k = 1. \quad (9)$$

The solution can be written as the ratio of products

$$P_k = \frac{\prod_{j=0}^{k-1} \alpha_j}{\prod_{j=1}^k (1 - \alpha_j)} P_0$$

It follows that the steady state probabilities exist and hence the system is stable about some persistence state  $j > j_0$  for which

$$\rho_{j_0} = A$$

or, using the definitions (1) and (5),

$$\frac{\rho \pi^{j_0}}{1 - P_E} = A = \frac{(1 - \eta)(W / R)}{\bar{E}_b / I_0}. \quad (10)$$

The average system throughput of correctly received messages is then

$$\bar{\rho} = \sum_{j=j_0}^{\infty} \rho_j P_j (1 - P_E) = \rho \sum_{j=j_0}^{\infty} \pi^j P_j = B \sum_{j=j_0}^{\infty} \pi^{(j-j_0)} P_j \quad (11)$$

\* Pessimistic means that all state probabilities are upper bounded except  $P_0$ . This follows from the fact that the transition probabilities in the direction of increasing persistence states are upper bounded; hence the average throughput (eq. 11) will be a lower bound.

where

$$B = (1 - P_E) A = \frac{(1 - P_E)(1 - \pi)W / R}{\bar{E}_b / I_0} . \quad (12)$$

At the same time the blocking probability, which is the probability that a newly arriving user will be denied service, is

$$P_{\text{block}} = 1 - \sum_{j=j_0}^{\infty} \pi^j P_j = 1 - (\bar{\rho} / \rho) \quad (13)$$

Table I shows the transition probabilities  $\alpha_j$  and steady-state probabilities  $P_j$ , as obtained from (8) and (9) for normalized arrival rate,  $\rho / B = .8$ , and  $\pi = .95$ ; also shown are the average normalized throughput  $\bar{\rho} / B$  and blockage probability  $1 - \bar{\rho} / \rho$ , as obtained from (11) and (13).

| j                       | $\alpha_j$           | $P_j$                          |
|-------------------------|----------------------|--------------------------------|
| 0                       | .328                 | .665                           |
| 1                       | .193                 | .270                           |
| 2                       | .104                 | .058                           |
| 3                       | .052                 | $6.4 \times 10^{-3}$           |
| 4                       | .024                 | $3.4 \times 10^{-4}$           |
| 5                       | .011                 | $8.5 \times 10^{-6}$           |
| 6                       | $4.4 \times 10^{-3}$ | $9.0 \times 10^{-8}$           |
| $\bar{\rho} / B = .784$ |                      | $1 - \bar{\rho} / \rho = .020$ |

Table I: Transition and Steady-State Probabilities  
for Normalized Arrival Rate  
 $\rho / B = .8$ ,  $\pi = 0.95$ ,  $A = 100$

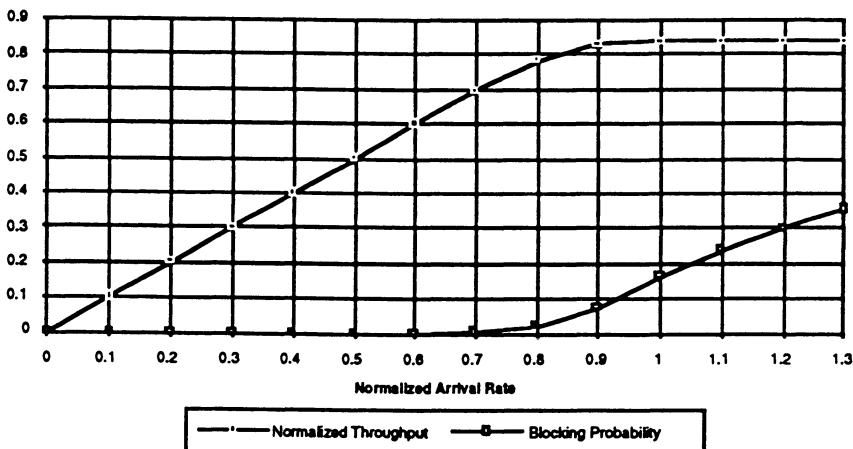


Figure 2: Throughput and Blocking Probability as Function of New Arrival Rate

Figure 2 shows the average normalized throughput as well as the blockage probability  $1 - \bar{P} / \rho$  as a function of the normalized new arrival rate  $\rho / B$ . It is clear that the blocking probability becomes unacceptable for  $\rho / B > 0.8$ .

Examining the parameter  $B$  as defined by (12), it is apparent that since  $1 - P_E$  must be a monotonically increasing function of  $\bar{E}_b / I_0$ , the factor  $(1 - P_E) / (\bar{E}_b / I_0)$  has a unique maximum. Suppose that transmission is protected by a very long code, such that

$$P_E(E_b / I_0) = \begin{cases} 1, & E_b / I_0 \leq \alpha \\ 0, & E_b / I_0 > \alpha \end{cases}.$$

Then since  $E_b / I_0$  is a chi-squared distributed variable, we have that the error probability averaged over this variable is

$$1 - P_E = \int_{\alpha}^{\infty} \frac{e^{-x/(\bar{E}_b / I_0)}}{\bar{E}_b / I_0} dx = e^{-\alpha/(\bar{E}_b / I_0)}. \quad (14)$$

In this case,

$$B = \frac{(1 - P_E)(1 - \eta) W / R}{\bar{E}_b / I_0} = \frac{(1 - \eta)W / R}{(\bar{E}_b / I_0) \exp[\alpha / (\bar{E}_b / I_0)]} \quad (15)$$

which is maximized at  $\bar{E}_b / I_0 = \alpha$ , to yield

$$B_{Max} = \frac{(1 - \eta)(W / R) e^{-1}}{\alpha}. \quad (16)$$

At the Shannon capacity for an arbitrarily wideband transmission,  $\alpha = \ln 2$ , while at the cutoff rate,  $\alpha = 2 \ln 2$ . In the case that the exact relationship between  $P_E$  and  $\bar{E}_b / I_0$  is unknown, the maximum may be found empirically by varying  $\bar{E}_b / I_0$  and measuring the resulting error probability.

## IV Conclusions

We have demonstrated a simple multiple access protocol which guarantees stability, as is evident from the fact that the throughput increases monotonically with newly offered traffic rate. When the Rayleigh fading is slower than the message duration, a worst case assumption, an upper bound on the throughput is given by  $B$  of eq. (12) which, in the case of idealized coded transmission, is upper bounded by (16).

## Appendix I

### Moment Generating Functions and Chernoff Bounds for Total Received Energy

Let  $z_k$  be the normalized received power from the  $k$ th user, with mean power unity. We determine the moment generating function of the sum  $\sum_{k=1}^K z_k$  where  $K$  is a Poisson random variable representing the number of active users,

$$M(s) = E_K \left[ E_z (e^{sz}) \right]^K. \quad (\text{A.1})$$

Since  $z_k$  is the square of a normalized Rayleigh variable,  $p(z)=e^{-z}$  and

$$E_z (e^{sz}) = \int_0^\infty p(z) e^{sz} dz = 1 / (1-s), \quad s < 1 \quad (\text{A.2})$$

and since  $K$  is Poisson with arrival rate  $\rho_j$  (when the system is in the  $j$ th persistence state)

$$M(s) = \sum_{K=0}^{\infty} \frac{\rho_j^K e^{-\rho_j}}{K!} \frac{1}{(1-s)^K} = \exp[\rho_j s / (1-s)]. \quad (\text{A.3})$$

The Chernoff bound on the probability that the sum exceeds  $A$  is

$$\begin{aligned} \Pr \left( \sum_{k=1}^K z_k > A \right) &< \min_{s > 0} [M(s) e^{-sA}] = \min_{s > 0} \exp \left\{ -s \left[ A - \rho_j / (1-s) \right] \right\} \\ &= \exp \left[ -A \left( 1 - \sqrt{\rho_j / A} \right)^2 \right], \quad \rho_j < A. \end{aligned} \quad (\text{A.4})$$

### Reference

- [1] A.M. Viterbi, A.J. Viterbi, "Erlang Capacity of Power Controlled CDMA System," *IEEE Journal on Selected Areas in Communication*, Vol. 11, No. 6, pp. 892-900, August 1993.

# The Sliding-Window Lempel-Ziv Algorithm is Asymptotically Optimal

A. D. Wyner

AT&T Bell Laboratories  
600 Mountain Avenue  
Murray Hill, New Jersey

J. Ziv\*

Faculty of Electrical Engineering  
Technion-Israel Institute of Technology,  
Haifa, Israel

## Abstract

The sliding-window version of the Lempel-Ziv data-compression algorithm (sometimes called LZ '77) has been thrust into prominence recently. This is the algorithm used in the highly successful "Stacker" program for personal computers. It is also incorporated into Microsoft's new MS-DOS-6. Although other versions of the Lempel-Ziv algorithm are known to be optimal in the sense that they compress a data-source to its entropy, optimality in this sense has never been demonstrated for this version.

In this paper, we will describe the algorithm, and show that as the "window-size", a quantity which is related to the memory and complexity of the procedure, goes to infinity, the compression rate approaches the source entropy. The proof is surprisingly general, applying to all finite alphabet stationary ergodic sources.

## I Introduction

The sliding-window version of the Lempel-Ziv (LZ) data compression algorithm (first proposed in [6] in 1977 and sometimes called LZ '77) has been thrust into prominence recently. A version of this algorithm is used in the highly successful "Stacker" program for personal computers [3]. It is also incorporated into Microsoft's new MS-DOS-6. Although other versions of the LZ algorithm are known to be optimal in the sense that they compress a data source to its entropy, such optimality has never been demonstrated for the sliding-window version.

This paper begins with a description of the sliding-window LZ algorithm. The main result is a theorem which asserts that as the "window-size" (a quantity directly related to the memory and complexity requirements of the procedure) becomes large, the compression rate approaches the source entropy. This theorem is surprisingly general, applying to all stationary, ergodic, finite-alphabet sources.

We will be concerned with a data source which is a random sequence  $\{X_k\}_{k=-\infty}^{\infty}$ , that is stationary, ergodic, and takes values in the alphabet  $\mathcal{A}$  with cardinality  $|\mathcal{A}| = A < \infty$ .

---

\*Work on this paper was done while visiting AT&T Bell Laboratories

For  $-\infty \leq i < j \leq \infty$ , let  $\mathbf{X}_i^j$  denote the substring  $(X_i, X_{i+1}, \dots, X_j)$ . Let

$$H_n = \frac{1}{n} H(X_1^n) \triangleq -\frac{1}{n} \sum_{\mathbf{x} \in \mathcal{A}^n} \Pr\{\mathbf{X}_1^n = \mathbf{x}\} \log(\Pr\{\mathbf{X}_1^n = \mathbf{x}\}) ,$$

be the  $n$ th order (normalized) entropy of  $\{X_k\}$ , and let  $H = \lim_{n \rightarrow \infty} H_n$  be the source entropy. (All logarithms in this paper are taken to the base 2.) It is well known that this limit always exists, and that the data source can be losslessly encoded using  $H + \epsilon$  bits per source symbol [1] [2] (for arbitrarily  $\epsilon > 0$ ). The LZ algorithm is a universal procedure (which does not depend on the source statistics) for encoding the source.

In Section II we give a precise description of the sliding-window LZ algorithm and state the main result. In Section III we establish the mathematical facts that we need in the proof of this theorem which we give in Section IV.

Let us remark at this point that the sliding-window LZ algorithm discussed here is not the same as the “LZ ‘78” algorithm, used for example in the UNIX “compress” command and in a CCITT standard for data-compression for modems. In LZ ‘78, the “dictionary” is allowed to grow until it reaches a certain specified size. In the sliding-window version the dictionary is of fixed size, and consists of the data symbols immediately preceding the data being compressed.

Finally we remark that the treatment in [6], where the sliding-window LZ algorithm is first described, is nonprobabilistic. In that reference, the algorithm is shown to be optimal for a certain (deterministic) family of sources. Furthermore, for a given source in this family, the rate of convergence is similar to that of the best nonuniversal compression scheme. In the present paper, we replace the assumption that the source belongs to this family, by the more conventional (and realistic) assumption that the source is stationary and ergodic.

## II Description of the Algorithm

Let  $\{X_k\}_{k=-\infty}^\infty$ , be a stationary ergodic random process with entropy  $H$ , as discussed in Section I. We will now describe the sliding-window Lempel-Ziv algorithm for encoding the sequence  $\{X_k\}_{k=1}^N$ , where  $N$  is a large integer. Later we shall let  $N \rightarrow \infty$ .

Let  $n_w > 0$  be an integer parameter, called the *window size*. Assume that  $n_w$  is a power of two. The first  $n_w$  symbols of  $\mathbf{X}_1^N$  are encoded with no attempt at compression. We call  $\mathbf{X}_1^{n_w}$  the *window*. The number of bits required to encode the window  $\mathbf{X}_1^{n_w}$  is  $\lceil n_w \log A \rceil$ . These bits are to be considered as overhead that will be amortized over a very long time. We next define the first *phrase*  $\mathbf{Y}_1 \triangleq \mathbf{X}_{n_w+1}^{n_w+L_1}$ , where  $L_1$  is the largest integer such that

$$\mathbf{X}_{n_w+1}^{n_w+L_1} = \mathbf{X}_{n_w-m}^{n_w-m+L_1-1} \quad \text{for some } m \in [0, n_w - 1] . \quad (1)$$

Thus  $L_1$  is the largest integer such that a copy of  $\mathbf{X}_{n_w+1}^{n_w+L_1}$  begins in the window. Let  $m_1$  be the (say) smallest of those  $m$ 's which satisfy (1). If (1) is not satisfied for any  $m \in [0, n_w - 1]$  that is  $X_{n_w+1} \neq X_m$ , for all  $m \in [1, n_w]$ , then we take  $L_1 = 1$ . For example if  $n_w = 5$ , and  $(X_1, X_2, \dots) = (a b c d e : d e d a \dots)$ , then  $L_1 = 3$  since the string  $(d e d)$  begins at position 4 ( $\leq n_w$ ), i.e.  $\mathbf{X}_{n_w+1}^{n_w+3} = \mathbf{X}_{n_w-1}^{n_w-1+2}$ . Also  $m_1 = 1$ . Note that there is

no upper bound on  $L_1$  and in particular it can exceed  $n_w$  — for example if  $n_w = 5$  and  $(X_1, X_2, \dots) = (b b b b a : a a a a a a a a a b \dots)$ , then  $L_1 = 10$ .

We now show how to encode the data sequence  $\mathbf{Y}_1$ . The first part of the encoding is a comma-free binary encoding of  $L_1$ . One such encoding is the mapping  $e(L)$  given in the Appendix. Immediately following  $e(L_1)$  is a binary string  $s_1$ , where

$$s_1 = \begin{cases} \text{binary encoding of } m_1, & \text{if } \log n_w < \lceil L_1 \log A \rceil, \\ \text{binary encoding (no compression) of } \mathbf{Y}_1, & \text{if } \log n_w \geq \lceil L_1 \log A \rceil. \end{cases} \quad (2)$$

Assume that  $n_w \geq A$ , so that when  $L_1 = 1$ ,  $s_1$  is an encoding of  $\mathbf{Y}_1$ . Thus the total number of bits needed to encode the first phrase  $\mathbf{Y}_1 = \mathbf{X}_{n_w+1}^{n_w+L_1}$  is  $|e(L_1)| + |s_1|$ . Since  $m_1 \in [0, n_w - 1]$ , its binary encoding requires  $\log n_w$  bits. Also  $\lceil L_1 \log A \rceil$  bits are required to encode  $\mathbf{Y}_1$  with no compression. Thus the length of the encoding of  $\mathbf{Y}_1$  is

$$\begin{aligned} & |e(L_1)| + \min(\log n_w, \lceil L_1 \log A \rceil) \\ & \leq \min(\log n_w + \gamma_1 \log(L_1 + 1), \gamma_2 L_1), \end{aligned} \quad (3)$$

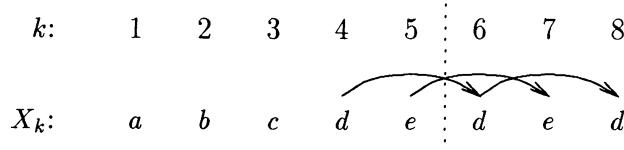
for suitably large  $\gamma_1, \gamma_2$ . We made use of (A.4b) in (3).

Now observe that with knowledge of the window  $\mathbf{X}_1^{n_w}$ ,  $e(L_1)$ , and  $s_1$ , the decoder can recover  $\mathbf{Y}_1 = \mathbf{X}_{n_w+1}^{n_w+L_1}$ . Here is how. The first step is to decode  $e(L_1)$  to recover  $L_1$ . The decoder now computes  $\lceil L_1 \log A \rceil$ . If this quantity is  $\leq \log n_w$ , the next  $\lceil L_1 \log A \rceil$  bits are a binary encoding of  $\mathbf{Y}_1$ . Otherwise the next  $\log n_w$  bits define a pointer to position  $(n_w - m_1)$  where a copy of  $\mathbf{Y}_1$  begins. This position is within the window, i.e.  $n_w - m_1 \in [1, n_w]$ . The decoder copies  $X_{n_w-m_1}$  into position  $n_w + 1$ , then  $X_{n_w-m_1+1}$  into position  $n_w + 2$ , etc. When  $X_{n_w-m_1+L_1}$  is copied into position  $n_w + L_1$ , the process is complete and  $\mathbf{Y}_1$  is reconstructed.

As an example, consider again the case  $n_w = 5$ , where

$$(X_1, X_2, \dots) = (a b c d e : d e d a \dots).$$

As above  $L_1 = 3$  and  $m_1 = 1$ . With knowledge of the window,  $\mathbf{X}_1^{n_w} = (a b c d e)$ , the decoder can copy ‘d’ and ‘e’ to positions 6 and 7, respectively, and then copy the ‘d’ in position 6 to position 8:



Thus  $\mathbf{Y}_1 = (d e d)$ .

After the first phrase  $\mathbf{Y}_1$  is encoded, the first  $L_1$  symbols in the window are deleted and the  $L_1$  symbols of  $\mathbf{Y}_1$  are added at the end of the window. Thus the new window is  $\mathbf{X}_{L_1+1}^{L_1+n_w}$ , and has length  $n_w$ . Phrase 2 is now constructed using the new window and data  $X_{n_w+L_1+1}, X_{n_w+L_1+2}, \dots$ . The process is repeated until the  $N$  symbols are exhausted.

The phrases are  $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_c$  and their lengths are  $L_1, L_2, \dots, L_c$ , respectively. The last phrase  $\mathbf{Y}_c$  is terminated prematurely in the obvious way when the data is exhausted.

From (3), the total number of bits to encode  $\mathbf{X}_1^N$  is

$$\begin{aligned}\nu(\mathbf{X}_1^N) &= \sum_{i=1}^c \{|e(L_i)| + \min(\log n_w, \lceil L_i \log A \rceil)\} \\ &\quad + \lceil n_w \log A \rceil,\end{aligned}\tag{4}$$

where the last term is overhead for the first window. Note that  $c$  and the  $\{L_i\}_{i=1}^c$  are random variables. The average rate is  $\bar{R}(N) \triangleq E[\nu(X_1^N)/N]$ . Using the bound of (3) we have

$$\boxed{\bar{R}(N) \leq \frac{\lceil n_w \log A \rceil}{N} + \frac{1}{N} E \sum_{i=1}^c \min(\log n_w + \gamma_1 \log(L_i + 1), \gamma_2 L_i)}\tag{5}$$

We are now ready to state our main result.

**Theorem 1**  $\lim_{n_w \rightarrow \infty} \lim_{N \rightarrow \infty} \bar{R}(N) = H$ .

It is interesting to note that the version of the sliding-window LZ algorithm used by Stac in their Stacker program [3] does not encode the length  $L_i$  of the  $i$ th phrase using  $O(\log L_i)$  bits. In fact, they encode  $L_i$  using  $4 + 4\lceil \frac{L_i - 7}{15} \rceil \sim \frac{4L_i}{15}$ , as  $L_i \rightarrow \infty$ . With this encoding, Theorem 1 will not hold. We should point out that their commercial algorithm was designed and optimized for a particular window size  $n_w$ .

The proof of Theorem 1 is given in the remainder of this paper. Some of the ideas were used in a previous paper by the authors [5] on the asymptotic optimality of a fixed data base version of LZ.

### III Mathematical Preliminaries

In this section we will state a result of Wyner and Ziv [4] which appeared in 1989. For an infinite sequence  $\mathbf{x} = \mathbf{x}_{-\infty}^\infty$ , where  $x_n \in \mathcal{A}$ , define for  $\ell > 0$

$$\begin{aligned}W_\ell(\mathbf{x}) &= \text{smallest } k > 0 \text{ such that} \\ \mathbf{x}_0^{\ell-1} &= \mathbf{x}_{-k+\ell-1}^{-k}.\end{aligned}\tag{6}$$

Thus  $W_\ell(\mathbf{x})$  is the number of positions that we have to “slide”  $\mathbf{x}_0^{\ell-1}$  to the left to get a perfect match. For example, with

$$\begin{array}{cccccc|cccccc} n : & -5 & -4 & -3 & -2 & -1 & | & 0 & 1 & 2 & 3 & 4 \\ \mathbf{x} : & \dots & a & b & c & a & b & | & c & a & b & c & d & \dots \end{array}$$

$W_4(\mathbf{x}) = 3$ , since  $\mathbf{x}_{-3}^0 = \mathbf{x}_0^3$ . Here is the theorem we need.

**Theorem 2** For  $\ell = 1, 2, \dots$ , and  $\epsilon > 0$  arbitrary,

$$\Pr\{W_\ell(\mathbf{X}) > 2^{\ell(H+\epsilon)}\} \rightarrow 0, \text{ as } \ell \rightarrow \infty,$$

where  $H$  is the entropy of  $\{X_n\}$ .

We will use the following form of Theorem 2.

**Corollary 3** *Let  $n(\ell)$ ,  $\ell = 1, 2, \dots$ , satisfy*

$$L \triangleq \lim_{\ell \rightarrow \infty} \frac{1}{\ell} \log n(\ell) > H . \quad (3.2a)$$

*Then*

$$\lim_{\ell \rightarrow \infty} \Pr\{W(\ell) > n(\ell)\} = 0 . \quad (3.2b)$$

**Proof:** Let  $\epsilon = L - H > 0$ . For  $\ell \geq \ell_0$  (sufficiently large),  $\frac{1}{\ell} \log n(\ell) \geq L - \frac{\epsilon}{2} = H + \frac{\epsilon}{2}$ . Thus

$$n(\ell) \geq 2^{\ell(H+\frac{\epsilon}{2})} .$$

Thus, for  $\ell \geq \ell_0$

$$\Pr\{W(\ell) > n(\ell)\} \leq \Pr\{W(\ell) > 2^{\ell(H+\frac{\epsilon}{2})}\} \rightarrow 0 ,$$

by Theorem 2.

## IV Proof of Theorem 1

That  $\overline{R}(N) \geq H$  for all  $N$  follows from the converse to the coding theorem for lossless source coding. Since the algorithm defined in Section II yields a prefix-free (“instantaneous”) code for  $X_1, \dots, X_N$ ,

$$E\nu(N) \geq H(X_1, \dots, X_N) .$$

(See Theorem 5.3.1 in [1].) Since  $H(X_1, \dots, X_N) \geq NH$ ,  $R(N) = E\nu(N)/N \geq H$ .

We now show that  $\lim \overline{R}(N) \leq H$ . Suppose that we have implemented the algorithm to encode  $X_1, \dots, X_N$ . Subdivide the interval  $[n_w + 1, N]$  into  $N'/\ell_0$  subintervals of length  $\ell_0$ , where

$$N' = N - n_w , \quad (4.1a)$$

and

$$\ell_0 = \frac{\log n_w}{H + \epsilon} , \quad (4.1b)$$

and  $\epsilon > 0$  is arbitrary. (Assume that  $\epsilon$  is adjusted so that  $\ell_0$  is an integer, and that  $\ell_0$  divides  $N'$ ). Recall that  $\mathbf{X}_1^{n_w}$  is the (uncoded) initial window. Denote the  $N'/\ell_0$  subintervals by  $\{I_j\}$ , where  $I_j = [n_w + (j-1)\ell_0 + 1, n_w + j\ell_0]$ ,  $j = 1, 2, \dots, N'/\ell_0$ . We say that a subinterval  $I_j$  is *bad* if a copy of  $\{X_k\}_{k \in I_j}$  does not begin in the string of  $(n_w - \ell_0)$  symbols preceding it, i.e.

$$\mathbf{X}_{n_w + (j-1)\ell_0 + 1}^{n_w + j\ell_0} \neq \mathbf{X}_{n_w + (j-1)\ell_0 + 1-m}^{n_w + j\ell_0 - m} , \quad 1 \leq m \leq n_w - \ell_0 . \quad (2)$$

Now referring to the discussion in Section III, we have that

$$\Pr\{I_j \text{ is bad}\} = \Pr\{W(\ell_0) > n_w - \ell_0\} . \quad (3)$$

Since  $n(\ell_0) \triangleq n_w - \ell_0 = 2^{\ell_0(H+\epsilon)} - \ell_0$  satisfies (3.2a), with  $L = H + \epsilon$ , Corollary 3 implies that

$$\Pr\{I_j \text{ is bad}\} \rightarrow 0 , \quad (4)$$

as  $n_w$  (and  $\ell_0 \rightarrow \infty$ ).

Now let  $\{Y_i\}$ ,  $1 \leq i \leq c$  be the phrases generated by the algorithm when encoding  $X_1^N$ . For  $1 \leq i \leq c$ , let  $Y'_i$  be the phrase  $Y_i$  augmented by the next symbol from  $X_1^N$ . Thus if  $Y_i = X_k^{k+L_i-1}$ , then  $Y'_i = X_k^{k+L_i}$ . We say that the phrase  $Y_i$  is an *internal* phrase, if the corresponding augmented phrase  $Y'_i$  begins and ends in the same subinterval  $I_j$ . We claim that if  $Y_i$  is an internal phrase supported on  $I_j$ , then  $I_j$  is a bad subinterval. We show this with the aid of the schematic diagram in Figure 1. The figure shows the augmented internal phrase  $Y'_i$  supported on the interval  $I_j$ . The window corresponding to phrase  $Y_i$

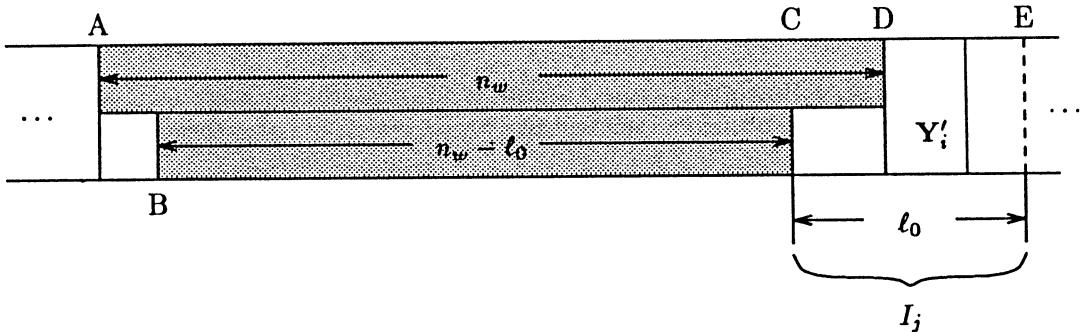


Figure 1

is supported on the interval AD. By definition, a copy of  $Y'_i$  does *not* begin in this window (since  $Y_i$  is the *longest* sequence, beginning at D, a copy of which begins in window). Since  $Y'_i$  is supported on  $I_j$ , a copy of  $\{X_k\}_{k \in I_j}$  cannot begin in the window AC, and therefore cannot begin in BC (since B is to the right of A). Thus  $I_j$  is a bad subinterval.

Let  $S_I = \{i : Y_i \text{ is internal}\}$ , and let  $\psi$  be the fraction of the intervals  $\{I_j\}$  which are bad. It follows that

$$\begin{aligned} \sum_{i \in S_I} |Y_i| &\leq \ell_0 \{\text{no. of bad intervals}\} \\ &\leq \ell_0 \frac{N'}{\ell_0} \psi = N' \psi. \end{aligned} \quad (5)$$

We are now ready to complete the proof of Theorem 1. Rewrite Inequality 5 as

$$\bar{R}(N) \leq \frac{[n_w \log A]}{N} + \frac{1}{N} E \left\{ \sum_{i \in S_I} \gamma_2 L_i + \sum_{i \in S_I^c} [\log n_w + \gamma_1 \log(L_i + 1)] \right\}. \quad (6)$$

Now the first expectation is from (5)

$$\frac{1}{N} E \sum_{i \in S_I} \gamma_2 L_i \leq \frac{\gamma_2 N'}{N} E \psi \leq \gamma_2 \Pr\{I_j \text{ is bad}\}. \quad (7)$$

Finally observe that if  $Y_i$  is *not* an internal phrase, then  $Y_i$  must occupy the last position of some subinterval  $I_j$ . Thus

$$c' \triangleq |S_I^c| \leq \{\text{number of subintervals}\} = \frac{N'}{\ell_0}. \quad (8)$$

The second term in (6) (without the expectation) is

$$\begin{aligned}
& \frac{1}{N} \sum_{i \in S_I^c} \log n_w + \gamma_1 \frac{1}{N} \sum_{i \in S_I^c} \log(L_i + 1) \\
&= |S_I^c| \frac{\log n_w}{N} + \gamma_1 \frac{c'}{N} \sum_{i \in S_I^c} \frac{1}{c'} \log(L_i + 1) \\
&\stackrel{(a)}{\leq} |S_I^c| \frac{\log n_w}{N} + \gamma_1 \frac{c'}{N} \log \left( \frac{1}{c'} \sum L_i + 1 \right) \\
&\stackrel{(b)}{\leq} \frac{N' \log n_w}{N \ell_0} + \gamma_1 \frac{c'}{N} \log \left( \frac{N}{c'} + 1 \right) \\
&\stackrel{(c)}{\leq} \frac{\log n_w}{\ell_0} + \frac{\gamma_1}{\ell_0} \log(\ell_0 + 1) \\
&\stackrel{(d)}{=} (H + \epsilon) + (\gamma_1/\ell_0) \log(\ell_0 + 1). \tag{9}
\end{aligned}$$

Step (a) in (9) follows from the concavity of  $\log(\cdot)$ ; step (b) from  $\sum_{i \in S_I^c} L_i \leq N$ ; step (c) from the fact that  $\frac{1}{x} \log(x + 1)$ ,  $x \geq 0$ , is decreasing in  $x$ , and from (8) which implies  $\frac{c'}{N} \leq \frac{N'}{N \ell_0} \leq \frac{1}{\ell_0}$ ; and step (d) from (4.1b).

Substituting (7) and (9) into (6) we have

$$\begin{aligned}
\bar{R}(N) &\leq \frac{\lceil n_w \log A \rceil}{N} + \gamma_2 \Pr\{I_j \text{ bad}\} + H + \epsilon \\
&\quad + (\gamma_1/\ell_0) \log(\ell_0 + 1),
\end{aligned}$$

and

$$\begin{aligned}
\lim_{N \rightarrow \infty} \bar{R}(N) &\leq \gamma_2 \Pr\{I_j \text{ is bad}\} + H + \epsilon \\
&\quad + (\gamma_1/\ell_0) \log(\ell_0 + 1).
\end{aligned}$$

Finally, letting  $n_w \rightarrow \infty$ , and using (4) we have

$$\lim_{n_w \rightarrow \infty} \lim_{N \rightarrow \infty} \bar{R}(N) \leq H + \epsilon,$$

which is Theorem 1, on letting  $\epsilon \rightarrow 0$ .

## Appendix: Comma-Free Coding of an Unbounded Integer

In this appendix we define a scheme for unambiguously encoding an integer  $L$  into a binary string. Let  $\{0, 1\}^*$  be the set of binary sequences of arbitrary length. We will give an encoding or mapping  $e : [1, \infty) \rightarrow \{0, 1\}^*$ , which maps the nonnegative integers into binary sequences, such that for any distinct  $L_1, L_2$ ,  $e(L_1)$  is not a prefix of  $e(L_2)$ . Thus the code is uniquely decipherable, see for example [1], Chapter 5.

For  $k \in [1, \infty)$  let  $b(k)$  be the binary expansion of the integer  $k$ . Thus  $b(1) = 1$ ,  $b(2) = 10$ , etc. The length of this expansion is

$$|b(k)| = \lceil \log(k + 1) \rceil. \tag{A.10}$$

Also, for  $k = 1, 2, \dots$ , let  $u(k) \triangleq 0^{k-1}1$ , the concatenation of  $(k-1)$  0's followed by a 1. First observe that

$$\hat{e}(k) = u(|b(k)|) * b(k), \quad (\text{A.11})$$

where  $*$  denotes concatenation, is a prefix-free encoding of  $k \in [1, \infty)$ . For example if  $\hat{e}(k_1) = 00011001$ , the prefix 0001 tells us that the next four bits, 1001, are the binary encoding of  $k_1 = 9$ .

The mapping that we will use is, for  $L = 1, 2, \dots$ ,

$$e(L) = \hat{e}(|b(L)|) * b(L). \quad (\text{A.12})$$

The prefix  $\hat{e}(|b(L)|)$  encodes the length of  $b(L)$ , and the next  $|b(L)|$  bits is the binary expansion of  $L$ . For example, when  $L = 7$ ,  $b(L) = 111$ , and  $|b(L)| = 3$  so that  $\hat{e}(|b(L)|) = 0111$ . Thus  $e(L) = e(7) = 0111111$ .

We need a bound in  $|e(L)|$ . From (A.11)  $|\hat{e}(k)| = 2|b(k)|$ . Thus from (A.12),

$$|e(L)| = 2|b(|b(L)|)| + |b(L)|.$$

Using (A.10) we see that for large  $L$ ,  $|e(L)| \sim \log L + 2 \log \log L$ . We can account for small values of  $L$  by writing, for all  $L = 1, 2, \dots$ ,

$$|e(L)| \leq \log(L+1) + \gamma \log \log(L+2), \quad (\text{A.4a})$$

for a suitably large  $\gamma$ . For our purposes the very weak bound

$$|e(L)| \leq \gamma \log(L+1) \quad (\text{A.4b})$$

will suffice, where  $\gamma$  is appropriately large, say  $\gamma = 10$ .

## References

- [1] Cover, T. and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.
- [2] Gallager, R., *Information Theory and Reliable Communication*, Wiley, New York, 1968.
- [3] Whiting, D., G. George and G. Ivey, "Data Compression Apparatus and Method", US Patent No. 5016009, 1991 (Assigned to Stac, Inc.).
- [4] Wyner, A. D. and J. Ziv, "Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression", *IEEE Transactions on Information Theory*, Vol. IT-36, pp. 1250-1258, 1989.
- [5] Wyner, A. D. and J. Ziv, "Fixed data base version of the Lempel-Ziv data compression algorithm", *IEEE Transactions on Information Theory*, Vol. IT-37, pp. 878-880, 1991.
- [6] Ziv, J. and A. Lempel, "A universal algorithm for sequential data compression", *IEEE Transactions on Information Theory*, Vol. IT-24, pp. 337-343, 1977.

# On Code Linearity and Rotational Invariance for a Class of Trellis Codes for M-PSK

Lars H. Zetterberg  
Royal Institute of Technology  
S-100 44 Stockholm

## Abstract

A class of trellis codes is defined based on a novel trellis structure. An algebraic basis is introduced using either the ring of integers  $\mathbf{Z}_M$  or the ring of polynomials  $\mathbf{P}_M$  defined over the primary field  $GF(p)$ . Codes are constructed for  $M$ -PSK by mapping subgroups and their cosets onto branch planes in the trellis. Special attention is given to codes that are linear in the defined algebra and to those for which decoding is unaffected by certain phase errors in the receiver.

## I Introduction

In a report [1] and the conference proceedings [2], the concept of a geometrically designed trellis was introduced. Code construction for digital phase modulation was developed by representation of phase values by integer coefficients. A scheme was set up to put coefficients as labels on trellis branches using groups defined in the ring of integers  $\mathbf{Z}_M$ . A detailed study was made of several code classes, with search for good codes, for  $M$ -PSK with  $M = 4, 6, 8$  and  $9$ .

This paper will deal with the same subject from a general theoretic point of view. Code construction will be described in a formal way that covers several subclasses of codes. The algebraic basis is extended to include both the ring of integers  $\mathbf{Z}_M$  and the polynomial ring  $\mathbf{P}_M$  with  $M = p^m$  and  $p$  a prime number.

Fundamental concepts are introduced such as procedures for coding and decoding, and code design equivalence. Analysis is focused on finding conditions for codes to be linear and making decoding insensitive to phase errors in the receiver. Design examples for  $M$  equal to 6 and 8 will illustrate the design procedure. The relation between our codes and coset codes investigated by Forney [3] will be observed.

## II Basic Concepts

### Trellis Design

A trellis may be described as a three-dimensional structure, segmented by nodes located in transversal node planes. Nodes are arranged in rectangular arrays with  $q$  horizontal and

$r$  vertical nodes in each plane. Connections go from nodes in one node plane to the next, alternating in horizontal and in vertical branch planes. Every node in a horizontal plane is connected to all  $q$  nodes in that plane and every node in a vertical plane is connected to all  $r$  nodes in that plane. The design will guarantee that any two code sequences will differ in at least three branches.

The design gives the trellis a block structure, each block containing one segment with horizontal connections and one with vertical connections. It is assumed that the first segment will have horizontal connections. Blocks are enumerated  $k = 1, 2, \dots$  and nodes are given coordinates  $(i, j)$  with  $i = 0, 1, \dots, q - 1$  and  $j = 0, 1, \dots, r - 1$ . The coordinates  $(0, 0)$  will define the reference nodes and branch planes through these nodes will be called reference planes. A node line is defined by connections between nodes with identical coordinates.

## Code Construction

Codes are constructed for phase modulation with  $M = qr$  phase values. These are identified by the set of integers  $\mathcal{I}_M = (0, 1, \dots, M - 1)$ , being used as coefficients of  $2\pi/M$  to define the phase values. A semi-infinite code sequence  $\underline{c}$  will be written  $(\underline{c}(1), \underline{c}(2), \dots)$  with  $\underline{c}(k) = (c_1(k), c_2(k))$  being the code sequence for code and trellis block  $k$ . Coefficient  $c_1(k)$  will be the label on the branch in the horizontal plane and  $c_2(k)$  on the branch in the vertical plane defining the path through the trellis. We may talk about code blocks with code variables defined for odd and even trellis segments.

The set of  $M$  integers is partitioned into sets in two ways to fit the trellis design for odd and even segments. For odd segments define the sets  $\mathcal{A}_s$ ,  $s = 1, 2, \dots, r$ , each with  $q$  elements and for even segments the sets  $\mathcal{B}_t$ ,  $t = 1, 2, \dots, q$ , each with  $r$  elements. An algebraic structure is introduced by defining a suitable ring  $\mathbf{R}_M$  with  $\mathcal{A}_0$  and  $\mathcal{B}_0$  being additive subgroups and the other sets their cosets.

A code is constructed by mapping cosets onto branch planes, i.e. branches are assigned labels from one coset only. For the reference planes, labels are taken from  $\mathcal{A}_0$  for odd segments and from  $\mathcal{B}_0$  for even segments, for other planes from their cosets.

First, enumeration of branches in the odd segments will be treated. Let branch parameters  $\underline{a}(k) = (a_0, a_1, \dots, a_{q-1})$  and  $\tilde{\underline{a}}(k) = (\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_{q-1})$  with  $a_0 = \tilde{a}_0 = 0$ , be the elements of  $\mathcal{A}_0$  ordered in two ways. Further, define coset coefficients  $\underline{e}(k) = (e_0, e_1, \dots, e_{r-1})$  with  $e_0 = 0$  and  $e_s$  the leader of coset  $\mathcal{A}_s$ . The branch from node  $i$  to node  $i'$  in the horizontal plane  $j$  will be given label  $a(i, i'; j)$  with

$$a(i, i'; j) = a_i + \tilde{a}_{i'} + e_j; \quad i, i' = 0, 1, \dots, q - 1 \text{ and } j = 0, 1, \dots, r - 1 \quad (1)$$

Clearly  $a(i, i'; j)$  is an element of  $\mathcal{A}_j$  and  $a_i$  is the label on the branch from node  $i$  in the horizontal reference plane to the reference node in the next node plane;  $\tilde{a}_{i'}$  is the label on the branch from the reference node in the first node plane to node  $i'$  in the next node plane of the horizontal reference plane.

Turning next to the even segments. Let branch parameters  $\underline{b} = (b_0, b_1, \dots, b_{r-1})$  and  $\tilde{\underline{b}} = (\tilde{b}_0, \tilde{b}_1, \dots, \tilde{b}_{r-1})$  with  $b_0 = \tilde{b}_0 = 0$  be the elements of  $\mathcal{B}_0$  ordered in two ways and define coset parameters  $\underline{f} = (f_0, f_1, \dots, f_{q-1})$  with  $f_0 = 0$  and  $f_t$  the leader of coset  $\mathcal{B}_t$ . The branch from node  $j$  to node  $j'$  in vertical plane  $i'$  will be given label  $b(j, j'; i')$  with

$$b(j, j'; i) = b_j + \tilde{b}_{j'} + f_{i'}, \quad (2)$$

This is clearly an element of  $\mathcal{B}_{i'}$  and  $b_j$  and  $\tilde{b}_{j'}$  can be seen as labels on branches from node  $j$  to the reference node and from the reference node to node  $j'$  in the vertical reference plane.

Indices have been chosen so that  $(a(i, i'; j), b(j, j'; i'))$  defines a code block starting from node  $(i, j)$  going to node  $(i', j')$  at the end of the code block.

**Statement 1** All branches that meet in a node have different labels.

Proof: The statement follows from the group structure. Take an odd segment as an example and note that the set  $\{a_i + \tilde{a}_{i'}; i' = 0, 1, \dots, q - 1\}$  for fixed  $a_i$  equals the group  $\mathcal{A}_0$ , hence labels are different for nodes in the reference plane. This property is preserved for nodes in other planes since adding  $e_j$  will map the group onto a coset.  $\square$

When branches of all odd segments have the same enumeration, and those of all even segments, then the trellis design will have a period of two symbols. With different enumeration in different blocks longer periods may be created. A design of period two will be completely specified by the sequences  $\underline{a}, \underline{\tilde{a}}, \underline{b}, \underline{\tilde{b}}, \underline{e}$  and  $\underline{f}$ . Given  $q$  and  $r$ , branch and coset parameters may be chosen in several ways.

Design examples will be given in Section III.A.

## Algebraic Basis

A ring of  $M$  integers may be specified in several ways. The most natural way may be to take the ring of integers  $\mathbf{Z}_M$  with addition defined modulo  $M$ . It turns out that using only this definition will severely restrict code construction. It is then motivated to also consider the polynomial ring  $\mathbf{P}_M$  defined over a primary field  $\text{GF}(p)$ . An element may be written  $g(x) = g_0 + g_1x + \dots + g_{m-1}x^{m-1}$  with  $g_i$  a member of  $\text{GF}(p)$ . This will make  $M = p^m$  and allow the factorization  $M = qr$  with  $q = p^\mu$  and  $r = p^\nu$  making  $m = \mu + \nu$ .

Next define the following sets:

$$\mathcal{G}_0 = (0, r, \dots, (q-1)r); \quad \mathcal{H}_0 = (0, q, \dots, (r-1)q) \quad (3)$$

**Statement 2**  $\mathcal{G}_0$  and  $\mathcal{H}_0$  are additive subgroups of  $\mathbf{Z}_M$  and  $\mathbf{P}_M$ .

Proof: The statement is clearly true for  $\mathbf{Z}_M$ . With the polynomial ring a transformation is required from the ring  $\{g(x)\}$  to the integer set  $\mathcal{I}_M = \{i\}$  which will be given by  $i = g(p)$ . Operations will be carried out in polynomial space. Next note that the integer  $q$  has the polynomial representation  $x^\mu$  and  $r$  has  $x^\nu$ . The polynomial representation of  $\mathcal{G}_0$  is the set  $\{g_1(x)x^\nu\}$  with  $g_1(x)$  a polynomial of degree  $\mu - 1$ . Similarly the representation of  $\mathcal{H}_0$  is the set  $\{g_2(x)x^\mu\}$  with  $g_2(x)$  a polynomial of degree  $\nu - 1$ . By definition, the additive group properties are satisfied in the polynomial space when coefficients in  $g_1(x)$  and  $g_2(x)$  are allowed to take on all possible values.  $\square$

Given the subgroups  $\mathcal{G}_0$  and  $\mathcal{H}_0$  their cosets are found by adding integers  $1, 2, \dots, r-1$  and  $1, 2, \dots, q-1$ , respectively, to elements of the subgroups. For further reference expressions are given formally

$$\mathcal{G}_s = (s, s+r, \dots, s+(q-1)r); \quad \mathcal{H}_t = (t, t+q, \dots, t+(r-1)q) \quad (4)$$

For the polynomial ring one needs the polynomial representations that will give cosets  $\{g_2 + g_1(x)x^\nu\}$  and  $\{g_1 + g_2(x)x^\mu\}$ , respectively.

Subgroups  $\mathcal{G}_0$  and  $\mathcal{H}_0$  and selected leaders of their cosets will be used as basis for code design, as described earlier. They match the number of nodes and branch planes in the respective trellis segment. They also give the best possible separation in signal space measured by the Euclidean distance between branches leaving from or meeting at a given node.

## Code Equivalence

Two code designs are said to be equivalent if one design can be transformed into the other by an operation that preserves mutual Euclidean distances in signal space between code sequences. It is sufficient to consider sequences of the same length as the code design period. Two such operations will be defined.

The first operation, applicable to operations in  $\mathbf{Z}_M$  only, is addition of a fixed element  $\gamma$  to all labels in a segment. This may be interpreted as adding  $\gamma$  to all coset parameters of that segment, hence coset sequence  $\underline{e}$  is transformed into  $\underline{e} + \gamma\underline{1}$  with  $\underline{1} = (1, 1, \dots, 1)$ . It is easy to see that this operation will not change the Euclidean distance between any two points on the circle.

By adding a suitable number  $\gamma$ , any branch in the segment may get the label 0 and, in particular, any node line may get this label. As a consequence any node line may be taken as reference.

The other operation applies in general and consists of simultaneous permutation of coset parameters in one segment and identical permutation of branch labels in neighbouring segments. As an example interchange  $(a_{i'}, f_{i'})$  and  $(a_k, f_k)$  or  $(e_j, b_j)$  and  $(e_m, b_m)$  in a trellis block. The result will be the interchange of code blocks  $(a(i, i'; j), b(j; j'; i'))$  and  $(a(i, k; j), b(j, j'; k))$ , and then the interchange of the first one and  $(a(i, i'; m), b(m, j'; i'))$ . If an interchange is being done also at the border of the code blocks, the result will be the interchange of connected code sequences. The set of distances from these two code sequences to other sequences will be unaffected by the change of position in the trellis. Note that the interchange of parameters will affect labels on all branch planes.

The result is that the ordering of cosets in the first segment can be standardized so that in  $\underline{e}$  component  $e_j$  is taken from coset  $\mathcal{G}_j$  according to Equation (4). Similarly, the order of components in  $\underline{a}$  can be standardized to  $\underline{a} = (0, r, \dots, (q-1)r)$  by permuting vertical planes in the even segments, or the ordering of cosets is standardized, so that  $f_i \in \mathcal{H}_i$  according to Equation (4). The latter alternative is preferred and will be used in the design examples.

## Encoding and Decoding

Encoding will be made blockwise. The node coordinates at the beginning and end of a block are needed to calculate branch labels according to Equations(1) and (2). Denote the coordinates at the end of block  $k$ :  $s(k) = (i(k), j(k))$  and call  $s(k)$  the code state for this block, then  $\underline{c}(k)$  is a function of  $s(k-1)$  and  $s(k)$ . Specifically encoding of block  $k$  may be described as setting, in Equations (1) and (2),  $i = i(k-1)$ ,  $j = j(k-1)$  and  $i' = i(k)$ ,  $j' = j(k)$ .

Let the data source produce blocks of data  $(i(k), j(k))$  for  $k = 0, 1, \dots$ . Encoding will map the data into code states  $s(0), s(1), \dots$  from which branch labels are calculated as described. The data rate per symbol will on the average be

$$R = [\log_2 q + \log_2 r]/2 \quad (5)$$

This is half the rate of uncoded transmission with M-PSK.

Another way of encoding is to let the data source generate a sequence of coset parameters  $(e(k), f(k))$  with  $e(k)$  the coset parameter for the odd segment in block  $k$  and  $f(k)$  for the even segment. Given  $(e(k), f(k))$  and  $(e(k+1), f(k+1))$  the code state  $s(k)$  of block  $k$  may be found. Coordinate  $i(k)$  is found from  $f(k)$  and  $j(k)$  from  $e(k+1)$  by table look up. From that point encoding will proceed as described already.

This way of encoding may be looked upon as generating a kind of coset code but not the type described and analyzed by Forney [3]. Decoding will, of course, involve the Viterbi algorithm with a metric defined by the Euclidean distance in signal space between received and coded signals. The result may be described as a sequence of estimated code states from which estimated data may be read.

### III Code Structure

#### Code Linearity

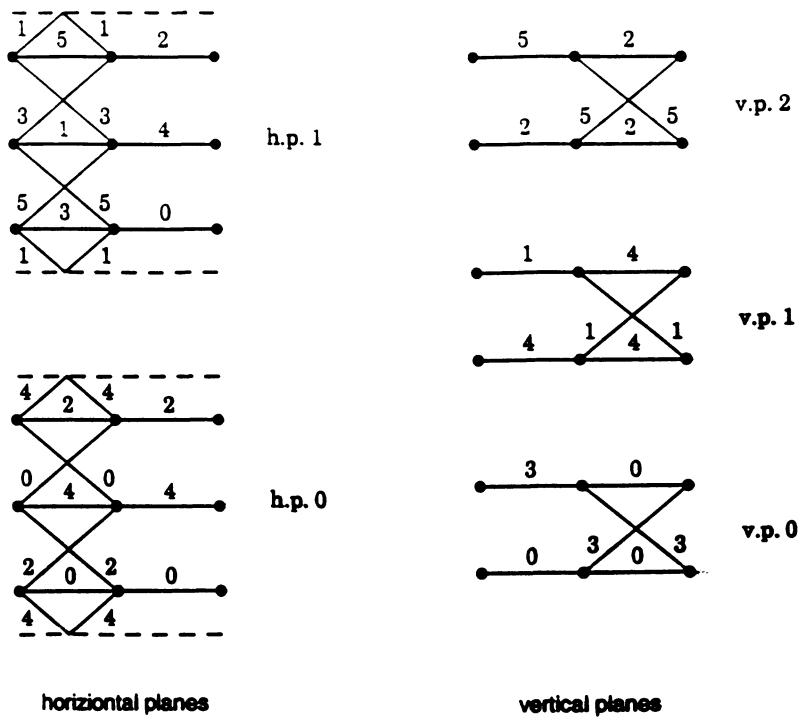
Code design maps cosets of the appropriate subgroup onto branch planes and this guarantees that the number of cosets equals the number of planes. It follows that addition of labels for branches within a code segment gives labels for some branches within the same segment. Linearity holds for labels in a code segment. It may be shown that this is true also for code blocks  $(c_1(k), c_2(k))$ .

Linearity for code blocks does not guarantee linearity for longer sequences as shown by examples in [1] with operations in  $\mathbf{Z}_M$ . Take sequences made up of four symbols from two consecutive blocks. Addition of symbols from the first block will produce a new code sequence and so will addition of symbols from the second block but the two sequences in general will not be connected and hence the result will not be a code sequence. In the following analysis it is enough to consider sequences consisting of two connected code blocks. Results apply to operations both in  $\mathbf{Z}_M$  and  $\mathbf{P}_M$ .

**Statement 3** Necessary and sufficient conditions for a trellis code of design period two to be linear is that for any pair of node coordinates  $(i, j)$  and  $(m, n)$  there exists a node coordinate  $(s, t)$  such that

$$\begin{aligned} a_i + a_m &= a_s; & \tilde{a}_i + \tilde{a}_m &= \tilde{a}_s; & f_i + f_m &= f_s; & \text{and} \\ b_j + b_n &= b_t; & \tilde{b}_j + \tilde{b}_n &= \tilde{b}_t; & e_j + e_n &= e_t \end{aligned} \quad (6)$$

Conditions imply consistent transformation of branch and coset parameters.



**Figure 1: Code design  $M=6$ ;  $q=3$ ,  $r=2$ .**

Proof: Suppose conditions apply, then consider two code sequences, each consisting of two connected code blocks defined by state sequences  $(i, j)$ ,  $(i', j')$ ,  $(i'', j'')$  and  $(m, n)$ ,  $(m', n')$ ,  $(m'', n'')$ . The code sequences may be written;

$$\begin{aligned}\underline{c}_1 &= (a(i, i'; j), b(j, j'; i'), a(i', i''; j'), b(j', j''; i'')) \\ \underline{c}_2 &= (a(m, m'; n), b(n, n'; m'), a(m', m''; n'), b(n', n''; m''))\end{aligned}$$

Adding these two sequences relations (6) will be applied. It is then convenient to express these as index additions in the following way:

$$\begin{array}{lll} \text{for } \underline{a}, \tilde{\underline{a}} \text{ and } \underline{f}: & [i] + [m] = [s]; & [i'] + [m'] = [s']; \\ & [i''] + [m''] = [s''] & \text{and} \\ \text{for } \underline{b}, \tilde{\underline{b}} \text{ and } \underline{e}: & [j] + [n] = [t]; & [j'] + [n'] = [t']; \\ & [j''] + [n''] = [t''] & \end{array}$$

The result will be the sequence  $(a(s, s'; t), b(t, t'; s'), a(s', s''; t'), b(t', t''; s''))$  which clearly is a code sequence.

Necessity is proved by taking special sequences, namely those which depart from a common node and then proceed in parallel. It is convenient to let sequences start from a reference node. Take first the node at the beginning of a code block and note that labels may be written  $(a_i, f_i)$  and  $(a_m, f_m)$  for code sequences ending in nodes  $(i, 0)$  and  $(m, 0)$ , respectively. Adding these will give  $(a_i + a_m), (f_i + f_m)$ . Suppose the result will be  $(a_s, f_k)$ , then it will describe labels for a connected path if and only if  $s = k$ . The result will be that the first three relations in Equation (6) will apply. Repeating the procedure starting from the middle of a code block will show that also the last three relations in Equation (6) are necessary.  $\square$

As a corollary it follows that coset leaders  $e_e$  and  $f_i$  must form groups. It is reasonable to next investigate under what conditions this is true. We start with operations in  $\mathbf{Z}_M$ .

**Statement 4** Coset leaders to subgroups  $\mathcal{G}_0 = \{jr\}$  and  $\mathcal{H}_0 = \{iq\}$  in  $\mathbf{Z}_M$  with  $M = qr$  will form groups if and only if  $q$  and  $r$  are relatively prime.

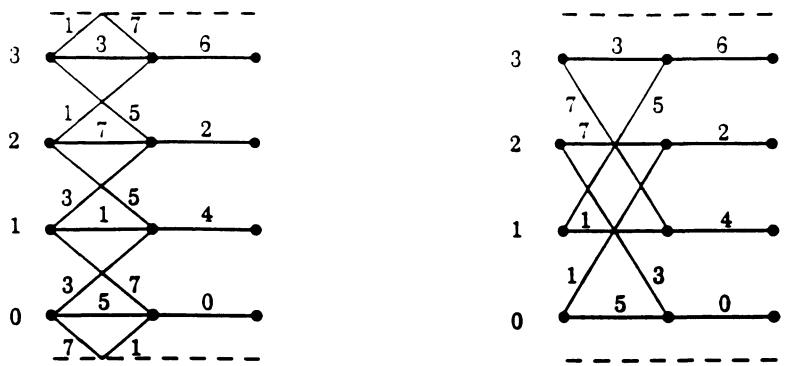
Proof: Suppose  $q$  and  $r$  are relatively prime, then  $iq$  and  $i'q$  belong to different cosets to  $\mathcal{G}_0$ . If not  $iq = s + jr$  and  $i'q = s + j'r$  for some  $s, j$  and  $j'$ . This leads to the relation

$$(i - i')q = (j - j')r; \quad \text{for } i, i' = 0, 1, \dots, r - 1; \quad j, j' = 0, 1, \dots, q - 1 \quad (7)$$

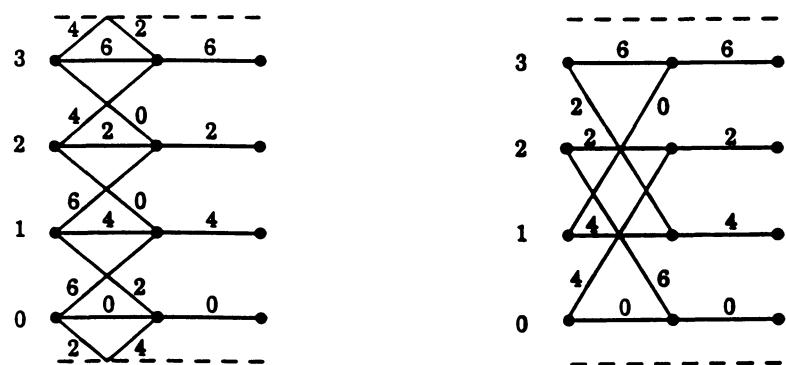
With  $q$  and  $r$  relatively prime  $i = i'$  and  $j = j'$  is the only solution. Hence the set  $\mathcal{H}_0$  may be taken as leaders to cosets of  $\mathcal{G}_0$  and vice versa.

Suppose next that leaders of cosets  $\mathcal{G}_0$  form a group, then it can only be the set  $\mathcal{H}_0$  since  $\mathcal{H}_0$  is the unique group with  $r$  elements. It then should not happen that  $iq$  and  $i'q$  belong to the same coset. This requires Equation (7) to have only the trivial solution  $i = i'$  and  $j = j'$  which only happens if  $q$  and  $r$  are relatively prime.  $\square$

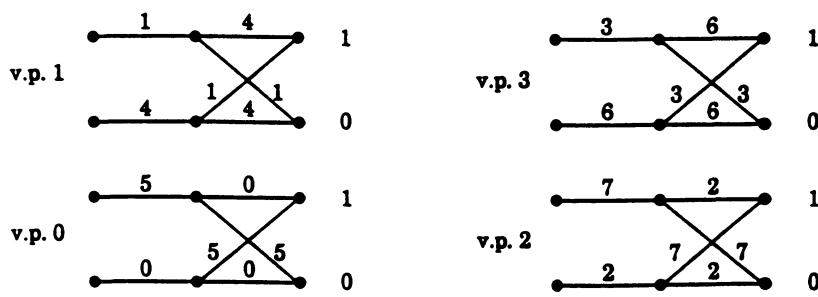
**Corollary 1** A code is linear over  $\mathbf{Z}_M$ , with  $M = qr$ , if and only if  $q$  and  $r$  are relatively prime.



horizontal plane 1



horizontal plane 0



vertical planes

**Figure 2:** Code design  $M=8$ ;  $c=4$ ,  $r=2$ .

It follows that linearity will make a code a group code.

### Design example 1: $M = 6, q = 3, r = 2$

$$\mathcal{G}_0 = (0, 2, 4); \mathcal{G}_1 = (1, 3, 5); \text{ and } \mathcal{H}_0 = (0, 3); \mathcal{H}_1 = (1, 4); \mathcal{H}_2 = (2, 5);$$

There are now two possibilities for each  $\underline{a}$  and  $\tilde{\underline{a}}$ . We will choose  $\underline{a} = \tilde{\underline{a}} = (0, 2, 4)$ . Branch parameters  $\underline{b} = \tilde{\underline{b}} = (0, 3)$  are uniquely determined. There are three possible choices of coset parameters  $\underline{e}$ , namely  $(0,1), (0,3)$  and  $(0,5)$ . Taking  $\underline{e} = (0,3)$  will make it a group code and also give good separation in signal space. Four possibilities appear to choose  $\underline{f}$  but only  $\underline{f} = (0, 4, 2)$  will make it a group. Label calculations are carried out with addition modulo six. Figure 1 illustrates the design. By construction the code is linear with addition modulo six.  $\square$

Requiring  $q$  and  $r$  to be relatively prime severely restricts the possibilities to construct linear codes. If instead  $q$  and  $r$  are powers of a common prime number  $p$  new possibilities appear with operations in polynomial space. It will include most common communication schemes, in particular for  $p = 2$  and 3.

It appears that linear codes exist for all possible values of  $M = p^m$  but no attempt will be made to give a proof. Instead the special case  $M = 8$  will be treated in detail.

### Design example 2: $M = 8, q = 4, r = 2$

$\mathcal{G}_0 = (0, 2, 4, 6); \mathcal{G}_1 = (1, 3, 5, 7); \text{ and } \mathcal{H}_0 = (0, h)$ ; will be a subgroup for all values of  $h$  but to avoid contradictions when calculating cosets,  $h$  should be either 5, 6 or 7. Take  $h = 5$ , which gives  $\mathcal{H}_0 = (0, 5)$  and  $\mathcal{H}_1 = (1, 4), \mathcal{H}_2 = (2, 7)$  and  $\mathcal{H}_3 = (3, 6)$ . This makes  $\underline{b} = \tilde{\underline{b}} = (0, 5)$  the only choice according to Section II.B. Out of four possible choices of  $\underline{e}$ , all groups over the binary field,  $(0,5)$  is selected to give good separation in signal space. Next coset parameters  $\underline{f}$  will be selected. Out of eight possibilities, the following four generate groups in polynomial space:

1.  $\underline{f} = (0, 1, 2, 3) = (0, 1, x, 1 + x)$
2.  $\underline{f} = (0, 1, 7, 6) = (0, 1, 1 + x + x^2, x + x^2)$
3.  $\underline{f} = (0, 4, 2, 6) = (0, x^2, x, x + x^2)$
4.  $\underline{f} = (0, 4, 7, 3) = (0, x^2, 1 + x + x^2, 1 + x)$

$\underline{f} = (0, 4, 2, 6)$  will be selected to make rotational invariance possible. For the same reason  $\underline{a} = (0, 2, 6, 4)$  and  $\tilde{\underline{a}} = (0, 6, 4, 2)$  are chosen. The result of label calculations is found in Figure 2.  $\square$

### Rotational Invariance

To achieve coherent detection the receiver must know the carrier reference phase which is not always the case due to unknown channel characteristic. One way around the problem is to design the code so that decoding is unaffected by certain phase errors. The standard technique is to perform differential precoding of data from the source before encoding. A fixed phase error shows up as a phase rotation of the transmitted signal. It is required that

this sequences can be decoded and moreover that the phase error gives rise to addition of a fixed number to the decoded sequence. Performing the inverse operation to precoding, the original sequence can be recovered if no decoding errors have occurred.

The standard technique will be modified before being applied to our coding scheme. It is more convenient to make differential encoding on the encoded symbols rather than on the data symbols, hence we may talk about doing postcoding.

Let once more  $\underline{c} = (\underline{c}(1), \underline{c}(2), \dots, \underline{c}(k), \dots)$  with  $\underline{c}(k) = (c_1(k), c_2(k))$  be the primary encoded sequence, then the result of postcoding may be described as giving the sequence  $\underline{C} = (\underline{C}_1, \underline{C}_2, \dots, \underline{C}_k, \dots)$  with  $\underline{C}(k) = (C_1(k), C_2(k))$  and  $\underline{C}(k) = \underline{C}(k-1) + \underline{c}(k)$ . Assume that the channel introduces a phase error  $\lambda \cdot 2\pi/M$  for some integer  $\lambda$ , then  $C_1(k)$  will be transformed into  $C'_1(k) = C_1(k) + \lambda$  and similarly for  $C_2(k)$ . The result is a sequence  $\underline{C}' = (\underline{C}'_1, \underline{C}'_2, \dots, \underline{C}'_k, \dots)$ .

To perform decoding requires  $\underline{C}$  and  $\underline{C}'$  to be code sequences. This will be true for  $\underline{C}$ , if and only if the code is linear. To make  $\underline{C}'$  decodable requires  $\underline{\lambda} = ((\lambda, \lambda), (\lambda, \lambda), \dots)$  to be a code sequence. It appears that this in turn requires that branches on one of the node lines have the label  $\lambda$ . This is possible if  $a_i + \tilde{a}_i + e_j = \lambda$  and  $b_j + \tilde{b}_j + f_i = \lambda$  for some state coordinates  $(i, j)$ . Two cases are of special interest, namely  $i = 0$  and  $j = 0$ . This leads to the following two alternatives:

$$(1) \ j = 0 : \quad a_i + \tilde{a}_i = \lambda \text{ and } f_i = \lambda \quad \text{or} \quad (2) \ i = 0 : \quad b_j + \tilde{b}_j = \lambda \text{ and } e_j = \lambda \quad (8)$$

Possible values of  $\lambda$  in the former case belong to  $\mathcal{G}_0$  and in the latter case to  $\mathcal{H}_0$ , hence they are multiples of  $r$  and  $q$ , respectively. It follows that possible phase errors may be multiples of  $2\pi/q$  and  $2\pi/r$ , respectively.

The two design examples are both rotational invariant at decoding and give for  $M = 6$ ;  $\lambda = 2$  and  $4$  while for  $M = 8$ ;  $\lambda = 2, 4$  and  $6$ . As a result in the former case decoding is insensitive to phase shifts  $2\pi/3$  and  $4\pi/3$  while in the latter case this is true for  $\pi/2$ ,  $2\pi/2$  and  $3\pi/2$ .

To put these results into perspective one may compare them with published data. Biglieri et al. [4] thoroughly treat rotational invariant codes and give as example a design of an 8-state Ungerboeck code to be used with 8-PSK transmission. They get the same rotational invariance as for our code. It may be observed that their code transmits 2 bit/symbol, ours 1.5, the reason being that their trellis includes parallel transitions between nodes, ours not.

## IV Acknowledgement

Work on this paper was initiated after discussions with Jim Massey at the Mölle conference. He asked if one could design codes, my way, for which decoding was unaffected by certain phase shifts on the received signals. It is a pleasure for me to be able to report some results on the subject at the seminar in honour of Jim Massey.

## References

- [1] L. H. Zetterberg, *Coded Phase Modulation with Geometrically Designed Trellises and Coset Mapping*, TRITA-SB-9308, Royal Institute of Technology (KTH), Stockholm, Sweden, 1993.
- [2] L. H. Zetterberg, “A class of trellis codes for phase modulation based on geometrical design and coset mapping”, *Proc. Sixth Joint Swedish-Russian International Workshop on Information Theory*, August 22-27 1993, Mölle, Sweden, pp 23-27.
- [3] G. D. Forney Jr., “Coset codes - part I: Introduction and geometrical classification”, *IEEE Trans. on Inform. Theory*, Vol. 34, No. 5, pp. 1123-1151, 1988.
- [4] E. Biglieri, D. Divsalar, P. J. McLane and M. K. Simon, *Introduction to Trellis Coded Modulation with Applications*, McMillan Publishing Co., New York, 1991.

# Algebraic-Sequential Decoding - Ideas and Results

K. Sh. Zigangirov

Dept. of Information Theory

Lund University S-221 00 LUND, Sweden

on leave from the Institute for Problems of Information Transmission,  
Moscow, Russia

## Abstract

Algebraic-sequential decoding of convolutional codes is described and a statistical analysis of the decoder is given. It is shown that for nonbinary alphabets these decoders provide a gain in decoding complexity over conventional sequential decoders.

## I Introduction

Algebraic and probabilistic coding form two trends of the modern coding theory that are little intersected. As a rule, algebraists, working in coding theory, are allergic to probabilistic coding and vice versa. (Jim Massey is one of a very few exceptions to this rule.) While algebraic encoding and decoding algorithms are mostly being used for block codes, various probabilistic algorithms are mostly being used for convolutional codes (Viterbi decoding, sequential decoding, etc.). Moreover, it is difficult to adapt algebraic approaches to channels, having different statistical characteristics, and the algebraic structure of the specific code is usually not taken into account when probabilistic approaches to the decoding are developed.

In this paper we deal with a decoding procedure of convolutional codes that is based both on algebraic and probabilistic approaches. The research was started together with Dan Costello in 1990 during the author's visit to the University of Notre Dame [1] and was presented at the IEEE International Workshop on Information Theory in Veldhoven, 1990. This paper is the first publication of the algebraic-sequential decoding in English.

The main part of the paper is devoted to the description and analysis of convolutional codes over  $GF(q)$ , where  $q > 2$ . To make the idea clear, we first describe the decoding procedure for block codes and then generalize it to convolutional codes.

## II Decoding of Block Codes

Let us consider a discrete symmetric memoryless channel (DSMC) with input and output alphabets  $\{0, 1, \dots, q-1\}$ . Since the channel is memoryless, each output letter of the channel is a function only of the corresponding input letter. Let  $p_{ij}$  be the conditional probability of reception of the  $j$ -th,  $j = 0, \dots, q-1$ , letter provided that the  $i$ -th letter has been

transmitted. According to the definition of the DSMC,

$$p_{ij} = \begin{cases} 1 - \varepsilon, & \text{if } i = j, \\ \varepsilon/(q-1), & \text{otherwise,} \end{cases} \quad (1)$$

where  $0 < \varepsilon < 1$ . Then, the probability to receive the word

$$\mathbf{r} = (r_0, r_1, \dots, r_N), \quad r_i \in GF(q), \quad i = 0, \dots, N,$$

given the transmission of the codeword

$$\mathbf{v} = (v_0, v_1, \dots, v_N), \quad v_i \in GF(q), \quad i = 0, \dots, N,$$

is equal to

$$P(\mathbf{r}|\mathbf{v}) = (1 - \varepsilon)^{N+1-d(\mathbf{r}, \mathbf{v})} \left( \frac{\varepsilon}{q-1} \right)^{d(\mathbf{r}, \mathbf{v})},$$

where  $d(\mathbf{r}, \mathbf{v})$  is the Hamming distance between  $\mathbf{r}$  and  $\mathbf{v}$ .

The component-by-component difference in  $GF(q) : \mathbf{n} = \mathbf{r} - \mathbf{v}$  is called a noise sequence.

Let  $\mathbf{v}$  be the output (code) sequence of a rate  $(K+1)/(N+1)$ -encoder which corresponds to the input (data) sequence

$$\mathbf{u} = (u_0, u_1, \dots, u_K), \quad u_i \in GF(q), \quad i = 0, \dots, K$$

by the relation

$$\mathbf{v} = \mathbf{u}\mathbf{G},$$

$$\mathbf{G} = \begin{vmatrix} g_{00} & g_{01} & \cdot & \cdot & \cdot & g_{0N} \\ g_{10} & g_{11} & \cdot & \cdot & \cdot & g_{1N} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{K0} & g_{K1} & \cdot & \cdot & \cdot & g_{KN} \end{vmatrix}.$$

To simplify an analysis, we suppose that the parameters of the code satisfy the Varshamov-Gilbert bound :

$$R = 1 - \varrho \log_q(q-1) + \varrho \log_q(\varrho) + (1-\varrho) \log_q(1-\varrho), \quad (2)$$

where  $\varrho = d_{\min}/(N+1)$ , and  $d_{\min}$  is the minimum distance of the code. Besides, we suppose that the covering radius of the code is less than  $d_{\min}$  (otherwise, we can add new codewords to the code and keep  $d_{\min}$  as its minimum distance).

Let us consider the maximum likelihood (ML) decoding of the code that satisfies the restrictions above. When implemented in the obvious way, ML decoding has complexity

$$C \sim \min(q^{(N+1)R}, q^{(N+1)(1-R)}),$$

by which we mean the number of comparisons of the received sequence  $\mathbf{r}$  and the codewords  $\mathbf{v}$ . We will prove that the complexity of the ML decoding for DSMC can be written as :

$$C \sim q^{(N+1)(1-R)}(q-1)^{-d_{\min}}. \quad (3)$$

First, let us show that if the decoder knows the error locations then it can easily find the ML decoded codeword. Indeed, let  $\mathbf{l} = (l_0, l_1, \dots, l_N)$  be the sequence of error locators :  $l_i \in \{c, e\}$ , where  $l_i = c$  ('correct') if the  $i$ -th symbol of the codeword has been received correctly, and  $l_i = e$  ('erroneous') otherwise. The number of symbols  $e$  will be referred to as the weight of the sequence  $\mathbf{l}$ .

Let us describe a decoding procedure that constructs an information sequence  $\mathbf{u}$  for a given received sequence  $\mathbf{r}$  and a sequence of error locators  $\mathbf{l}$ . The decoder starts with puncturing the symbols  $r_i$  from  $\mathbf{r}$  for all  $i$  such that  $l_i = e$ . Let  $\tilde{\mathbf{r}}$  be the obtained (punctured) sequence. Then the decoder punctures all columns of the matrix  $\mathbf{G}$  that correspond to the punctured symbols in  $\mathbf{r}$ . Let  $\tilde{\mathbf{G}}$  be the obtained matrix. After that the decoder solves the system of linear equations :

$$\mathbf{u}\tilde{\mathbf{G}} = \tilde{\mathbf{r}} \quad (4)$$

with respect to  $\mathbf{u}$ . The solution  $\tilde{\mathbf{u}}$  is declared as the decoded sequence.

If the error locators are determined correctly, i.e., the sequence  $\mathbf{l}$  is correct, then the decoder punctures only erroneous symbols from  $\mathbf{r}$ . The nonpunctured symbols of  $\mathbf{r}$  coincide with corresponding symbols of the sequence  $\mathbf{v}$ . In this case, system (4) always has a solution. If the solution is unique, it is correct, but if it is not unique, then the maximum-likelihood decoder also cannot make a unique correct decision.

If the error locator is incorrectly determined, and the weight of  $\mathbf{l}$  is less than or equal to the number of errors in the channel, then :

- (a) either system (4) does not have a solution, which is an indication that  $\mathbf{l}$  is incorrect, or
- (b) system (4) has a solution, and the maximum-likelihood decoder also cannot make a unique correct decision.

Now, we describe the decoding for unknown sequence  $\mathbf{l}$ . First, the decoder checks whether (4) has a solution, when  $\mathbf{l}$  consists of all symbols  $c$  (the weight of  $\mathbf{l}$  is equal to zero), or not. This case corresponds to the assumption that there were no errors during transmission. If a solution exists, then it is the ML estimate of the transmitted sequence.

Otherwise, the decoder examines  $\binom{N+1}{1}$  cases when  $\mathbf{l}$  has the weight 1. If a solution (for at least one  $\mathbf{l}$ ) exists, then it also gives the ML estimate of the transmitted sequence.

Otherwise, the decoder examines  $\binom{N+1}{2}$  cases when  $\mathbf{l}$  has the weight 2, etc. Since the covering radius of the code does not exceed  $d_{min} - 1$ , the number of computation does not exceed

$$C = \sum_{i=0}^{d_{min}-1} \binom{N+1}{i} \sim 2^{(N+1)H(\varrho)}, \quad (5)$$

where

$$H(\varrho) = -\varrho \log_2 \varrho - (1 - \varrho) \log_2 (1 - \varrho)$$

is the binary entropy function, and  $\varrho = (d_{min} - 1)/(N + 1)$ .

Using (2), we can write :

$$(N + 1)H(\varrho) = (N + 1)[1 - R - \varrho \log_q (q - 1)] \log_2 q.$$

Therefore, (5) leads to (3).

The algebraic-sequential decoding of convolutional codes is based on the same idea as the described algorithm.

### III Decoding of Convolutional Codes

Let  $\mathbf{v}|_0^n$  be the output (code) sequence of the length  $n + 1$ , where  $n = 0, 1, \dots, N + m$ , which was generated at the output of the convolutional encoder of rate  $R = b/c$  and memory  $m$ :

$$\mathbf{v}|_0^n = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n),$$

where

$$\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{ic}), \quad v_{ik} \in GF(q), \quad i = 0, \dots, n, \quad k = 1, \dots, c,$$

to transmit the input (data) sequence

$$\mathbf{u}|_0^n = (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_n),$$

where

$$\mathbf{u}_i = (u_{i1}, u_{i2}, \dots, u_{ib}), \quad u_{ik} \in GF(q), \quad i = 0, \dots, n, \quad k = 1, \dots, b.$$

Then

$$\mathbf{v}|_0^n = \mathbf{u}|_0^n \mathbf{G}|_0^n.$$

Here

$$\mathbf{G}|_0^n = \begin{vmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_m & \mathbf{0} & \dots & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \dots & \dots \\ \mathbf{0} & \mathbf{G}_0 & \dots & \mathbf{G}_{m-1} & \mathbf{G}_m & \dots & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \dots & \dots \\ \cdot & \cdot & \dots & \cdot & \cdot & \dots & \cdot & \dots & \cdot & \cdot & \dots & \dots \\ \cdot & \cdot & \dots & \cdot & \cdot & \dots & \cdot & \dots & \cdot & \cdot & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \dots & \mathbf{G}_0 & \dots & \mathbf{G}_{m-1} & \mathbf{G}_m & \dots & \dots \end{vmatrix}$$

is an  $(n + 1) \times (n + 1)$ -matrix, having  $b \times c$  submatrices

$$\mathbf{G}_i = \begin{vmatrix} g_{i1}^{(1)} & g_{i2}^{(1)} & \dots & g_{ic}^{(1)} \\ g_{i1}^{(2)} & g_{i2}^{(2)} & \dots & g_{ic}^{(2)} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ g_{i1}^{(b)} & g_{i2}^{(b)} & \dots & g_{ic}^{(b)} \end{vmatrix}$$

as elements;  $\mathbf{0}$  denotes the all-zero  $b \times c$  matrix, and all operations are performed in  $GF(q)$ . As usual, the sequence  $\mathbf{u}|_0^{N+m}$  consists of  $b(N + 1)$  information symbols, followed by  $bm$  dummy zeroes.

Let

$$\mathbf{r}|_0^n = (\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_n),$$

where

$$\mathbf{r}_i = (r_{i1}, r_{i2}, \dots, r_{ic}), \quad r_{ik} \in GF(q), \quad i = 0, \dots, n, \quad k = 1, \dots, c,$$

be the received sequence of the length  $n + 1$ ;  $n = 0, 1, \dots, N + m$ .

The algebraic-sequential decoding consists of two steps. In the first step, the decoder determines the positions of erroneous symbols, i.e., the error locator sequence

$$\mathbf{l}|_0^n = (\mathbf{l}_0, \mathbf{l}_1, \dots, \mathbf{l}_n), \quad n = 0, 1, \dots, N + m,$$

where

$$\mathbf{l}_i = (l_{i1}, l_{i2}, \dots, l_{ic}), \quad l_{ik} \in \{c, e\}, \quad i = 0, \dots, n, \quad k = 1, \dots, c,$$

In the second step, the decoder forms a punctured received sequence  $\tilde{\mathbf{r}}|_0^{N+m}$  and a punctured generator matrix  $\tilde{\mathbf{G}}$ , such as it did for the block code, on the basis of  $\mathbf{l}|_0^{N+m}$ . Then the decoder solves the system of linear equations :

$$\mathbf{u}|_0^n \tilde{\mathbf{G}}|_0^n = \tilde{\mathbf{r}}|_0^n, \quad n = N + m, \quad (6)$$

with respect to  $\mathbf{u}|_0^N$ . Note that a condition for the existence of a solution of system (6) for  $n = 0, 1, \dots, N + m$  is that the rank of the matrix  $\tilde{\mathbf{G}}|_0^n$  is equal to the rank of the augmented matrix  $\tilde{\mathbf{G}}\tilde{\mathbf{r}}|_0^n$  of system (6) :

$$\text{rank } \tilde{\mathbf{G}}|_0^n = \text{rank } \tilde{\mathbf{G}}\tilde{\mathbf{r}}|_0^n. \quad (7)$$

Now we shall describe the first step of the decoding - that of finding the sequence  $\mathbf{l}$  using principles of sequential decoding. We shall describe a modification based on the stack algorithm.

The set of sequences  $\mathbf{l}|_0^n$ ,  $n = 0, 1, \dots$ , can be represented as a set of paths of an error locator tree. The nodes of the tree are connected by branches of length  $c$ , and  $2^c$  branches, that correspond to all possible combinations of  $c$  symbols belonging to  $\{c, e\}$ , leave each node. The value

$$\mu(\mathbf{l}|_0^n) = [cn - w(\mathbf{l}|_0^n)]\alpha + w(\mathbf{l}|_0^n)\beta,$$

where  $w(\mathbf{l}|_0^n)$  is the weight of the sequence  $\mathbf{l}|_0^n$ , will be referred to as the metric assigned to the node  $\mathbf{l}|_0^n$ ; the values  $\alpha > 0$  and  $\beta < 0$  will be given later.

A node is considered *alive* if the corresponding system (6) has a solution, i.e., condition (7) is satisfied; otherwise, it is considered *dead*. The decoder follows the conventional rules of a stack algorithm, i.e., at any moment it selects from the stack for continuation the path whose metric is maximal. A substantial change is that only alive nodes of the error-locator tree are entered into the stack. The aliveness of a node is determined using (7).

It can be shown [1,2], that for  $\varepsilon < \varepsilon_{comp}^{(a)}$ , where  $\varepsilon_{comp}^{(a)}$  is defined later, the choice of values

$$\alpha = \log_q \frac{1 - \varepsilon}{1 - z}, \quad \beta = \log_q \frac{\varepsilon}{z}, \quad (8)$$

where

$$z = (1 - R) \log_{q-1} q, \quad (9)$$

is optimal. It is interesting to note, that for large  $q$  we can choose

$$\alpha \approx \log_q \frac{1 - \varepsilon}{R}, \quad \beta \approx \log_q \frac{\varepsilon}{1 - R}.$$

In the next section we present the bound on the decoder performance that is the mathematical expectation of the number of computation of the decoder.

## IV Bounds on Computational Effort of the Decoder

To prove an upper bound on the average number of computations of the decoder, we deal with an ensemble of time-varying codes, as it is done in the theory of convolutional codes, and estimate the average number of computations to the study of incorrect subtrees of the error-locator tree. As usual, the computational cut-off rate  $R_{comp}$  is defined as supremum of transmission rates at which the average number of computations (in the ensemble of codes and noise in the channel) remains bounded, regardless of the memory of the encoder and back-search limit. It is assumed that the tree is infinite ( $m = \infty$ ).

The path of the error-locator tree that corresponds to the correct vector, consisted of error locators, will be referred to as the correct path. The set of the paths of the tree, that branch from the correct path at the  $(i - 1)$ -th node will be referred to as the  $i$ -th incorrect subtree. Let  $V^{(i)}$  be a random variable that is equal to the number of visited (alive) nodes in the  $i$ -th incorrect subtree. We are interested in an upper bound on  $M(V^{(i)})$  in the ensemble of time-varying convolutional codes and noise in the channel. Obviously, it is sufficient to obtain a bound on  $M(V^{(1)}) = M(V^{(i)})$ .

Let  $\eta$  be the minimal metric along the correct path. Then the following statement is valid [3].

**Lemma 1.**

$$P(\eta \leq x) \leq q^{-hx}, \quad (10)$$

where  $h < 0$  is a root of the equation :

$$1 = (1 - \varepsilon)q^{-h\alpha} + \varepsilon q^{-h\beta},$$

which, in view of choice (8) is equal to  $-1$ .

Let  $M(V_n)$  be the average number of visited nodes at the depth  $n+1$  in the first incorrect subtree. Then

$$M(V^{(1)}) = \sum_{n=0}^{\infty} M(V_n). \quad (11)$$

In order that an arbitrary node  $l|_0^n$  at the depth  $n$  will be visited, it is necessary to satisfy two conditions:

- 1) the node must be alive;
- 2)  $\mu(l|_0^n) \geq \eta$ .

To satisfy the first condition, in the code tree there must exist a path that coincides with the received sequence in the positions in which  $l|_0^n$  has the symbol  $c$  and does not coincide in the positions in which  $l|_0^n$  has the symbol  $e$ . Since, in the ensemble of time-varying codes, the paths of the code tree are pairwise statistically independent, the probability  $P(l|_0^n \text{ is alive})$  is upper-bounded by the product of two factors. The first is the probability that some incorrect code sequence coincides with the received sequence in the positions in which the sequence  $l|_0^n$  has the symbol  $c$  and does not coincide in the positions in which  $l|_0^n$  has the symbol  $e$ . The second factor is the upper bound  $q^{Rc(n+1)}$  of the total number of incorrect paths in the first incorrect subtree of the code tree at depth  $n$ . Thus, we obtain

the inequality :

$$\begin{aligned} P(l|_0^n \text{ is alive}) &\leq q^{Rc(n+1)} \left(\frac{q-1}{q}\right)^w \left(\frac{1}{q}\right)^{c(n+1)-w} \\ &= q^{(R-1)c(n+1)}(q-1)^w, \end{aligned}$$

where  $w = w(l|_0^n)$  is the weight of the sequence  $l|_0^n$ . Therefore,

$$P(l|_0^n \text{ is alive}) \leq \min(1, q^{(R-1)c(n+1)}(q-1)^w), \quad (12)$$

and

$$P(l|_0^n \text{ is alive}) \leq q^{\lambda(R-1)c(n+1)}(q-1)^{\lambda w} \quad (13)$$

for any  $\lambda$ ,  $0 < \lambda \leq 1$ .

On the other hand, using (10), we conclude that the second condition is satisfied with probability upper bounded by the following inequality :

$$P(\eta < \mu(l|_0^n)) \leq q^{-h\mu(l|_0^n)} = q^{\alpha[c(n+1)-w]+\beta w}. \quad (14)$$

Hence, as it follows from (13) and (14), the probability of the node  $l|_0^n$  to be visited is upper bounded by the inequality :

$$\begin{aligned} P(l|_0^n \text{ is visited}) &\leq P(l|_0^n \text{ is alive})P(\eta \leq \mu(l|_0^n)) \leq \\ &\leq q^{\lambda(R-1)c(n+1)}(q-1)^{\lambda w}q^{\alpha[c(n+1)-w]+\beta w}. \end{aligned} \quad (15)$$

However,

$$M(V_n) = \sum_{l|_0^n} P(l|_0^n \text{ is visited}), \quad (16)$$

where the sum is taken on all  $l|_0^n$  belonging to the first incorrect subtree. Using (15) and (16), we write :

$$\begin{aligned} M(V_n) &\leq \sum_{w=0}^{c(n+1)} \binom{c(n+1)}{w} q^{\lambda(R-1)c(n+1)}(q-1)^{\lambda w} q^{\alpha[c(n+1)-w]+\beta w} = \\ &= q^{\lambda(R-1)c(n+1)}(q^\alpha + (q-1)^\lambda q^\beta)^{c(n+1)}. \end{aligned} \quad (17)$$

If

$$\Lambda = q^{\lambda(R-1)}(q^\alpha + (q-1)^\lambda q^\beta) < 1, \quad (18)$$

then, using (11) and (17), we obtain :

$$M(V^{(1)}) < \frac{\Lambda^c}{1 - \Lambda^c}. \quad (19)$$

We note that (18) is the condition for the existence of the bound (19) on the average number of computation and choose  $\lambda$  to minimize  $\Lambda$  :

$$\lambda = \left( \log_q \frac{1-\varepsilon}{\varepsilon} \frac{z^2}{(1-z)^2} \right) / \log_q(q-1), \quad (20)$$

where  $z$  is defined in (9). Then

$$\Lambda = q^{-2[z \log_q \frac{z}{\sqrt{\varepsilon}} + (1-z) \log_q \frac{1-z}{\sqrt{1-\varepsilon}}]}.$$

The computational cut-off rate of the procedure corresponds to the maximal root  $z_{comp}$  of the equation :

$$\varphi(z) = z \log_q \frac{z}{\sqrt{\varepsilon}} + (1-z) \log_q \frac{1-z}{\sqrt{1-\varepsilon}} = 0, \quad (21)$$

and it gives the following value of the computational cut-off rate :

$$R_{comp}^{(a)} = 1 - z_{comp} \log_q (q-1). \quad (22)$$

Let us note that in DSMC the classical sequential decoding has the computational cut-off rate

$$R_{comp} = 1 - 2 \log_q \{(1-\varepsilon)^{1/2} + (\varepsilon(q-1))^{1/2}\}.$$

Since  $\lambda \leq 1$ , the boundary condition :  $\lambda = 1$ , according to (20), corresponds to the value of  $z_{comp}$  such that

$$z_{comp} = \frac{(\varepsilon(q-1))^{1/2}}{(\varepsilon(q-1))^{1/2} + (1-\varepsilon)^{1/2}}. \quad (23)$$

Combining (21) and (23), we obtain the boundary value  $\varepsilon = \varepsilon_{comp}^{(a)}$  as the maximal root of the equation

$$\begin{aligned} & (\varepsilon(q-1))^{1/2} \log_q \frac{(q-1)^{1/2}}{(\varepsilon(q-1))^{1/2} + (1-\varepsilon)^{1/2}} + \\ & + (1-\varepsilon)^{1/2} \log_q \frac{1}{(\varepsilon(q-1))^{1/2} + (1-\varepsilon)^{1/2}} = 0. \end{aligned} \quad (24)$$

It can be seen that

$$R_{comp}^{(a)} = R_{comp}$$

for  $\varepsilon = \varepsilon_{comp}^{(a)}$ . Thus, algebraic-sequential decoding provides a gain in computational cut-off rate when  $\varepsilon < \varepsilon_{comp}^{(a)}$ , but not when  $\varepsilon \geq \varepsilon_{comp}^{(a)}$ .

We have proved the following

**Theorem.** Algebraic-sequential decoding in a DSMC provides a gain in computational cut-off rate as compared with conventional sequential decoding when the probabilities of symbol distortion in the channel  $\varepsilon$  are smaller than  $\varepsilon_{comp}^{(a)}$  - the maximal root of equation (24). The computational cut-off rate of algebraic-sequential decoding for  $\varepsilon < \varepsilon_{comp}^{(a)}$  is given by formula (22).

Analogously to the conventional sequential decoding, an analysis of the high-order moments for number of computations can be done based on the assumption that the paths of the code tree have not pairwise but mutual statistical independence. Under this condition, we find [2]  $R_\varrho^{(a)}$  - supremum of the transmission rates at which the  $\varrho$ -th moment of computations remains bounded regardless of encoder memory and back search limit. Using this bounding we found a Pareto-type estimation of the distribution function for the number of computations of algebraic-sequential decoding :

$$P(V^{(1)} < x) \simeq 1 - x^{-\gamma},$$

where the Pareto exponent  $\gamma$  is, generally speaking, larger than that for conventional sequential decoding.

## V Discussion of Results

As was established above, the decoding process consists of two steps : 1) determination of the positions of the erroneous symbols in the received sequence; and 2) calculation of the transmitted data sequence. We estimate the decoder complexity, i.e., the number of computations required by the decoder. We note that each visit of a node of the error locator tree requires of the decoder a cycle of operations associated with determination of the aliveness of the node, calculation of its distance function, etc. It is obvious that the most time-consuming operation is that of determination of the aliveness of  $\mathbf{l}_0^n$ , which involves calculation of the ranks of the matrices  $\tilde{\mathbf{G}}|_0^n$  and  $\tilde{\mathbf{G}}_{\mathbf{r}}|_0^n$ .

It is known that calculation of the rank of a matrix has the polynomial complexity. The solution of a system of linear equations also has the polynomial complexity. Hence, we obtain the following relationship between the probability  $P$  of erroneous decoding of the received sequence and the number of operations  $C$  :

$$P \sim f(N)C^{-\gamma},$$

where  $f(N)$  is a polynomial function of  $N$ . The probability of erroneous decoding associated with the nonuniqueness of the solution of (6) at the end of the procedure is negligible, since the choice of a sufficiently large  $m$  makes this probability substantially smaller than  $P$  (the decoding complexity rises insignificantly in this case) and the corrections do not change the overall nature of the relationship between the complexity and reliability.

Thus, the overall relationship between the complexity and reliability for algebraic sequential decoding has the form  $C \sim P^{-1/\gamma}$ . The numerical calculations, that were performed, show that this relationship is better than for both conventional sequential decoding and Viterbi decoding.

Note, in conclusion, that the proposed methods are promising for decoding in channels with errors occurring in bursts.

## References

- [1] K.Sh.Zigangirov and D.J. Costello, "Algebraic-sequential decoding of convolutional codes," ( unpublished manuscript ).
- [2] K.Sh.Zigangirov, "Mathematical analysis of algebraic-sequential decoding," *Problems of Information Transmission*, V. 28, no. 1, 1992, pp.3-13.
- [3] K.Sh.Zigangirov, "Some sequential decoding procedures," *Problems of Information Transmission*, V. 2, no. 4, 1966, pp.13-25.

# Index

- achievable rate region, 409
- adder channel, 49
- Agnew, G. B., 1
- Ahlswede, R., 13
- algebraic decoding, 72
- algorithm
  - Berlekamp-Massey, 72, 209, 211, 288, 307
  - Berlekamp-Welch, 217
  - Cooley-Tukey, 144
  - Coppersmith, 223
  - Euclidean, 211
  - Fano, 73
  - gradient, 358, 360
  - Good-Thomas, 144
  - Lanczos, 213
  - Lempel-Ziv, 431
  - Levinson, 209, 215
  - Minty, 116
  - Schur, 209, 216
  - Stack, 453
  - Sugiyama, 211
  - Viterbi, 73, 116, 120, 124, 158, 201, 399, 449
- amplitude modulation, 71
- aperiodic autocorrelation function, 381
- authentication, 273
- Barker sequence, 326, 381
- base station, 129, 423
- baseline wander, 173
- BCH code, 43, 72, 141
- Berlekamp, E., 25
- Berlekamp-Massey algorithm, 35, 72, 209, 211, 288, 307
- Berlekamp-Welch algorithm, 217
- binary entropy function, 451
- binary multiplying channel, 56
- binary sequence, 381
- binary symmetric channel, 157, 251, 259
- binomial coefficient, 142
- biorthogonal code, 72
- birthday paradox, 222
- bit error rate, 72
- Blackburn, S., 35
- Blahut, R. E., 43
- Blahut's theorem, 49, 141, 149
- Blake, I. F., 49
- block code, 56, 116, 201
- blowing up lemma, 55
- bound
  - Elias, 400
  - Hamming, 251
  - Heller, 400
  - Levenshtein, 110
  - Odenwalder, 400
  - Plotkin, 400
  - random-coding, 251
  - Singleton, 95
  - Varshamov-Gilbert, 204, 251, 255, 450
- bounded distance decoding, 391, 392
- broadcast channel, 271, 353, 409
- broadcast protocol, 13
- burst error correcting, 252
- capacity, 69, 82, 316
- capacity region, 50, 410
- Carter, G., 35
- cascading, 155
- catastrophic error propagation, 399
- Cayley-Hamilton theorem, 368

CDMA system, 134  
 channel  
     adder, 49  
     broadcast, 271, 353, 409  
     collision, 353, 419  
     interference, 409  
     multiaccess, 49, 81, 315, 409  
 channel coding theorem, 69  
 charge constrained convolutional code, 173  
 cipher  
     IDEA, 227  
     PES, 227  
     Rip Van Winkle, 179  
 circuit theory, 216  
 code  
     BCH, 43, 72, 141, 209  
     concatenated, 206  
     constacyclic, 93  
     convolutional, 56, 115, 187, 201, 399, 449  
     Golay, 43  
     group, 297, 300  
     Kerdock, 391  
     negacyclic, 93  
     Nordstrom-Robinson, 391  
     Reed-Muller, 72, 149, 162, 391  
     Reed-Solomon, 43, 74, 83, 141, 209, 259  
     runlength-limited, 259  
     Ungerboeck, 74  
     Wyner-Ash, 264  
 code division multiple access, 357, 423  
 Codex Corporation, 115  
 coherent detection, 447  
 Cohn, D. L., 59  
 collision channel, 81, 353, 419  
 collision entropy, 277  
 comma-free coding, 437  
 communication network, 13  
 complexity, 221  
 computational complexity, 223, 391  
 concatenation, 74, 109, 155, 162, 206, 259  
 conjugacy constraint, 44, 145  
 connection polynomial, 43  
 constacyclic code, 93  
 constant envelope signaling, 71  
 constraint length, 72, 73, 187  
 continuous-phase frequency shift keying, 333  
 convolution lemma, 143  
 convolution property, 141  
 convolutional code, 56, 115, 187, 201, 399, 449  
 Coppersmith's algorithm, 223  
 Cooley-Tukey algorithm, 144  
 Costello, D. J., Jr., 69  
 covering radius, 450  
 Coxeter group, 297  
 crosstalk, 409  
 cryptanalyst, 97  
 cryptogram, 97  
 cryptography, 1, 96, 151, 180, 221, 227, 258, 271, 287  
 cryptology, 35  
 Csibi, S., 81  
 cutoff rate, 73, 315, 316, 454  
 cyclotomic set, 145  
 da Rocha, V. C., Jr., 93  
 data compression, 431  
 de Bruijn sequence, 39  
 defect, 193  
 demultiplexer, 263  
 Deng, R. H., 173  
 derivative, 228  
 DES, 227  
 Diaconis' problem, 26  
 differential, 231  
 differential cryptanalysis, 227  
 differential minimum-shift-keying, 333  
 Diffie-Hellman key exchange, 1  
 dihedral group, 298  
 discrete logarithm problem, 223  
 disk access time, 60

- disk drive, 60  
 diversity modulation, 333  
 divide and conquer, 294  
 Doppler shift, 131  
 Drolet, G., 381  
 dual lattice, 124  
  
 Elias bound, 400  
 Elias, P., 101  
 elliptic curve, 7  
 encoding matrix, 187  
 entropy, 293, 411, 432  
 erasure channel, 84, 89  
 Ericson, T., 109  
 Erlang capacity, 423  
 error-correction, 263  
 error detection, 263  
 error locator, 451  
 Euclidean algorithm, 211  
 Euclidean norm, 123  
  
 fading, 423  
 fading channel, 130, 155, 235  
 Fano algorithm, 73  
 Fano's lemma, 274  
 fast algorithm, 209  
 feedback information, 353  
 feedback polynomial, 222  
 finite Fourier transform, 142  
 finite geometry, 101  
 finite reflection group, 297, 298  
 Forney, G. D., Jr., 115  
 Frechet space, 309  
 free Hamming distance, 399  
 frequency hopping, 81, 137  
  
 Galileo mission, 73  
 Gallager, R. G., 129  
 Galois field, 26  
 Gauss sum, 326  
 Gaussian channel, 236  
 Gaussian elimination, 213  
 generalized concatenated code, 394  
 generalized constraint length, 187  
 generator matrix, 189  
  
 Golay code, 43  
 Golay merit factor, 325  
 Gollmann, D., 35  
 Good-Thomas algorithm, 144  
 Gossett lattice, 123  
 gradient algorithm, 365  
 Green Machine, 72  
 group  
 Coxeter, 297  
 dihedral, 298  
 finite reflection, 297  
 isometry, 297  
 group code, 300  
 Günther, C. G., 141  
 Gyorfi, L., 81  
  
 Hadamard transform, 391  
 Hagenauer, J., 155  
 Hamming bound, 251, 258  
 Hamming distance, 236  
 Han-Kobayashi region, 411  
 Hankel map, 308  
 Hankel matrix, 214  
 Hankel matrix factorization, 209  
 Haroutunian, H. S., 13  
 hash function, 252, 258, 278  
 Hasse derivative, 94, 146  
 Heinrich Hertz Institut, 263  
 Heller bound, 400  
 Herro, M. A., 173  
 hexagonal lattice, 109  
 Hoeffding's inequality, 85  
  
 IDEA cipher, 227  
 Information Theory Transactions, 263  
 Ingemarsson, I., 179  
 insecure channel, 273  
 integral lattice, 124  
 INTELSAT, 72  
 interference channel, 409  
 interleaving, 155  
 International Symposium on Information Theory, 263  
 intraset distance, 237

- invariant-factor theorem, 187
- inverse filter, 357
- inverse Fourier transform, 144, 150
- invertible sequence, 323
- Iwahori theorem, 298, 304
- Jet Propulsion Laboratory, 73
- Johannesson, R., 187
- Justesen, J., 201
- Kailath, T., 209
- Kalman filter, 158, 216
- Kerdock code, 391
- key equation, 210
- Key's theorem, 149
- Khachatrian, L. H., 13
- Krawtchouk polynomial, 53
- Kronecker indices, 375
- Kuhn, G. J., 221
- Lai, Xuejia, 227
- Lanczos algorithm, 213
- Lanczos recursions, 209
- lattice, 123
  - dual, 124
  - Gosset, 123
  - integral, 124
- leaf entropy, 350
- Legendre sequence, 325, 327
- Lempel-Ziv algorithm, 431
- Levenshtein bound, 110
- Levinson algorithm, 209, 215
- Li, Yuan Xing, 173
- likelihood ratio, 155
- Lin, S., 235
- linear complexity, 35, 43, 288, 290
- linear complexity profile, 36
- linear complexity property, 44
- linear feedback shift register, 35, 43, 221, 287
- linear system, 367
- linear system theory, 123
- Loeliger, H. A., 251
- log-likelihood ratio, 117, 156
- Lucas' lemma, 142, 150
- magnetic storage, 173
- Markov cipher, 227
- Massey, J. L., 343
- Massey-Omura lock, 1
- Massey-Omura multiplier, 3
- Massey's theorem, 43
- Matt, H. J., 263
- Maurer, U. M., 271
- maximum distance separable code, 95
- maximum likelihood decoding, 72, 160, 316, 334, 391, 450
- maximum likelihood estimation, 324
- Meier, W., 287
- Milne-Thomson theorem, 147
- minimal polynomial, 368
- minimum distance, 236
- minimum Hamming distance, 95
- minimum-shift-keying, 333
- Minkowski-Hlawka theorem, 259
- Minty algorithm 116, 126
- Mittelholzer, T., 297
- Mitter, S. K., 307
- module homomorphism, 308
- module theory, 307
- multiaccess coding theorem, 135
- multiaccess communication, 49, 81, 129, 252, 258, 315, 357, 409
- multilevel code, 75, 235, 394
- multipath, 129, 135, 323, 357
- multiplexer, 263
- multiuser communication, 135, 357, 409
- Murphy, S., 35
- Nakagami distribution, 132
- Narayan, P., 315
- NASA, 75
- near-far resistance, 135
- negacyclic code, 93
- network, 423
- noise enhancement factor, 358
- nonlinear code, 76
- Noordwijk, 263

- Nordstrom-Robinson code, 391  
 normal basis, 1  
 NRZI, 174  
 Nyquist signaling, 70  
 observability, 307  
 Odenwalder bound, 400  
 Odlyzko, A. M., 323  
 offset quadrature PSK, 333  
 one-time pad, 272  
 optical communication, 315  
 optical fiber, 173  
 optical storage, 173  
 packet, 82  
 Pareto exponent, 456  
 Parseval theorem, 141, 146  
 partial realization theory, 307  
 Paterson, K., 35  
 path multiplicity, 239  
 Penzhorn, W. T., 221  
 Perez, L. C., 69  
 permutation modulation, 298  
 PES cipher, 227  
 phase-shift-keying, 71, 442  
 Pioneer, 73, 115  
 Piper, F., 35  
 plaintext attack, 97, 227  
 Plotkin bound, 400  
 point to point communication, 130  
 Poisson process, 82  
 power limited signaling, 71  
 precoding, 447  
 privacy amplification, 276  
 probability of decoding failure, 204  
 processing gain, 358  
 product distance, 236  
 protocol, 13, 82, 273, 423  
 pseudorandom generator, 287  
 public channel, 273, 276  
 public-key cryptography, 1, 272  
 quadrature amplitude modulation, 71, 75  
 quadrature phase-shift-keying, 71  
 Qualcomm, 129  
 RAID system, 60  
 Rajpal, S., 235  
 rake receiver, 133  
 random-access, 353  
 random-coding bound, 251, 252, 256  
 rational transfer function, 188  
 Rayleigh channel, 163, 236  
 Rayleigh distribution, 132  
 reachability, 307  
 Riccati equation, 216  
 Rician distribution, 132  
 Rieger bound, 263  
 Rimoldi, B., 333  
 ring code, 76  
 ring of integers, 439  
 ring of polynomials, 439  
 Rip Van Winkle cipher, 179  
 root system, 298  
 rooted tree, 345  
 rotational invariance, 76, 447  
 rotational latency, 60  
 Rueppel, R. A., 343  
 runlength, 174  
 runlength-limited code, 259  
 running digital sum, 174  
 Ruprecht, J., 357  
 Ruprecht merit factor, 324  
 Sain, M. K., 367  
 satellite communication, 71  
 scattering theory, 216  
 Schur algorithm, 209, 216  
 secrecy capacity, 271  
 secret key rate, 271, 275  
 Seguin, G., 381  
 sequence, 37  
     Barker, 326, 381  
     binary, 381  
     de Bruijn, 39  
     invertible, 323  
     Legendre, 325, 327

- sequential decoding, 73, 115, 317, 449, 453  
 set-partitioning, 401  
 Shannon, C. E., 69  
 Shannon entropy, 277  
 Shannon limit, 155  
 Shannon packing, 252, 257  
 shrinking generator, 287  
 signal constellation, 237  
 signal set partitioning, 235, 237  
 signal-to-noise ratio, 70  
 simple root, 299  
 simplex conjecture, 319  
 simplex signal set, 302, 315, 318  
 Singleton bound, 95  
 Snyder, D. L., 315  
 soft decision, 72  
 soft decision decoding, 155, 239  
 soft output, 157  
 space communication, 71, 115  
 span, 116  
 spectral efficiency, 70  
 spectral zero, 44  
 sphere packing, 252, 257  
 spherical code, 109  
 spread-spectrum communication, 180, 315  
 squared Euclidean distance, 236  
 Stack algorithm, 453  
 Staffelbach, O., 287  
 stereo audio signal, 270  
 Stevenson, R. L., 59  
 stream cipher, 147, 221, 287  
 Sun, Feng-Wen, 391  
 symmetric group, 26  
 synchronization, 174, 180, 263, 423  
 syndrome, 202, 258, 264  
 systems theory, 307  
 telephone channel, 76  
 theorem
  - Blahut, 141, 149
  - Cayley-Hamilton, 368
  - Iwahori, 298, 304
 Key, 149  
 Lucas, 142  
 Massey, 43  
 Milne-Thomson, 147  
 Minkowski-Hlawka, 259  
 multiaccess CDMA, 135  
 Parseval, 141  
 threshold decoding, 72, 115, 160, 399  
 time division multiple access, 357  
 time hopping, 84  
 Toeplitz equation, 215  
 torsion module, 368  
 transmission-line theory, 216  
 tree code, 116  
 trellis, 123, 201, 399  
 trellis code, 74, 235, 399, 439  
 turbo-code, 74  
 two-way channel, 409  
 Ungerboeck code, 74, 446  
 Ungerboeck, G., 399  
 uniquely decodable, 51  
 unit memory convolutional code, 202  
 van der Meulen, E. C., 409  
 van Tilborg, H. C. A., 391  
 Varshamov-Gilbert bound, 54, 204, 251, 255, 450  
 Viterbi algorithm, 73, 116, 120, 124, 155, 158, 201, 239, 334, 399, 443, 449  
 Viterbi, A. J., 423  
 Voronoi region, 392  
 Voyager, 74, 75  
 Wagner decoding, 394  
 Wan, Zhe-Xian, 187  
 Wedderburn numbers, 375  
 weight, 44  
 weight retaining property, 93  
 Wild, P., 35  
 wireless communication, 130  
 Wyner, A. D., 431  
 Wyner-Ash code, 264  
 Zech's logarithm, 222

zero-error capacity, 419

Zetterberg, L. H., 439

Zigangirov, K. Sh., 449

Zinoviev, V., 109

Ziv, J., 431

# PUBLICATIONS BY

James L. Massey

## I Books

- \*[1] J. L. Massey, *Threshold Decoding*. Cambridge, MA: M.I.T. Press, 1963.

\*Awarded 1963 Paper Award by the IEEE Group on Information Theory.

## II Books: Chapters in Contributed Volumes

- [1] J. L. Massey, "Information, Machines, and Men," in *Philosophy and Cybernetics* (Eds. F. J. Crosson and K. M. Sayre). Notre Dame, IN: Notre Dame Press, 1967, pp. 37-69.
- [2] J. L. Massey, "Advances in Threshold Decoding," in *Advances in Communication Systems*, Vol. III (Ed. A. Ralarishnan). New York: Academic Press, 1968, pp. 91-115.
- [3] J. L. Massey, "Applications of Automata Theory in Coding," in *Applied Automata Theory* (Ed. J. Tou). New York: Academic Press, 1968, pp. 125-146.
- [4] J. L. Massey, "Error Correcting Codes," in *Encyclopaedia of Linguistics, Information, and Control* (Ed. A. R. Meetham). London: Pergamon, 1969, pp. 173-176.
- [5] J. L. Massey and O. N. Garcia, "Error-Correcting Codes in Computer Arithmetic," in *Advances in Information Systems Science*, Vol.4 (Ed. J. Tou). New York: Plenum Press, 1972, pp. 273-326.
- [6] J. L. Massey, "Coding Theory," in *Handbook of Applicable Mathematics*, Vol.5, *Combinatorics and Geometry* (Eds. W. Lederman and S. Vajda). Chichester and New York: Wiley, 1985, pp. 623-676.

## III Articles in Conference Proceedings Published in Book Form

- [1] J. L. Massey, "Some Algebraic and distance Properties of Convolutional Codes," in *Error Correcting Codes* (Ed. H. B. Mann). New York: Wiley, 1968, pp. 89-109.
- [2] J. L. Massey, "Methods of Alleviation of Ionospheric Scintillation Effects on Digital Communications," in *Communication Satellite Developments: Technology, Progress in Astronautics and Aeronautics*, Vol. 42 (Ed. W. G. Schmidt and G. E. LaVean). Cambridge, MA: M.I.T. Press, 1976, pp. 279-287.

- [3] J. L. Massey, "The Codeword and Syndrome Methods for Data Compression with Error-Correcting Codes," in *New Directions in Signal Processing in Communication and Control*, (Ed. J. K. Skwirzynski), NATO Advanced Study Institutes Series E12. Leyden, The Netherlands: Noordhoff, 1975, pp. 3-13.
- [4] J. L. Massey, "Markov Information Sources," in *New Directions in Signal Processing in Communication and Control*, (Ed. J. K. Skwirzynski), NATO Advanced Study Institutes Series E25. Leyden, The Netherlands: Noordhoff, 1975, pp. 15-26.
- [5] J. L. Massey, "Error Bounds for Tree Codes, Trellis Codes, and Convolutional Codes with Encoding and Decoding Procedures," in *Coding and Complexity* (Ed. G. Longo with Preface by J. L. Massey), CISM Courses and Lectures No. 216. Wien and New York: Springer-Verlag, 1976, pp.1-57.
- [6] J. L. Massey, "Joint Source and Channel Coding," in *Communication Systems and Random Process Theory* (Ed. J. K. Skwirzynski), NATO Advanced Studies Institutes Series E25. The Netherlands: Noordhoff, 1978, pp. 279-293.
- [7] J. L. Massey, "Collision-Resolution Algorithms and Random-Access Communications," in *Multi-User Communication Systems* (Ed. G. Longo), CISM Courses and Lectures No. 265. Wien and New York: Springer, 1981, pp. 73-137.
- [8] P. Schoebi and J. L. Massey, "Fast Authentication in a Trapdoor-Knapsack Public Key Cryptosystem," in *Cryptography* (Ed. T. Beth), Lecture Notes in Computer Science, No. 149. New York: Springer, 1983, pp. 289-294.
- [9] J. L. Massey, "An Information-Theoretic Approach to Algorithms," in *The Impact of Processing Techniques in Communications* (Ed. J. K. Skwirzynski), NATO Advanced Study Institutes Series E91. Dordrecht, The Netherlands: Nijhoff, 1985, pp. 3-20.
- [10] J. L. Massey and R. A. Rueppel, "Linear Ciphers and Random Sequence Generators with Multiple Clocks," in *Advances in Cryptology-Eurocrypt '84* (Eds T. Beth, N. Cot and I. Ingemarsson), Lecture Notes in Computer Science, No. 209. Heidelberg and New York: Springer 1985, pp. 74-87.
- \*[11] J. L. Massey, "Cryptography - A Selective Survey," in *Digital Communications* (Eds. E. Biglieri and G. Prati). Amersterdam: North-Holand, 1986, pp. 3-21.
- \*Reprinted as Invited Paper in *Alta Frequenza*, Vol. 55, No.1, pp. 4-11, Jan. - Feb. 1986.
- [12] J. L. Massey, U. M. Maurer and M.-Z. Wang, "Non-Expanding, Key-Minimal, Robustly Perfect, Linear and Bilinear Ciphers," in *Advances in Cryptology - Eurocrypt '87* (Eds. D. Chaum and W. L. Price), Lecture Notes in Computer Science, No. 304. Heidelberg and New York: Springer, 1988, pp. 237-247.
- [13] J. L. Massey, "Channel Models for Random-Access Systems," in *Performance Limits in Communication Theory and Practice* (Ed. J.K. Skwirzynski), NATO Advances

Studies Institutes Series E142. Dordrecht, The Netherlands: Kluwer Academic, 1988, pp. 391-402.

- \*[14] J. L. Massey, "Some New Approaches to Random-Access Communications," in *Performance' 87* (Eds. P.-J. Courtois and G. Latouche), Proc. 12th IFIP WG 7.3 Int. Symp. in *Computer Performance Modelling*. Amsterdam and New York: North Holland 1988, pp. 551-569.
- \*Reprinted in *Multiple Access Communications: Foundations for Emerging Technologies* (Ed. N. Abramson). New York: IEEE Press, 1993, pp. 354-378 .
- [15] J. L. Massey and T. Schaub, "Linear Complexity in Coding Theory," in *Coding Theory and Applications* (Eds G. Cohen and Ph. Godlewski), Lecture Notes in Computer Science, No. 311. Heidelberg and New York: Springer, 1988, pp. 19-32.
- [16] H. N. Jendal, Y. J. B. Kuhn and J. L. Massey, "An Information-Theoretic Approach to Homophonic Substitution," in *Advances in Cryptology-Eurocrypt' 89* (Eds. J.-J. Quisquater and J. Vandewalle), Lecture Notes in Computer Science, No. 434. Heidelberg and New York: Springer, 1990, pp. 382-394.
- [17] U. M. Maurer and J. L. Massey, "Perfect Local Randomness in Pseudo-Random Sequences," in *Advances in Cryptology-Crypto'89* (Ed. G. Brassard), Lecture Notes in Computer Science, No. 435. Heidelberg and New York: Springer, 1990, pp. 100-112.
- [18] J. L. Massey, "The Relevance of Information Theory to Modern Cryptography," in *Communications, Control and Signal Processing* (Ed. E. Arikan), Proc. of 1990 Bilkent Int. Conference. Amsterdam: Elsevier, 1990, pp. 176-182.
- [19] X. Lai and J. L. Massey, "A Proposal for a New Block Encryption Standard," in *Advances in Cryptology- Eurocrypt'90* (Ed. I.B. Damgard), Lecture Notes in Computer Science No. 473. Heidelberg and New York: Springer, 1991, pp. 389-404.
- [20] X. Lai, J. L. Massey and S. Murphy, "Markov Ciphers and Differential Cryptanalysis," in *Advances in Cryptology-Eurocrypt'91* (Ed. D. W. Davies), Lecture Notes in Computer Science No. 547. Heidelberg and New York: Springer 1991, pp. 17-38. [A preliminary version with the same title by X. Lai and J. L. Massey appears in *AGEN Communications*, Nr. 53, pp. 25-29, June 1991.]
- [21] J. L. Massey, "Contemporary Cryptology: An Introduction," in *Contemporary Cryptology-The Science of Information Integrity* (Ed. G. J. Simmons). New York: IEEE Press, 1992, pp. 1-39.
- [22] J. L. Massey, "Deep-Space Communications and Coding: A Marriage Made in Heaven," in *Advanced Methods for Satellite and Deep Space Communications* (Ed. J. Hagenauer), Lecture Notes in Control and Information Sciences No. 182. Heidelberg and New York: Springer 1992, pp. 1-17.

- [23] X. Lai and J. L. Massey, "Hash Functions Based on Block Ciphers," in *Advances in Cryptology-Eurocrypt'92*. (Ed. R. A. Rueppel), Lecture Notes in Computer Science, No. 658. Heidelberg and New York: Springer, 1993, pp. 55-70.
- [24] J. L. Massey and T. Mittelholzer, "Welch's Bound and Sequence Sets for Code-Divison Multiple-Access Systems," in *Sequences II: Methods in Communication, Security and Computer Sciences* (Eds. R. Capocelli, A. De Santis and U. Vaccaro). Heidelberg and New York: Springer, 1993, pp. 63-78.
- [25] C. P. Waldvogel and J. L. Massey, "The Probability Distribution of the Diffie-Hellman Key," in *Advances in Cryptology-Auscrypt'92* (Eds. J. Seberry and Y. Zheng), Lecture Notes in Computer Science No. 718. Heidelberg and New York: Springer 1993, pp. 492-504.
- [26] V. C. da Rocha and J. L. Massey, "A New Approach to the Design of Codes for the Binary Adder Channel," in *Cryptography and Coding III* (Ed. M. J. Ganley), IMA Conf. Series, New Series No. 45. Oxford: Clarendon Press, 1993, pp. 179-185.
- [27] B. Blakley, G. R. Blakley, A. H. Chan and J. L. Massey, "Threshold Schemes with Disenrollment," in *Advances in Cryptology Crypto'92* (Ed. E. F. Brickell), Lecture Notes in Computer Science No. 740. New York: Springer, 1993, pp. 540-548.
- [28] J. L. Massey, "SAFER K-64: A Byte-Oriented Block-Ciphering Algorithm," to appear in *Proc. Cambridge Algorithms Workshop*, Cambridge, England, 1994.

#### IV Articles in Refereed Journals

- [1] W. W. Peterson and J. L. Massey, "Coding Theory," *IEEE Trans. on Info. Theory*, Vol. IT-9, pp. 223-229, Oct. 1963.
- [2] J. L. Massey and R. Liu, "Application of Lyapunov's Direct Method to the Error-Propagation Problem in Convolutional Codes," *IEEE Trans. on Info. Theory*, Vol. IT-10, pp. 248-250, July 1964.
- [3] J. L. Massey, "Reversible Codes," *Information & Control*, Vol. 7, pp. 369-380, Sept. 1964.
- [4] J. L. Massey and R. Liu, "Equivalence of Nonlinear-Feedback Shift-Registers," *IEEE Trans. on Info. Theory*, Vol. IT-10, pp. 378-379, Oct. 1964.
- [5] J. L. Massey, "Survey of Residue Coding for Arithmetic Errors," *ICC Bulletin*, Vol. 3, pp. 195-209, Oct. 1964.
- [6] J. L. Massey, "Implementation of Burst-Correcting Convolutional Codes," *IEEE Trans. on Info. Theory*, Vol. IT-11, pp. 416-422, July 1965.
- [7] J. L. Massey, "Step-by-step Decoding of the Bose-Chaudhuri-Hocquenghen Codes," *IEEE Trans. on Info. Th.*, Vol. IT-11, pp. 580-585, Oct. 1965.

- [8] J. L. Massey, "Uniform Codes," *IEEE Trans. on Info. Th.*, Vol. IT-12, pp. 132-134, April 1966.
  - [9] J. L. Massey, "Note on Finite-Memory Sequential Machines," *IEEE Trans. on Elec. Comp.*, Vol. EC-15, pp. 658-659, Aug. 1966.
  - [10] J. L. Massey and L. V. Auth, "Impedance Magnitude Measurements from a Resistive Bridge," *IEEE Trans. on Education*, Vol. E-10, pp. 50-51, March 1967.
  - \*[11] J. L. Massey, "Shift-Register Synthesis and BCH Decoding," *IEEE Trans. on Info. Th.*, Vol. IT-15, pp. 122-127, Jan. 1969.
- \*Reprinted in *Algebraic Coding Theory: History and Development* (Ed. I. F. Blake), Benchmark Papers in Electrical Engineering and Computer Science. Stroudsburg, PA: Dowden, Hutchison and Ross, 1973, pp. 233-238.
- [12] J. L. Massey and M. K. Sain, "Codes, Automata and Continuous Systems: Explicit Interconnections," *IEEE Trans. on Auto. Control*, Vol. AC-12, pp. 644-50, December 1967. [Also appears in *Proc. NEC*, Vol. 23, pp. 33-38, 1967.]
  - \*\*[13] J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits," *IEEE Trans. on Computers*, Vol. C-17, pp. 330-337, April 1968. [See also J. L. Massey and M. K. Sain, "Postscript to Inverses of Linear Sequential Circuits," *IEEE Trans. on Computers*, Vol. C-17, pp. 1177, Dec. 1968.]
- \*\* Reprinted in *Key Papers in the Development of Coding Theory* (Ed. E. R. Berlekamp). New York: IEEE Press, 1974, pp. 205-212.
- [14] M. K. Sain and J. L. Massey, "Invertibility of Linear, Time-Invariant Dynamical Systems," *IEEE Trans. on Auto. Cont.*, Vol. AC-14, pp. 141-149, April 1969.
  - [15] J. L. Massey, "Variable-Length Codes and the Fano Metric," *IEEE Trans. on Info. Th.*, Vol. IT-18, pp. 196-198, Jan. 1972.
  - [16] J. L. Massey and D. J. Costello, Jr., "Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications," *IEEE Trans. Com. Tech.*, Vol. Com-19, pp. 806-813, Oct. 1971.
  - [17] J. L. Massey, "Optimum Frame Synchronization," *IEEE Trans. on Comm.*, Vol. COM-20, pp. 115-119, April 1972.
  - [18] J. L. Massey, M. K. Sain and J. M. Geist, "Certain Infinite Markov Chains and Sequential Decoding," *Discrete Math.*, Vol. 3, pp. 163-175, Sept. 1972.
  - [19] J. L. Massey, D. J. Costello, Jr., and J. Justesen, "Polynomial Weights and Code Constructions," *IEEE Trans. Info. Th.*, Vol. IT-19, pp. 101-110, Jan. 1973.
  - [20] J. L. Massey, "On the Fractional Weight of Distinct Binary n-Tuples," *IEEE Trans. Info. Th.*, Vol. IT-20, pp. 131, Jan. 1974

- [21] H. J. Matt and J. L. Massey, "Determining the Burst-Correcting Limit of Cyclic Codes," *IEEE Trans. Info. Th.*, Vol. IT-26, pp. 289-297, May 1980.
  - [22] J. L. Massey, "Capacity, Cut-Off Rate, and Coding for a Direct-Detection Optical Channel," *IEEE Trans. Comm.*, Vol Com-29, pp. 1615-1612, Nov. 1981.
  - [23] J. L. Massey, "What is a Bit of Information?," *Scientia Electrica*, Vol. 28, No. 1, pp. 1-11, 1982. Reprinted in German Translation as "Was ist ein Bit Information?," *Frequenz*, Band 37, S. 110-115, Mai 1983.
  - [24] M. Schlatter and J. L. Massey, "Capacity of Interconnected Ring Communication systems with Unique Loop-free Routing," *IEEE Trans. on Info. Th.*, Vol. IT-29, pp. 774-778, Sept. 1983.
  - \*[25] J. L. Massey and P. Mathys, "The Collision Channel without Feedback," *IEEE Trans. Info. Th.*, Vol. IT-31, pp. 192-204, March 1985.
- \*Awarded the IEEE 1986 W. R. G. Baker Award.
- [26] J. L. Massey, "Standardisierung Kryptographischer Dienste," *Bulletin des Schweizerischen Elektrotechnischen Vereins*, Vol. 77, No. 7, pp. 367-372, 12 April 1986.
  - [27] Z. Zhang, T. Berger and J. L. Massey, "Some Families of Zero-Error Block Codes for the 2-User Binary Adder Channel with Feedback," *IEEE Trans Info. Th.*, Vol. IT-33, pp. 613-619, Sept. 1987.
  - [28] J. L. Massey, "Probabilistic Encipherment," *E. & M. (Elektrotechnik & Maschinenbau, Austria)*, Vol. 104, No. 12, pp. 561-562, Dec. 1987.
  - [29] J. L. Massey, "An Introduction to Contemporary Cryptology," *Proc. IEEE*, Vol 76, pp. 533-549, May 1988.
  - [30] W. Hirt and J. L. Massey, "Capacity of the Discrete-Time Gaussian Channel with Intersymbol Interference," *IEEE Trans. Info. Th.*, Vol. IT-34, pp. 380-388, May 1988.
  - [31] G. Z. Xiao and J. L. Massey, "A Spectral Characterization of Correlation-Immune Combining Functions," *IEEE Trans. Info. Th.*, Vol. IT-34, pp. 569-571, May 1988.
  - [32] U. M. Maurer and J. L. Massey, "Local Randomness in Pseudorandom Sequences," *J. of Cryptology*, Vol. 4, No. 2, pp. 135-149, 1991.
  - [33] G. Gastagnoli, J. L. Massey, P. Schoeller and N. von Seeman, "On Repeated-Root Cyclic Codes," *IEEE Trans. Info. Th.*, Vol. IT-37, pp. 337-342, March 1991.
  - [34] U. M. Maurer and J. L. Massey, "Cascade Ciphers: The Importance of Being First," *J. of Cryptology*, Vol. 6, No. 1, pp. 55-61, 1993
  - [35] N. Q. A, L. Gyorfi and J. L. Massey, "Constructions of Binary Constant-Weight Cyclic Codes and Cyclically Permutable Codes," *IEEE Trans. Info. Th.*, Vol. IT-38, pp. 940-949, May 1992.

- [36] J. L. Massey, "Linear Codes with Complementary Duals," *Discrete Math.*, Vol. 106/107, pp. 337-342, 1992. [Also appears in *A Collection of Contributions in Honor of Jack van Lint* (Eds. P. J. Cameron and H. C. A. van Tilborg), Topics in Discrete Math. 7. Amsterdam: Elsevier 1992, pp. 337-342.]
- [37] F. D. Neeser and J. L. Massey, "Proper Complex Random Processes with Applications to Information Theory," *IEEE Trans. Info. Th.*, Vol. IT-39, pp. 1293-1302, July 1993.
- [38] M. Rupf and J. L. Massey, "Optimum Sequence Multisets for Synchronous Code-Division Multiple-Access Channels," to appear in *IEEE Trans. Info. Th.*, 1994.
- [39] X. Yang and J. L. Massey, "The Condition for a Cyclic Code to Have a Complementary Dual," to appear in *Discrete Math.*, 1994.
- [40] M. Rupf , F. Tarkoy and J. L. Massey, "User-Separating Demodulation for Code-Division Multiple-Access Systems," to appear in *IEEE J. Selected Areas in Comm.*, 1994.
- [41] D. R. Stinson and J. L. Massey, "An Infinite Class of Counterexamples to a Conjecture Concerning Non-Linear Resilient Functions," to appear in *J. of Cryptology*, 1994.

## V Articles in Conference Proceedings and Non-Refereed Journals

- \*[1] J. L. Massey, "Error-Correcting Codes Applied to Computer Technology," in *Proc. Nat. Elec. Conf.*, Vol. 19, Oct. 1963, pp. 142-147.  
\*Awarded the Best Tutorial Paper Award at the 1963 National Electronics Conference.
- [2] J. L. Massey and R. Liu, "Monotone Feedback Shift-Registers," in *Proc. of 2nd Annual Allerton Conference on Circuit & System Theory*, 1964, pp. 864-874.
- [3] J. L. Massey and R. Liu, "A New Diagram for Feedbck Shift-Registers," in *Proc. of 3rd Annual Allerton Conf. on Circuit & Systems Theory*, 1965 pp. 73-81.
- [4] J. L. Massey and M. K. Sain, "Inverse Problems in Coding, Automata and Continuous Systems," in *IEEE Conf. Record of 8th Symposium on Switching and Automata Theory*, Oct. 1967, pp. 226-232.
- [5] J. L. Massey, "Catastrophic Error-Propagation in Convolutional Codes," in *Proc. 11th Midwest Circuit Th. Symp.*, 1968, pp. 583-587.
- [6] J. L. Massey and M. K. Sain, "Trunk and Tree Searching Properties of the Fano Sequential Decoding Algorithm," in *Proc. of 6th Annual Allerton Conf. on Circuit and System Theory*, 1968, pp. 153-160.
- [7] J. L. Massey and M. K. Sain, "A Modified Inverse for Linear Dynamical Systems," in *Proc. IEEE 8th Adaptive Processes Symp.*, 1969, pp. 5a1-5a3.

- [8] J. L. Massey and M. K. Sain, "Derivative Controllability," in *Proc. 4th Princeton Conf. on Info. Sci. and Systems*, 1970, p. 189.
- [9] J. L. Massey, "Can Coding Beat the System?," in *IEEE Int. Convention Digest*, 1970, pp. 354-355.
- [10] J. J. Uhran, Jr., and J. L. Massey, "Analysis of Satellite Communications in a Multi-path Environment," in *ICC '70 Conf. Record*, 1970, pp. 10-20 to 10-24.
- [11] J. L. Massey, "Coding and Modulation in Digital Communications," in *Proc. Int. Zurich Seminar on Digital Comm.*, 1974, pp. E2(1)-E2(4).
- [12] J. L. Massey and J. J. Uhran, Jr., "Sub-Baud Coding," in *Proc. 13th Annual Allerton Conf. on Circuit and System Theory*, 1975, pp. 539-547.
- [13] J. L. Massey, "The Statistical Significance of Error Probability as Determined from Decoding Simulations for Long Codes," in *Proc. 7th Annual Pittsburgh Conf. on Modeling and Simulation*, April 26-27, 1976, pp. 63-64.
- [14] J. L. Massey, "The Use of Redundant Packets in Slotted Aloha Type Random Access Schemes," in *Proc. 1978 Conf. on Info. Sciences and Systems*, March 29-31, 1978, pp. 78-81.
- [15] J. L. Massey, "Foundations and Methods of Channel Encoding," in *Proc. Int. Conf. on Info. and System Theory in Digital Comm.*, NTG-Fachberichte, Band 65, 1978, pp. 148-157.
- [16] J. L. Massey, "Coding Techniques for Digital Data Networks," in *Proc. Int. Conf. on Info. and System Theory in Digital Comm.*, NTG-Fachberichte, Band 65, 1978, pp. 307-315.
- [17] J. L. Massey, "A Generalized Formulation of Minimum Shift Keying Modulation," in *ICC '80 Conf. Record*, Vol. 2, 1980, pp. 26.5.1-26.5.4.
- [18] J. L. Massey, "Logarithms in Finite Cyclic Groups—Cryptographic Issues," in *Proc. 4th Benelux Symp. on Info. Theory*, 1983, pp. 17-25.
- [19] J. L. Massey, "A Simplified Treatment of Wyner's Wire-Tap Channel," in *Proc. 21st Allerton Conf. on Comm., Control and Computing*, 1983, pp. 268-276.
- [20] J. L. Massey, A. Gubser, A. Fischer and P. Hochstrasser, B. Huber and R. Sutter, "A Self-Synchronizing Digital Scrambler for Cryptographic Protection of Data," in *Proc. Int. Zurich Seminar on Digital Comm.*, 1984, pp. 163-169.
- [21] J. L. Massey, "The How and Why of Channel Coding," in *Proc. Int. Zurich Seminar on Digital Comm.*, 1984, pp. 67-73.
- \*[22] J. L. Massey, "Information Theory, The Copernican System of Communications," *Proc. ICC '84*, Vol. 1, 1984, pp. 159-162.

\*Reprinted in *IEEE Communication Magazine*, Vol. 22, pp. 26-28, Dec. 1984.

- [23] J. L. Massey, "Delayed-Decimation/Square Sequences," *Proc. 2nd Joint Swedish-Soviet Int. Workshop on Info. Th.*, 1985, pp. 118-123.
- [24] Z. Zhang, T. Berger and J. L. Massey, "Some Families of Zero-Error Codes for the 2-User Binary Adder Channel," in *Proc. 23rd Allerton Conf. on Comm., Control and Computing*, 1985.
- \*[25] A. S. Glass and J. L. Massey, "Plastic Cards— How Smart is Secure?," *Landis & Gyr Review*, Vol 32, No. 2, pp. 22-38, 1986.  
 \*\*Reprinted in German version, "Plastikkarten – wie intelligent ist sicher?," in *Schweizerische Technische Zeitschrift*, No. 9, pp. 10-20, No. 10, pp. 36-46, and No. 11, pp. 21-29, May-June, 1986.
- [26] J. L. Massey, "Cryptography and System Theory," in *Proc. 24th Allerton Conf. on Comm., Control and Computing*, 1986, pp. 1-8.
- [27] J. L. Massey, "On the Entropy of Integer-Valued Random Variables," in *Proc. 1988 Beijing Int. Workshop on Info. Th.*, 1988, pp. C1.1-C1.4.,
- [28] X. Lai and J. L. Massey, "Some Connections between Scramblers and Invertible Automata," in *Proc. 1988 Beijing Int. Workshop on Info. Th.*, 1988, pp. DI5.1-DI5.5.
- [29] J. L. Massey, "A Short Introduction to Coding Theory and Practice," in *Proc. Int. Symp. on Signals, Systems and Electronics*, 1989, pp. 629-633.
- [30] J. L. Massey and T. Mittelholzer, "Convolutional Codes over Rings," in *Proc. 4th Joint Swedish-Soviet Int. Workshop on Info. Th.*, 1989, pp. 14-18.
- [31] J. L. Massey and T. Mittelholzer, "Systematicity and Rotational Invariance of Convolutional Codes over Rings," in *Proc. 2nd Int. Workshop on Algebraic and Combinatorial Coding Theory*, Leningrad, 1990, pp. 154-158.
- [32] J. L. Massey, "Causality, Feedback and Directed Information," in *Proc. 1990 Int. Symp. on Info. Th. & its Applications*, 1990, pp. 303-305.
- [33] R. A. Rueppel and J. L. Massey, "The Security of Natel D GSM" [presented at the Berner Technologie Forum on Mobile Communications, Berne, Oct. 24, 1991] *Technische Mitteilungen PTT*, No. 3, pp. 108-112, 1992. (Republished in German as "Die Sicherheit von Natel D GSM," *Technische Mitteilungen PTT*, No. 4, pp. 150-153, 1992.)
- [34] J. L. Massey, "Coding and Modulation for Code-Division Multiple Accessing," in *Proc. Third Int. Workshop on Digital Signal Processing Techniques Applied to Space Communications*, ESTEC, Noordwijk, 1992, pp. 3.1-3.17.
- [35] J. L. Massey, "Some Applications of Source Coding in Cryptography," in *Proc. 3rd Symp. on State and Progress of Research in Cryptography*, Rome, 1993, pp. 143-160.
- [36] J. L. Massey, "Minimal Codewords and Secret Sharing," in *Proc. 6th Joint Swedish-Russian Int. Workshop on Info. Theory*, 1993, pp. 276-279.

## VI Papers Presented at IEEE International Symposia on Information Theory

- [1] J. L. Massey, "Uniform Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1966, p. 19.
- [2] J. L. Massey and M. K. Sain, "Distribution of the Minimum Cumulative Metric for Sequential Decoding," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1969, p. 56.
- [3] J. L. Massey and J. J. Uhran, Jr., "Sub-Baud Coding and Cyclic Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1970, pp. 25-26.
- [4] D. J. Costello, Jr., and J. L. Massey, "Constructing Good Convolutional Codes from Cyclic Block Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1972, p. 44.
- [5] J. L. Massey, "An Error Bound for Random Tree Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1973, pp. D3.1-D3.2.
- [6] J. L. Massey, "An Information-Theoretic Approach to Data-Processing Algorithms," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1974, pp. 35-36.
- [7] J. L. Massey, "All Signal Sets Centered about the Origin Are Optimal at Low Energy-to-Noise Ratios on the AWGN Channel," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1976, pp. 80-81.
- [8] J. L. Massey, "A Class of Maximum Distance Separable Codes over GF(p) Encodable Using Only Addition and Subtraction," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1976, p. 133.
- [9] J. L. Massey, "Shannon's 'Proof' of the Noisy Coding Theorem," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1977, p. 107.
- [10] J. L. Massey, "Convolutional Codes - Theory Lagging Practice," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1979, p. 20.
- [11] J. L. Massey, "Capacity, Cut-Off Rate, and Coding for a Direct-Detection Optical Channel," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1981, p. 58.
- [12] J. L. Massey, "Effect of Channel Error on the Capetanakis and Related Random-Access Algorithms," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1981, p. 65.
- [13] M. Schlatter and J. L. Massey, "Capacity of Interconnected Ring Communication Systems with Unique Loop-Free Routing," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1982, p. 61-62.
- [14] J. L. Massey, "The Capacity of the Collision Channel without Feedback," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1982, p. 101.

- [15] J. L. Massey, "The Entropy of a Rooted Tree with Probabilities," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1983, p. 127.
- [16] N. Q. A, L. Gyorfi, and J. L. Massey, "Some Constructions of Protocol Sequences for Collision Channels and a Class of Optimum Cyclic Constant Weight Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1985, p. 40.
- [17] R. A. Rueppel and J. L. Massey, "The Knapsack as a Nonlinear Function," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1985, p. 46.
- [18] I. Ingemarsson and J. L. Massey, "Synchronization of Truly Random Binary Sequences," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1985, p. 57.
- [19] D. J. Costello, Jr., and J. L. Massey, "A Lower Bound on the Minimum Distance Growth Rate of Fixed Convolutional Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1985, p. 90.
- [20] W. Hirt and J. L. Massey, "On the Mutual Information and Cut-Off Rate of Channels with Intersymbol Interference," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1985, pp. 114-115.
- [21] J. L. Massey and I. Ingamarsson, "The Rip van Winkle Cipher - A Simple and Provably Computationally Secure Cipher with a Finite Key," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1985, p. 146.
- [22] J. L. Massey, N. von Seeman and P. Schoeller, "Hasse Derivatives and Repeated-Root Cyclic Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1986, p. 39.
- [23] T. Schaub and J. L. Massey, "Bounds on the Minimum Distance of Cyclic Codes via Bounds on the Linear Complexity of Periodic Sequences with Known Patterns of Zeros," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1986, p. 71.
- [24] A. Gubser and J. L. Massey, "Node Synchronization of  $R = 1/2$  Binary Convolutional Codes," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1986, p. 127.
- [25] N. Q. A, L. Gyorfi, and J. L. Massey, "Families of Sequences with Optimal Generalized Hamming Correlation Properties," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1986, pp. 136-137.
- \*[26] J. L. Massey, "Towards a Proof of the Simplex Conjecture?," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1988, p. 59.

\*Presented as the Shannon Lecture at this symposium.

- [27] J. Ruprecht and J. L. Massey, "Binary Input Sequences for Maximum-Likelihood Estimation of Multipath Channels," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1990, p. 32.
- [28] U. M. Maurer and J. L. Massey, "Cascade Ciphers: The Importance of Being First," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1990, p. 118.

- [29] J. L. Massey, T. Mittelholzer, T. Riedel and M. Vollenweider, "Ring Convolutional Codes for Phase Modulation," in *Abstracts of Papers, IEEE Int. Symp. on Info. Th.*, 1990, p. 176.
- [30] J. L. Massey, "On Welch's Bound for the Correlation of a Sequence Set," in *Proc. IEEE Int. Symp. on Info. Th.*, 1991, p. 385.
- [31] B. Blakley, G. R. Blakley, A. H. Chan and J. L. Massey, "Threshold Schemes with Disenrollment," in *Proc. IEEE Int. Symp. on Info. Th.*, 1993, p. 229.
- [32] M. Rupf and J. L. Massey, "Optimum Sequence Multisets for Symbol-Synchronous Code-Division Multiple-Access Channels," in *Proc. IEEE Int. Symp. on Info. Th.*, 1993, p. 373.
- [33] J. L. Massey, "Guessing and Exponentiated Entropy," to appear in *Proc. IEEE Int. Symp. on Info. Th.*, 1994.

## VII Published Book Reviews

- [1] J. L. Massey, (*Information Theory* by R. Ash. New York: Wiley, 1965) IEEE Trans. Info. Th., Vol. IT-12, pp. 488-489, Oct. 1966.
- [2] J. L. Massey, (*An Introduction to Error-Correcting Codes* by S. Lin. Englewood Cliffs, N. J.: Prentice-Hall, 1970) IEEE Trans. Info. Th., Vol. IT-17, pp. 768-769, Nov. 1971.
- [3] J. L. Massey, (*Topics in Mathematical System Theory* by R. E. Kalman, P. L. Falb, and M. A. Arbib, Part III; Automata Theory. New York: McGraw-Hill, 1969) IEEE Trans. Auto. Cont., Vol. AC-17, p. 182, Feb. 1972.
- [4] J. L. Massey, (*Error-Correcting Codes*, 2nd Edition, by W. W. Peterson and E. J. Weldon, Jr., Cambridge, Mass. and London, M.I.T. Press, 1972) IEEE Trans. Info. Th., vol IT-19, pp. 373-374, May 1973.
- [5] J. L. Massey, (*The Theory of Error-Correcting Codes* by F. J. MacWilliams and N. J. A. Sloane. Amsterdam: North-Holland and New York: Elsevier/North Holland, 1977) IEEE Trans. Info. Th., Vol. IT-15, pp. 501-502, July, 1979.
- [6] J. L. Massey, (*Theory and Practice of Error Control Codes* by R. E. Blahut. Reading, MA: Addison-Wesley, 1983) IEEE Trans. Info. Th. , Vol. IT-31, pp. 553-554, July 1985. Reprinted in Proc. IEEE, Vol. 74, pp. 1293-1294, 1986.

## VIII U. S. Patents

- [1] J. L. Massey, U. S. Patent No. 3,303,333, "Error Detection and Correction System for Convolutional Codes," Feb. 7, 1967.
- [2] J. L. Massey, U. S. Patent No. 3,402,393, "Error Detection and Correction in Signal Transmission by Use of Convolutional Codes," Sept. 17, 1968.

- [3] J. L. Massey, U. S. Patent No. 3,439,334, "Processing Signal Information," April 15, 1969.
- [4] J. L. Massey, U. S. Patent No. 3,500,320, "Error Correcting Means for Digital Transmission Systems," March 10, 1970.
- [5] J. L. Massey, U. S. Patent No. 3,566,352, "Error Correction in Coded Messages," Feb. 23, 1971.
- [6] J. L. Massey and T. Schaub, U. S. Patent No. 4,506,372, "Method and Apparatus for Recognizing in a Receiver the Start of a Telegram Signal Consisting of a Bit Impulse Sequence," March 19, 1985.
- [7] J. L. Massey and J. K. Omura, U. S. Patent No. 4,567,600, "Method and Apparatus for Maintaining the Privacy of Digital Messages Conveyed by Public Means," Jan. 28, 1986.
- [8] J. K. Omura and J. L. Massey, U. S. Patent No. 4,587,627, "Computational Method and Apparatus for Finite Field Arithmetic," May 6, 1986.
- [9] J. L. Massey and R. A. Rueppel, U. S. Patent No. 4,797,922, "Method of, and Apparatus for, Transforming a Digital Data Sequence into an Encoded Form," Jan. 10, 1989.
- [10] J. L. Massey, U. S. Patent No. 5,170,412, "Multiple Access Method," Dec. 8, 1992.
- [11] J. L. Massey and X. Lai, U. S. Patent No. 5,214,703, "Device for the Conversion of a Digital Block and Use of Same," May 25, 1993.