CODES, CURVES, AND SIGNALS

Common Threads in Communications

The Kluwer International Series in Engineering and Computer Science

Communications and Information Theory

Consulting Editor

ROBERT G. GALLAGER

Other books in the series:

PERSPECTIVES IN SPREAD SPECTRUM, Amer A. Hassan, John E. Hershey, and Gary J. Saulnier; ISBN: 0-7923-8265-X

WIRELESS PERSONAL COMMUNICATIONS: ***Advances in Coverage and Capacity***, Jeffrey H. Reed, Theodore S. Rappaport, Brian D. Woerner; ISBN: 0-7923-9788-6

ASYMPTOTIC COMBINATORIAL CODING THEORY, Volodia Blinovsky; ISBN: 0-7923-9988-9

PERSONAL AND WIRELESS COMMUNICATIONS: ***Digital Technology and Standards***, Kun Il Park; ISBN: 0-7923-9727-4

WIRELESS INFORMATION NETWORKS: ***Architecture, Resource Managment, and Mobile Data***, Jack M. Holtzman; ISBN: 0-7923-9694-4

DIGITAL IMAGE COMPRESSION: ***Algorithms and Standards***, Weidong Kou; ISBN: 0-7923-9626-X

CONTROL AND PERFORMANCE IN PACKET, CIRCUIT, AND ATM NETWORKS, XueDao Gu, Kazem Sohraby and Dhadesugoor R. Vaman; ISBN: 0-7923-9625-1

DISCRETE STOCHASTIC PROCESSES, Robert G. Gallager; ISBN: 0-7923-9583-2

WIRELESS PERSONAL COMMUNICATIONS: ***Research Developments***, Brian D. Woerner, Theodore S. Rappaport and Jeffrey H. Reed; ISBN: 0-7923-9555-7

PLANNING AND ARCHITECTURAL DESIGN OF INTEGRATED SERVICES DIGITAL NETWORKS, A. Nejat Ince, Dag Wilhelmsen and Bilent Sankur; ISBN: 0-7923-9554-9

WIRELESS INFRARED COMMUNICATIONS, John R. Barry; ISBN: 0-7923-9476-3

COMMUNICATIONS AND CRYPTOGRAPHY: ***Two sides of One Tapestry***, Richard E. Blahut, Daniel J. Costello, Jr., Ueli Maurer and Thomas Mittelholzer; ISBN: 0-7923-9469-0

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

WIRELESS PERSONAL COMMUNICATIONS: ***Trends and Challenges***, Theodore S. Rappaport, Brian D. Woerner, Jeffrey H. Reed; ISBN: 0-7923-9430-5

ELLIPTIC CURVE PUBLIC KEY CRYPTOSYSTEMS, 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

CODES, CURVES, AND SIGNALS

Common Threads in Communications

Edited by

ALEXANDER VARDY

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

ISBN 978-1-4613-7330-8 ISBN 978-1-4615-5121-8 (eBook)
DOI 10.1007/978-1-4615-5121-8

Library of Congress Cataloging-in-Publication Data

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

Originally published by Kluwer Academic Publishers in 1998
Softcover reprint of the hardcover 1st edition 1998

Printed on acid-free paper.

Dedicated to Richard E. Blahut

Contents

Foreword

This book is a direct consequence of the broad and profound influence of Richard E. Blahut on the fields of algebraic coding, information theory, and digital signal processing. The book consists of seventeen chapters that provide a representative cross-section of cutting-edge contemporary research in the fields of study to which R.E. Blahut has contributed during his career. These include: codes defined on algebraic curves and the associated decoding algorithms, the use of signal processing techniques in coding theory, and the application of information-theoretic methods in communications and signal processing. The book is organized accordingly into three parts, titled: *Curves and Codes*, *Codes and Signals*, and *Signals and Information*.

The list of contributors to this volume (see p.xv) consists of 21 researchers, from six countries in North America, Europe, and Asia. All the authors have individually and collectively dedicated their work to R.E. Blahut --- this book is thus testimony not only to the impact of his scholarship, but also to the deep affection in which he is held around the world.

Richard E. Blahut was born in 1937 in Orange, New Jersey. He received his B.Sc. degree in electrical engineering from the Massachusetts Institute of Technology, M.Sc. degree in physics from the Stevens Institute of Technology, and Ph.D. degree in electrical engineering from Cornell University.

He is now a Professor of Electrical and Computer Engineering at the University of Illinois and a Research Professor in the Coordinated Science Laboratory. Previously, he had served as a Courtesy Professor of Electrical Engineering at Cornell University, where he taught from 1973 to 1994, and as a Consulting Professor at the South China University of Technology. He has taught at Princeton University, the Swiss Federal Institute of Technology, and the NATO Advanced Study Institute. In addition to his numerous academic positions over the past 24 years, he was employed by the Federal Systems Division of IBM from 1964 to 1994. At IBM, he had general responsibility for the analysis and design of coherent signal processing systems, digital communications systems, and statistical information processing systems. Early in his career at IBM, he has pioneered the emitter location technology that is now used in the U.S. Department of Defense surveillance systems. For this work, he was named Fellow of the IBM Corporation in 1980. His direct contributions to industry also include the development of error-control codes and decoders that protect both the video-text data transmitted via the U.S. public broadcasting network and the messages to the Tomahawk missile. He designed a damage-resistant bar code used by the British Royal Mail, and developed error-control algorithms used in the data link for the U.S. Navy's LAMPS helicopter.

R.E. Blahut is a Systems Consultant to the Information Optics Corporation, Issaquah, Washington, where he is involved in the development of the first truly two-dimensional

data storage device. Some of the challenging theory of two-dimensional recording systems is examined in Chapter 10 of this volume, contributed by Jørn Justesen.

For his paper *Computation of channel capacity and rate distortion functions*, he received the 1974 IEEE Information Theory Society best paper Award. The impact of this work on our field is still felt today. Two chapters in this volume (Chapter 13 by Joseph A. O'Sullivan and Chapter 14 by Gottfried Ungerboeck) relate to the well-known Blahut-Arimoto algorithm for computing the capacity of discrete channels.

During the years 1975--1985, Richard E. Blahut has championed the use of signal processing techniques (in particular, Fourier transforms over a finite field) in algebraic coding theory. His ideas turned out to be extremely productive. Three chapters in this volume are devoted to Fourier transforms. Chapter 7 by G. David Forney, Jr. extends the theory to transforms defined over abelian groups, Chapter 8 by Thomas Ericson uses Fourier transforms to design spherical codes, and Chapter 9 by James L. Massey examines applications of the Günther-Blahut theorem (which relates the DFT to linear complexity) in coding theory and cryptography.

Later in his career, R.E. Blahut became interested in codes defined on algebraic curves. He pursued this subject with remarkable determination to explain the underlying mathematics on an elementary level. The paradigm of "algebraic-geometry codes without algebraic geometry" is universally attributed to R.E. Blahut. The first six chapters in this volume relate to his work in this field.

These contributions notwithstanding, his major project of the past two decades was a series of advanced textbooks and monographs in error-control coding, information theory, and signal processing, including:

Theory and Practice of Error-Control Codes, Addison-Wesley, 1983

Fast Algorithms for Digital Signal Processing, Addison-Wesley, 1985

Principles and Practice of Information Theory, Addison-Wesley, 1987

Digital Transmission of Information, Addison-Wesley, 1990

Algebraic Methods of Signal Processing and Communication Coding, Springer-Verlag, 1992.

These books were translated into several languages, and a generation of students worldwide has been introduced to information theory and coding theory using this material. There is more to come: R.E. Blahut is currently working on revisions to some of his earlier textbooks, and on new books in the series. By way of an advance preview, Chapter 17 in this volume (contributed by David C. Munson, Jr.) relates to Blahut's forthcoming book on remote surveillance.

R.E. Blahut served as President of the Information Theory Society of the IEEE in 1982, and as Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY, from

1992 until 1995. Among other honors, he was named Fellow of the Institute of Electrical and Electronics Engineers in 1981, elected to membership in the National Academy of Engineering in 1990, and received the IEEE Alexander Graham Bell medal in 1998.

As editor, I would like to conclude this foreword on a personal note. I feel fortunate to have chosen coding theory and information theory as my field of research. It is a field in which insights derived from theoretical investigation often have immediate and far-reaching impact on technology and practice. The work of R.E. Blahut has exemplified this trait in the best possible way, throughout his career. Furthermore, our field has inherent intellectual beauty that has always attracted the best and brightest people to information theory, and we have been particularly lucky to have Richard E. Blahut join our research community 30 years ago. His insights, his philosophy, and his personality left an indelible mark on our field.

Alexander Vardy
Urbana, Illinois

Acknowledgement

This book grew out of the workshop in honor of Richard E. Blahut -- the BlahutFest -- that was held on the campus of the University of Illinois at Urbana-Champaign in September 1997. Bruce E. Hajek, David C. Munson, Jr., and Dilip V. Sarwate helped with the organization of this workshop, and I am indebted to them for their efforts. I wish to acknowledge the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering at the University of Illinois for hosting the BlahutFest and providing financial support for this event. I would like to thank the speakers at the BlahutFest: Toby Berger, Ian F. Blake, A. Robert Calderbank, Daniel J. Costello, Jr., Anthony Ephremides, Thomas Ericson, Gui-Liang Feng, David G. Forney, Jr., Tom Høholdt, Jørn Justesen, David C. Munson, Jr., Joseph A. O'Sullivan, Ruud Pellikaan, Shojiro Sakata, Gottfried Ungerboeck, Judy L. Walker, and Jack K. Wolf. Each of these individuals has contributed to the success of the workshop. I am grateful to Francie Bridges, Amy Hendrickson, and Robert F. MacFarlane for their help in typesetting this manuscript.

Last but not least, this book owes its existence to the authors of the fine chapters collected herein. The full list of contributors to this volume may be found on the next page; working with each of them was a pleasure and a privilege.

List of Contributors

TOBY BERGER, *Cornell University*, Ithaca, NY, USA

IAN F. BLAKE, *Hewlett-Packard Labs*, Palo Alto, CA, USA

A. ROBERT CALDERBANK, *Shannon Labs*, Florham Park, NJ, USA

DANIEL J. COSTELLO, JR., *University of Notre Dame*, South Bend, IN, USA

ANTHONY EPHREMIDES, *University of Maryland,* College Park MD, USA

THOMAS ERICSON, *Linköping University*, Linköping, Sweden

GUI-LIANG FENG, *University of Southwestern Louisiana,* Lafayette, LA, USA

G. DAVID FORNEY, JR., *Motorola, Inc.*, Mansfield, MA, USA

JIALI HE, *University of Notre Dame*, South Bend, IN, USA

TOM HØHOLDT, *Technical University of Denmark*, Lyngby, Denmark

JØRN JUSTESEN, *Technical University of Denmark*, Lyngby, Denmark

JAMES W. MANDELL, *University of Virginia*, Charlottesville, VA, USA

JAMES L. MASSEY, *ETH*, Zurich, Switzerland, and Copenhagen, Denmark

DAVID C. MUNSON, JR., *University of Illinois at Urbana-Champaign*, IL, USA

JOSEPH A. O'SULLIVAN, *Washington University*, St. Louis, MO, USA

RUUD PELLIKAAN, *Eindhoven University of Technology*, Eindhoven, The Netherlands

THAMMAVARAPU R. N. RAO, *University of Southwestern Louisiana,* Lafayette, LA, USA

SHOJIRO SAKATA, *University of Elecrtro-Communications*, Tokyo, Japan

GOTTFRIED UNGERBOECK, *IBM Research*, Zurich, Switzerland

ALEXANDER VARDY, *University of Illinois at Urbana-Champaign*, IL, USA

JOSÉ FELIPE VOLOCH, *University of Texas*, Austin, TX, USA

JUDY L. WALKER, *University of Nebraska*, Lincoln, NE, USA

Part I

CURVES AND CODES

On (or in) the Blahut Footprint

Tom Høholdt

Department of Mathematics
Technical University of Denmark
DK–2800 Lyngby, Denmark
tom@mat.dtu.dk

Abstract. We review the notion of a *footprint* and discuss its applications in the decoding of codes from algebraic curves.

1. Introduction

Richard E. Blahut suggested in 1991 the term *footprint* for a particular set of points in the plane that occurs in the construction and decoding of a class of error-correcting codes. The same set was previously called the *deltaset*, the *excluded point set* and some other things, but somehow the name footprint has now become standard. In this paper, we will recall the definition of a footprint, determine its size, and explain some of its uses. The phrase "(or in)" in the title refers to the fact that we will try to explain everything in as simple terms as possible, thereby staying within the tradition initiated by R.E. Blahut.

2. The footprint

We will consider ideals in the ring $\mathcal{R} = \mathbb{F}_q[x, y]$, where $\mathbb{F}_q$ is a finite field with $q = p^r$ elements, for some prime p and some natural number r.

Let $\preccurlyeq$ be an *admissible* ordering of the set of monomials $\mathcal{M}$ in $\mathcal{R}$, meaning that $1 \preccurlyeq M(x, y)$ for any nonconstant monomial $M(x, y)$, and $M_1(x, y) \preccurlyeq M_2(x, y)$ implies $M_1(x, y)M(x, y) \preccurlyeq M_2(x, y)M(x, y)$, where $M_1, M_2, M \in \mathcal{M}$. As usual, we can now talk about the *leading monomial* $\mathrm{Lm}(F)$ and the *leading term* $\mathrm{Lt}(F)$ of a polynomial $F(x, y) \in \mathbb{F}_q[x, y]$. Let $\mathcal{B}$ be a finite subset of $\mathcal{R}$. Then $\mathcal{B}$ is a *Gröbner basis* if and only if

$$\{\mathrm{Lm}(F) : F \in (\mathcal{B}), F \neq 0\} = \{\mathrm{Lm}(BM) : B \in \mathcal{B}, M \in \mathcal{M}\}$$

where $(\mathcal{B})$ denotes the ideal generated by $\mathcal{B}$. The *footprint* $\Delta(\mathcal{B})$ of a Gröbner basis $\mathcal{B}$ is defined as follows:

$$\Delta(\mathcal{B}) = \mathcal{M} \setminus \{\mathrm{Lm}(BM) : B \in \mathcal{B}, M \in \mathcal{M}\} = \mathcal{M} \setminus \{\mathrm{Lm}(F) : F \in (\mathcal{B}), F \neq 0\}$$

This means that the footprint consists of the monomials that are not leading monomials of any polynomial from the ideal generated by $\mathcal{B}$. If $I \subseteq \mathcal{R}$ is an ideal and $I = (\mathcal{B})$ then $\mathcal{B}$ is a Gröbner basis for I and we put $\Delta(I) = \Delta(B)$.

Example. Let $\preccurlyeq$ be the graded reverse lexicographic order. In other words, $x^{a_1}y^{b_1} \preccurlyeq x^{a_2}y^{b_2}$ if $a_1 + b_1 < a_2 + b_2$ or $a_1 + b_1 = a_2 + b_2$ and $b_1 < b_2$, so that

$$1 \preccurlyeq x \preccurlyeq y \preccurlyeq x^2 \preccurlyeq xy \preccurlyeq y^2 \preccurlyeq \cdots$$

Let $\mathcal{B} = \{x^2, y^2 + 1\}$, then $\mathcal{B}$ is a Gröbner basis and the footprint is $x^a y^b$, where $a + b \leq 1$ or $a = b = 1$. The set $\mathcal{A} = \{x^2, x^2y + 1\}$ is not a Gröbner basis, since $(\mathcal{A})$ contains the polynomial y. □

One of the ways to use the footprint is the following. Let $\mathcal{B}$ be a Gröbner basis for an ideal I. Then the cosets of the elements of $\Delta(\mathcal{B})$ form a basis of $\mathcal{R}/I$ as an $\mathbb{F}_q$ vector space, so in particular

$$\dim_{\mathbb{F}_q} \mathcal{R}/I \;=\; |\Delta(\mathcal{B})|$$

This observation leads to the determination of $|\Delta(\mathcal{B})|$, when $\mathcal{B}$ is a Gröbner basis for a zero-dimensional ideal.

Theorem 1. *Let $P_1, \ldots, P_s$ be different points of $\mathbb{F}_q^2$, and let the ideal I be defined by $I = \{F \in \mathcal{R} : F(P_i) = 0$ for $i = 1, 2, \ldots, s\}$. Then $|\Delta(I)| = s$.*

Proof. We look at the evaluation map $ev : \mathbb{F}_q[x, y] \to \mathbb{F}_q^s$ which is defined by $ev(F(x, y)) = (F(P_1), , \ldots, F(P_s))$. This is clearly a linear mapping. Now, if $P_i = (a_i, b_i)$ for $i = 1, 2, \ldots, s$, then the polynomials $G_i(x, y)$ defined by

$$G_i(x, y) \;=\; \prod_{a_\ell \neq a_i} (x - a_\ell) \prod_{b_\ell \neq a_i} (y - b_\ell)$$

satisfy $G_i(P_i) \neq 0$ and $G_i(P_j) = 0$ for $i \neq j$. Hence it follows that the vectors $ev(G_1), ev(G_2), \ldots, ev(G_s)$ constitute a basis for $\mathbb{F}_q^s$, and the mapping ev is surjective. Thus, since $F \in I$ implies $F(P_i) = 0$, the mapping $\widehat{ev} : \mathbb{F}_q[x, y]/I$, where $\widehat{ev}(F + I) = (F(P_1), \ldots, F(P_s))$, is well-defined and also surjective.

On the other hand this mapping is also injective since $\widehat{ev}(F + I) = \underline{0}$ implies $F(P_i) = 0$ for $i = 1, \ldots, s$, so that $F \in I$. Combining the two statements shows that $\mathbb{F}_q^s \simeq \mathbb{F}_q[x, y]/I$, and hence $\dim \mathbb{F}_q[x, y]/I = |\Delta(I)| = s$. ■

The material presented in this section is not new. For a more detailed exposition, we refer the reader to [1] for instance.

3. An application to decoding

In this section we will show how the footprint is used in decoding of codes coming from Hermitian curves. For a survey of decoding methods see [2].

Recall [6] that the Hermitian curve is defined by the equation $x^{r+1} + y^r + y = 0$, where $q = r^2$ is an even power of a prime. The curve has $n = r^3$ rational points over $\mathbb{F}_q$, namely points $P_i(a_i, b_i)$ with $a_i, b_i \in \mathbb{F}_q$ that satisfy the curve equation.

Let $M = \{x^i y^i : i \geq 0,\ 0 \leq j < r\}$. For each $x^i y^j \in M$, we define the *order* ρ as $\rho(x^i y^j) = ir + j(r+1)$. Then this gives a total ordering of the elements of M, which can be extended to the $\mathbb{F}_q$-vectorspace $\mathcal{R}$ of polynomials generated by the elements of M. Let φ_i be the elements of M with respect to this ordering. Thus

i	1	2	3	4	5	6	$\cdots$
φ_i	1	x	y	x^2	xy	y^2	$\cdots$
$\rho(\varphi_i)$	0	r	$r+1$	r^2	$r(r+1)$	$(r+1)^2$	$\cdots$

For $\underline{y} \in \mathbb{F}_q^n$ and $f \in \mathcal{R}$ we define the *syndrome* $S_{\underline{y}}(f)$ as follows:

$$S_{\underline{y}}(f) \;=\; y_1 f(P_1) + y_2 f(P_2) + \cdots + y_n f(P_n)$$

and define the code $\mathbb{C}_m$ by the relation

$$\underline{y} \in \mathbb{C}_m \iff S_{\underline{y}}(f) = 0 \qquad \text{for all } f \in \mathcal{R} \text{ such that } \rho(f) \leq m$$

It is well known [**2**] that if $r(r-1) - 2 < m < n$ then the dimension of $\mathbb{C}_m$ is given by $n - \left(m + \frac{1}{2}r(r-1) - 1\right)$ and the minimum distance of $\mathbb{C}_m$ is greater than or equal to $m - r(r-1) + 2$, with equality if $m \geq 2r(r-1) - 2$.

In the decoding situation, we receive a vector $\underline{y}$ which is the sum of a codeword $\underline{c}$ and an error vector $\underline{e}$. We have

$$S_{\underline{e}}(f) \;=\; S_{\underline{y}}(f) \quad \text{if } \rho(f) \leq m$$

so the syndromes $S_{\underline{e}}(f)$ can be calculated directly from the received word if $\rho(f) \leq m$. If τ is the Hamming weight of $\underline{e}$, then it is well known [**4**] that if one knows the syndromes $S_{\underline{e}}(f)$ where $m < \rho(f) \leq 2(\tau + r(r+1)) - 1$ then the error vector can be easily found. Thus the objective of the decoder is to determine the syndromes $S_{\underline{e}}(f)$, where $m < \rho(f) \leq 2(\tau + r(r-1)) - 1$.

To formulate the decoding algorithm we introduce a partial ordering $\leq_p$ on $\rho(\mathcal{R})$ as follows. We say that $\rho(f) \leq_p \rho(g)$ if there exists an $h \in \mathcal{R}$ such that $\rho(fh) = \rho(g)$. We define the *span* of an element $f \in \mathcal{R}$ as follows

$$\mathsf{span}(f) \;=\; \rho(\varphi_i) \qquad \text{if } S(f\varphi_i) \neq 0 \text{ but } S(f\varphi_j) = 0 \text{ for all } j < i$$

Notice that here, and hereafter, we omit the subscript $\underline{e}$ in the syndrome notation. All the syndromes should be understood as error syndromes.

The decoding algorithm described below is a version of Sakata's generalization of the Berlekamp-Massey algorithm [**5**]. The first part of the algorithm consists of Steps 1, 2, and 3. This part is an iterative procedure that, given the syndromes $S(\varphi_r)$ for all φ_r such that $\rho(\varphi_r) \leq m$, computes two sets of functions F_M and G_M, where $\rho(\varphi_M) = m$, and a set of orders Δ_M. The second part of the algorithm consists of Steps 4 and 5. This part uses the obtained sets $F_{M'}$, $G_{M'}$, $\Delta_{M'}$ for $M' \geq M$, to determine $S(f)$, where $\rho(f) = m' + 1$ for $m' \geq m$.

Decoding Algorithm: Input $S(\varphi_r)$ for all φ_r such that $\rho(\varphi_r) \leq m$

Initialization: $F_0 = \{1\}, \Delta_0 = \varnothing, G_0 = \varnothing$; and $A = \varnothing$ at order $\rho(\varphi_{\ell+1})$

Step 1. For each $f \in F_\ell$ do:

if $\rho(f) \not\leq_p \rho(\varphi_{\ell+1})$ or $S(f\varphi_{\ell+1}) = 0$ then $f \in F_{\ell+1}$
else
if $\rho(\varphi_{\ell+1}) - \rho(f) \leq_p \text{span}(g)$ for some $g \in G_\ell$
then $\widehat{f} = f + \beta g \varphi_j \in F_{\ell+1}$
where $\rho(\varphi_j) = \text{span}(g) - (\rho(\varphi_{\ell+1}) - \rho(f))$ and $\beta \in \mathbb{F}_q^*$
else $f \in G_{\ell+1}$ and $f \in A$

Step 2. For each $g \in G_\ell$ do:

if $\text{span}(g) \not\leq_p \text{span}(f)$ for some $f \in A$
then $g \in G_{\ell+1}$ and

$$\Delta_{\ell+1} = \{\rho : \rho \leq_p \text{span}(g), g \in G_{\ell+1}\}$$

Step 3. For each $\rho \notin \Delta_{\ell+1}$, minimal with respect to $\leq_p$ do:

$$\rho = \rho(f) + \rho(\varphi_i), \quad f \in A, \quad \rho(\varphi_i) > 0$$

if $\rho(\varphi_{\ell+1}) - \rho \leq_p \text{span}(g)$ for some $g \in G_\ell$
then

$$\widehat{f} = \varphi_i f + \beta \varphi_j g \in F_{\ell+1}$$

where $\rho(\varphi_j) = \text{span}(g) - (\rho(\varphi_{\ell+1}) - \rho)$ and $\beta \in \mathbb{F}_q^*$
else

$$\widehat{f} = f\varphi_i \in F_{\ell+1}$$

Remark. To explain the voting procedure in Step 4, we need the following notation. Let $\Sigma_i = \rho(\mathcal{R}) \setminus \Delta_i$ and define:

$$\Gamma_i = \{\rho \in \Sigma_{i-1} : \rho \leq_p \rho(\varphi_i) \text{ and } \rho(\varphi_i) - \rho \in \Sigma_{i-1}\}$$

Suppose we have $F_{M'}, G_{M'}, \Delta_{M'}$, and hence also $\Sigma_{M'}, \Gamma_{M'}$, for some $M' \geq M$.

Step 4. For each $\rho \in \Gamma_{M'+1}$, choose $\rho(\varphi_s) \in \Sigma_{M'}$ minimal with respect to $\leq_p$, such that $\rho(\varphi_s) \leq_p \rho$, and select $\omega \in \mathbb{F}_q^*$ such that:

$$g = \varphi_{M'+1} + F(\rho(\varphi_s))\varphi_{M'+1-s} \cdot \omega$$

satisfies $\rho(g) < \rho(\varphi_{M'+1})$, where $F(\rho(\varphi_s))$ denotes the elements of $F_{M'}$ with order $\rho(\varphi_s)$.

Step 5. Let the vote by g for $S(\varphi_{M'+1})$ be $S(g)$. Put:

$$S(\varphi_{M'+1}) = \text{majority}_{\,g \text{ as in Step 4}} \{S(g)\}$$

That the above algorithm actually solves the decoding problem when $\tau \leq \lfloor \frac{d-1}{2} \rfloor$ can be seen in [4]. Intuitively the reason for the success in Steps 4 and 5 is the following. Let $P_{i_1}, \ldots, P_{i_\tau}$ be the error points and let

$$I_{\underline{e}} = \{F \in \mathcal{R} : F(P_{i_j}) = 0, \quad i = 1, 2, \ldots, \tau\}$$

This is the so-called error-locator ideal. If $F \in I_{\underline{e}}$ then $S(F\varphi) = 0$ for all $\varphi \in \mathcal{R}$, since

$$S(F\varphi) = \sum_{i=1}^{n} F(P_i)\varphi(P_i)e_i = \sum_{j=1}^{\tau} F(P_{i_j})\varphi(P_{i_j})e_{i_j} = 0$$

and on the other hand if $S(F\varphi) = 0$ for all $\varphi \in \mathcal{R}$, then $F \in I_{\underline{e}}$, since for each $\ell \in \{1, 2, \ldots, \tau\}$ we can choose φ such that $\varphi(P_{i_j}) = 0$, $j \neq \ell$, and $\varphi(P_{i_\ell}) = 0$. Actually, is is sufficient that $S(F\varphi) = 0$ for $\rho(\varphi) \leq 2(\tau + r(r-1)) + 1$.

At each step of the algorithm, we have $\Delta_j \subseteq \Delta(I_{\underline{e}})$, so since $|\Delta(I_{\underline{e}})| = \tau$ we also have $|\Delta_j| \leq \tau$. It can be seen that a choice of $S(\varphi_{M'+1})$ different from the one made in Step 5 would cause $\Delta_{M'+1}$ to have size greater than τ, contradicting the above. Thus it is precisely our knowledge of the size of the footprint that makes the algorithm work.

Actually, by a more careful study of the footprint one can analyze the performance of the above algorithm, this is the subject of the forthcoming paper [3].

4. Conclusions

We have defined the footprint, determined its size and demonstrated its application in the decoding of algebraic geometry codes.

References

[1] D. COX, J. LITTLE, AND D. O'SHEA, *Ideals, Varieties and Algorithms*, New York: Springer-Verlag 1992.

[2] T. HØHOLDT AND R. PELLIKAAN, On the decoding of algebraic-geometric codes, *IEEE Trans. Inform. Theory*, vol. 41, pp. 1589–1614, 1995.

[3] H. ELBRØND JENSEN, R. REFSLUND NIELSEN, AND T. HØHOLDT, Performance analysis of a decoding algorithm for algebraic geometry codes, MAT-Report 1997, Department of Mathematics, Technical University of Denmark.

[4] M.E. O'SULLIVAN, Decoding of codes defined by a single point on a curve, *IEEE Trans. Inform. Theory*, vol. 41, pp. 1709–1719, 1995.

[5] S. SAKATA, H. ELBRØND JENSEN, AND T. HØHOLDT, Generalized Berlekamp-Massey decoding of algebraic-geometric codes up to half the Feng-Rao bound," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1762–1769, 1995.

[6] H. STICHTENOTH, *Algebraic Function Fields and Codes*, Berlin: Springer-Verlag 1993.

The Klein Quartic, the Fano Plane and Curves Representing Designs

Ruud Pellikaan

Department of Mathematics and Computing Science
Technical University of Eindhoven
5600 MB Eindhoven, The Netherlands
ruudp@win.tue.nl

1. Introduction

The projective plane curve with defining equation $X^3Y + Y^3Z + Z^3X = 0$ has been studied for numerous reasons since Klein [**19**].

Hurwitz [**16, 17**] showed that a curve considered over the complex numbers has at most $84(g-1)$ automorphisms, where g is the genus and $g > 1$. The above curve, called the *Klein quartic*, has genus 3 and an automorphism group of 168 elements. Thus it is optimal with respect to the number of automorphisms.

This curve has 24 points with coordinates in the finite field of 8 elements. This is optimal with respect to Serre's improvement of the *Hasse-Weil bound*:

$$N \leq q + 1 + g\lfloor 2\sqrt{q} \rfloor$$

where N is the number of $\mathbb{F}_q$-rational points of the curve [**22**]. Therefore the geometric Goppa codes on the Klein quartic have good parameters, and these were studied by many authors after [**13**].

R.E. Blahut challenged coding theorists to give a self-contained account of the properties of codes on curves [**2**]. In particular, it should be possible to explain this for codes on the Klein quartic. With the decoding algorithm [**7**] by a *majority vote* among *unknown syndromes*, his dream became true. The elementary treatment of algebraic geometry codes [**8, 9, 10, 14**] is now based on the following observation: if a decoding algorithm corrects t errors, then the minimum distance of the code is at least $2t + 1$. This is very much the point of view of Blahut in his book [**3**], where he proves the BCH bound on the minimum distance as a corollary of a decoding algorithm.

When Blahut was in Eindhoven for a series of lectures on algebraic geometry codes [**4**], he gave the picture of the Klein quartic over $\mathbb{F}_8$ shown in Figure 1. Let α be an element of $\mathbb{F}_8$ such that $\alpha^3 + \alpha + 1 = 0$. Then α is a primitive

element of $\mathbb{F}_8$. The $\bullet$ in row i and column j of the diagram in Figure 1 means that (α^i, α^j) is a point of the Klein quartic with affine equation:

$$X^3Y + Y^3 + X = 0$$

We get all points with nonzero coordinates in $\mathbb{F}_8$ in this way. It was noted by J.J. Seidel, who was in the audience, that this diagram can be interpreted as the incidence matrix of the the Fano plane*, that is the projective plane of order two. This means that every row has three $\bullet$. The same holds for every column. Furthermore for every two distinct columns there is exactly one row with a $\bullet$ in that row and in those given columns.

	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^6		$\bullet$				$\bullet$	$\bullet$
α^5			$\bullet$	$\bullet$		$\bullet$	
α^4	$\bullet$		$\bullet$				$\bullet$
α^3				$\bullet$	$\bullet$		$\bullet$
α^2	$\bullet$	$\bullet$		$\bullet$			
α^1	$\bullet$				$\bullet$	$\bullet$	
α^0		$\bullet$	$\bullet$		$\bullet$		

Figure 1. *The Klein quartic in* $\mathbb{F}_8^{*2}$

This paper grew out of an attempt to understand this phenomenon. Consider the affine plane curve $\mathcal{X}_q$ with equation

$$F_q(X,Y) = X^{q+1}Y + Y^{q+1} + X = 0 \qquad (1)$$

over the finite field $\mathbb{F}_{q^3}$. Let $\mathcal{P}_q$ be the cyclic subgroup of the units of $\mathbb{F}_{q^3}^*$ of order q^2+q+1. Thus

$$\mathcal{P}_q = \{ x \in \mathbb{F}_{q^3}^* : x^{q^2+q+1} = 1 \}$$

Define an incidence relation on $\mathcal{P}_q$ by:

$$x \text{ is incident with } y \text{ if and only if } F_q(x,y) = 0$$

Then this defines an incidence structure of a projective plane of order q, that is to say a 2-$(q^2+q+1, q+1, 1)$ design. This means that there are q^2+q+1 points, that the lines $B_y = \{x \in \mathcal{P}_q : F_q(x,y) = 0\}$ have $q+1$ elements for all $y \in \mathcal{P}_q$, that there is exactly one line incident with two distinct points, and that there is exactly one point incident with two distinct lines.

For every q a power of a prime there exists a projective plane of order q, called the *Desarguesian plane* and denoted by $\mathrm{PG}(2,q)$. Not all projective

*This refers to G. Fano (1871-1952), one of the pioneers in finite geometry, not to be confused with R.M. Fano known for his contributions to information theory.

planes of a given order are isomorphic. It was explained to me by S. Ball and A. Blokhuis [1] that the above projective plane of order q is Desarguesian.

In the next section, the notion of curves or bivariate polynomials representing t-designs is defined. It is shown that $F_q(X,Y)$ represents $\mathrm{PG}(2,q)$. In § 3, the number of $\mathbb{F}_{q^3}$-rational points is computed of the projective curve $\mathcal{X}_q$ with affine equation $F_q(X,Y)=0$. In § 4, a general construction is given of a bivariate polynomial representing symmetric designs, starting with a univariate polynomial. A complete characterization is given of polynomials $\alpha X^n Y + Y^{n+1} + \beta X$ that represent finite projective planes. It is shown that the design of points/hyperplanes of the projective geometry $\mathrm{PG}(m,q)$ is represented by a curve.

2. Designs represented by curves

The name design comes from the design of statistical experiments. For its basic properties and its relations with coding theory, the reader is referred to [**5, 20**].

Definition 1. *Let t, v, k, and λ be positive integers. Let $\mathcal{P}$ be a set of v elements, called points. Let $\mathcal{B}$ be a collection of elements, called blocks. Let $\mathcal{I} \subset \mathcal{P} \times \mathcal{B}$ be a relation between points and blocks. We say that $P \in \mathcal{P}$ is incident with $B \in \mathcal{B}$ if $(P,B) \in \mathcal{I}$. Assume that every block $B \in \mathcal{B}$ has k elements, and that for every choice of t distinct points of $\mathcal{P}$ there are exactly λ blocks of $\mathcal{B}$ that are incident with the given t-set. Then the triple $(\mathcal{P},\mathcal{B},\mathcal{I})$ is called a t-(v,k,λ) design.*

Let b be the number of blocks of a t-(v,k,λ) design. Then $b = \lambda\binom{v}{t}/\binom{k}{t}$. Every point is contained in the same number $r = bk/v$ of blocks.

A 2-design is called *symmetric* if $b = v$. If $(\mathcal{P},\mathcal{B},\mathcal{I})$ is a symmetric 2-design, then $(\mathcal{B},\mathcal{P},\mathcal{I}^{-1})$ is also a symmetric 2-design with the same parameters, where $\mathcal{I}^{-1} = \{(B,P) : (P,B) \in \mathcal{I}\}$. It is called the *dual design*.

Definition 2. *A 2-$(n^2+n+1, n+1, 1)$ design is called a projective plane of order n. The blocks are than called lines.*

For properties and constructions of finite projective planes, one consults [**6, 15**]. The standard example is the *Desarguesian projective plane* $\mathrm{PG}(2,q)$ of order q, where q is a power of a prime. A point P of $\mathrm{PG}(2,q)$ is a 1-dimensional linear subspace of $\mathbb{F}_q^3$, a line B of $\mathrm{PG}(2,q)$ is a 2-dimensional linear subspace of $\mathbb{F}_q^3$, and P is incident with B if and only if $P \subset B$.

Definition 3. *Let $F \in \mathbb{F}_q[X,Y]$. Let $\mathcal{P}$ and $\mathcal{B}$ be subsets of $\mathbb{F}_q$. Let $\mathcal{I}$ be the relation on $\mathcal{P} \times \mathcal{B}$ defined by $(x,y) \in \mathcal{I}$ if and only if $F(x,y) = 0$ for $x \in \mathcal{P}$ and $y \in \mathcal{B}$. We say that $(\mathcal{P},\mathcal{B},F)$, or simply F, represents a t-(v,k,λ) design if $(\mathcal{P},\mathcal{B},\mathcal{I})$ is a t-(v,k,λ) design, $\deg_X(F) = k$, and $\deg_Y(F) = r$ is the number of blocks through a given point of this design.*

Consider the polynomial of the introduction $F_q(X,Y) = X^{q+1}Y + Y^{q+1} + X$ over the field $\mathbb{F}_{q^3}$. Define

$$\mathcal{P}_q = \{x \in \mathbb{F}_{q^3} \ : \ x^{q^2+q+1} = 1\}$$

Let $\mathcal{I}_q$ be the incidence relation on $\mathcal{P}_q \times \mathcal{P}_q$ defined by $(x,y) \in \mathcal{I}_q$ if and only if $F_q(x,y) = 0$. The following lemma will be used several times in this paper.

Lemma 1. *The polynomial $f_q(T) = T^{q+1}+T+1$ has $q+1$ distinct zeros in $\mathcal{P}_q$.*

Proof. Let $\alpha^{q+1} + \alpha + 1 = 0$ for some α in the algebraic closure of the field $\mathbb{F}_q$. Then raising this equation to the q-th power and multiplying by α gives $\alpha^{q^2+q+1} + \alpha^{q+1} + \alpha = 0$. Thus

$$\alpha^{q^2+q+1} \ = \ -(\alpha^{q+1} + \alpha) \ = \ 1$$

Hence $\alpha \in \mathcal{P}_q$. The polynomial $f_q(T)$ has $q+1$ distinct zeros, since its derivative $f_q'(T) = T^q + 1$ has greatest common divisor 1 with $f_q(T)$. ∎

Proposition 2. *The triple $(\mathcal{P}_q, \mathcal{P}_q, F_q)$ represents a 2-$(q^2+q+1, q+1, 1)$ design.*

Proof. See also [1]. The polynomial $f_q(T)$ has $q+1$ distinct zeros in $\mathcal{P}_q$ by Lemma 1. Let $x \in \mathcal{P}_q$, and let α be a zero of $f_q(T)$. Further define $y = \alpha x^{-q}$. Then $y \in \mathcal{P}_q$. Substituting $\alpha = x^q y$ in $f_q(T)$ gives

$$(x^q y)^{q+1} + x^q y + 1 \ = \ 0$$

Multiplying by x gives $y^{q+1} + x^q y + x = 0$, since $x^{q^2+q+1} = 1$. Hence for every $x \in \mathcal{P}_q$ there are exactly $q+1$ elements $y \in \mathcal{P}_q$ such that $F_q(x,y) = 0$.

Now let $x_1, x_2 \in \mathcal{P}_q$ be such that $F_q(x_1,y) = F_q(x_2,y) = 0$ and $x_1 \neq x_2$. Then

$$(x_1^{q+1} - x_2^{q+1})y - (x_1 - x_2) \ = \ F_q(x_1,y) - F_q(x_2,y) \ = \ 0$$

The order of x_1/x_2 divides q^2+q+1 and is not one. Furthermore $q+1$ and q^2+q+1 are relatively prime. Thus $(x_1/x_2)^{q+1} \neq 1$. Hence $x_1^{q+1} - x_2^{q+1} \neq 0$ and

$$y \ = \ \frac{x_1 - x_2}{x_1^{q+1} - x_2^{q+1}}$$

is uniquely determined by x_1 and x_2. It follows that for every $x_1, x_2 \in \mathcal{P}_q$ and $x_1 \neq x_2$, there is at most one $y \in \mathcal{P}_q$ such that $F_q(x_1,y) = F_q(x_2,y) = 0$. Counting the set

$$\{ \ ((x_1,x_2),y) \ : \ x_1, x_2, y \in \mathcal{P}_q, \ x_1 \neq x_2, \ F_q(x_1,y) = F_q(x_2,y) = 0 \ \}$$

in two ways, it is easy to see that for every $x_1, x_2 \in \mathcal{P}_q$ and $x_1 \neq x_2$, there is exactly one $y \in \mathcal{P}_q$ such that $F_q(x_1,y) = F_q(x_2,y) = 0$. Hence $(\mathcal{P}_q, \mathcal{P}_q, \mathcal{I}_q)$ is a 2-$(q^2+q+1, q+1, 1)$ design. ∎

The parameters of $(\mathcal{P}_q, \mathcal{P}_q, \mathcal{I}_q)$ and $\mathrm{PG}(2,q)$ are the same, and indeed one can show that they are isomorphic as designs.

Proposition 3. *The triple $(\mathcal{P}_q, \mathcal{P}_q, \mathcal{I}_q)$ is isomorphic to the Desarguesian plane of order q.*

Proof. The proof is from [1]. The finite field $\mathbb{F}_{q^3}$ is isomorphic to $\mathbb{F}_q^3$ as an $\mathbb{F}_q$-vector space. Let ξ_0, ξ_1, ξ_2 be a basis of $\mathbb{F}_{q^3}$ over $\mathbb{F}_q$. Let $(x_0, x_1, x_2) \in \mathbb{F}_q^3$ be a nonzero vector. Then the line $\{\lambda(x_0, x_1, x_2) : \lambda \in \mathbb{F}_q\}$ in $\mathbb{F}_q^3$ is a point in $\mathrm{PG}(2,q)$, and is denoted by $(x_0{:}x_1{:}x_2)$. Furthermore $(x_0'{:}x_1'{:}x_2') = (x_0{:}x_1{:}x_2)$ if and only if $(x_0', x_1', x_2') = \lambda(x_0{:}x_1{:}x_2)$ for some $\lambda \in \mathbb{F}_q^*$. Let

$$\sigma(x_0, x_1, x_2) \;=\; (x_0\xi_0 + x_1\xi_1 + x_2\xi_2)^{q-1}.$$

Then $\sigma(x_0, x_1, x_2)^{q^2+q+1} = (x_0\xi_0 + x_1\xi_1 + x_2\xi_2)^{q^3-1} = 1$ for all nonzero vectors $(x_0, x_1, x_2) \in \mathbb{F}_q^3$. Thus $\sigma(x_0, x_1, x_2) \in \mathcal{P}_q$. Now $\sigma(\lambda x_0, \lambda x_1, \lambda x_2) = \sigma(x_0, x_1, x_2)$ for all nonzero $\lambda \in \mathbb{F}_q$, since $\lambda^{q-1} = 1$. Hence we have a well defined map

$$\sigma : \mathrm{PG}(2,q) \longrightarrow \mathcal{P}_q$$

given by $\sigma(x_0{:}x_1{:}x_2) = \sigma(x_0, x_1, x_2)$. The map is surjective, since every element of $\mathcal{P}_q$ is $(q-1)$-th power. The sets $\mathrm{PG}(2,q)$ and $\mathcal{P}_q$ both have q^2+q+1 elements. So the map σ is a bijection and we have identified the points of $\mathrm{PG}(2,q)$ with elements of $\mathcal{P}_q$. The lines of $\mathrm{PG}(2,q)$ are of the form

$$L(a_0, a_1, a_2) = \{\ (x_0{:}x_1{:}x_2) \in \mathrm{PG}(2,q) \ :\ a_0x_0 + a_1x_1 + a_2x_2 = 0\ \},$$

where (a_0, a_1, a_2) is a nonzero vector of $\mathbb{F}_q^3$. Given an $\mathbb{F}_q$-linear map $l : \mathbb{F}_q^3 \to \mathbb{F}_q$, define $L(l)$ as the set of points $(x_0{:}x_1{:}x_2)$ in $\mathrm{PG}(2,q)$ such that $l(x_0, x_1, x_2) = 0$.

Consider the trace map $\mathrm{Tr} : \mathbb{F}_{q^3} \to \mathbb{F}_q$ defined by $\mathrm{Tr}(u) = u^{q^2} + u^q + u$. Then Tr is an $\mathbb{F}_q$-linear map. More generally, let $v \in \mathbb{F}_{q^3}^*$. Define $l_v(u) = \mathrm{Tr}(uv)$. Then $l_v : \mathbb{F}_{q^3} \to \mathbb{F}_q$ is a nonzero $\mathbb{F}_q$-linear map, and every nonzero $\mathbb{F}_q$-linear function on $\mathbb{F}_{q^3}$ is of the form l_v.

For the chosen basis ξ_0, ξ_1, ξ_2 there is a unique dual basis α_0, α_1, α_2 such that $\mathrm{Tr}(\xi_i\alpha_j)$ is one if $i = j$ and zero otherwise. Define

$$\tau : \mathrm{PG}(2,q) \longrightarrow \mathcal{P}_q$$

by $\tau(a_0{:}a_1{:}a_2) = (a_0\alpha_0 + a_1\alpha_1 + a_2\alpha_2)^{q^2-q}$. Then τ is well defined and a bijection. This is proved in the same way as done for σ, since q is relatively prime to $q^2 + q + 1$.

Let $(x_0{:}x_1{:}x_2)$ be a point P of $\mathrm{PG}(2,q)$ and let $(a_0{:}a_1{:}a_2)$ represent a line B of $\mathrm{PG}(2,q)$. Let $u = x_0\xi_0 + x_1\xi_1 + x_2\xi_2$ and $v = a_0\alpha_0 + a_1\alpha_1 + a_2\alpha_2$. Then P is incident with B if and only if

$$\mathrm{Tr}(uv) \;=\; \sum_{i,j} a_ix_j\mathrm{Tr}(\xi_i\alpha_j) \;=\; a_0x_0 + a_1x_1 + a_2x_2 \;=\; 0$$

Let $s = \sigma(x_0{:}x_1{:}x_2) = u^{q-1}$ and $t = \tau(a_0{:}a_1{:}a_2) = v^{q^2-q}$. Then u and v are not zero. Thus $\mathrm{Tr}(uv) = 0$ if and only if

$$0 \;=\; \frac{\mathrm{Tr}(uv)}{uv} \;=\; u^{q^2-1}v^{q^2-1} + u^{q-1}v^{q-1} + 1$$

Dividing by v^{q^2-1} gives

$$0 = (u^{q-1})^{q+1} + u^{q-1}v^{-q^2+q} + (v^{q-1})^{-q-1}$$

We have that $s = u^{q-1}$ and $t = v^{q^2-q}$. Thus $t^q = (v^{q-1})^{q^2} = (v^{q-1})^{-q-1}$, since $v^{q-1} \in \mathcal{P}_q$. Hence

$$s^{q+1} + st^{-1} + t^q = 0.$$

Therefore $0 = s^{q+1}t + s + t^{q+1}$, that is to say $F(s,t) = 0$. It follows that P is incident with B if and only if $\sigma(P)$ is incident with $\tau(B)$. ■

Remark 1. Let $\mathrm{Tr} : \mathbb{F}_q^{m+1} \to \mathbb{F}_q$ be the trace map. Then two bases $\xi_0, \ldots, \xi_m$ and $\alpha_0, \ldots, \alpha_m$ of $\mathbb{F}_q^{m+1}$ over $\mathbb{F}_q$ are called *dual* if $\mathrm{Tr}(\xi_i\alpha_j)$ is one if $i = j$ and zero otherwise. A basis is called *self-dual* if it is dual to itself.
Using the classification of symmetric bilinear forms over finite fields and some Galois theory one can show the following facts. If q is even, then a self dual basis always exists. If q is odd, then $\mathbb{F}_q^{m+1}$ has a dual basis over $\mathbb{F}_q$ if and only if m is even. Hence $\mathbb{F}_{q^s}$ has a self dual basis over $\mathbb{F}_q$ for every q ◇

Example 1. Let $\alpha \in \mathbb{F}_8$ be such that $\alpha^3 + \alpha + 1 = 0$. Then α^3, α^5, α^6 is a self-dual basis for $\mathbb{F}_8$ over $\mathbb{F}_2$. Choose for ξ_0, ξ_1, ξ_2 and α_0, α_1, α_2 this self-dual basis. The following table:

P	$\sigma(P)$							B	$\tau(B)$
(1:0:0)	α^3					$=$	α^3	$X = 0$	α^6
(0:1:0)			α^5			$=$	α^5	$Y = 0$	α^3
(0:0:1)					α^6	$=$	α^6	$Z = 0$	α^5
(1:1:0)	α^3	$+$	α^5			$=$	α^2	$X + Y = 0$	α^4
(1:0:1)	α^3		$+$		α^6	$=$	α^4	$X + Z = 0$	α^1
(0:1:1)			α^5	$+$	α^6	$=$	α^1	$Y + Z = 0$	α^2
(1:1:1)	α^3	$+$	α^5	$+$	α^6	$=$	α^0	$X + Y + Z = 0$	α^0

shows the isomorphisms σ and τ in the proof of Proposition 3 for this case. One verifies that this is in agreement with Figure 1. □

3. The number of rational points of $\mathcal{X}_q$

Consider the *Hurwitz* curve with defining equation

$$X^mY + Y^mZ + Z^mX = 0$$

It is not difficult to show that his curve is nonsingular over a field of characteristic p if and only if p is relatively prime to $m^2 - m + 1$. See [**14**, Example 2.14]. Thus the curve $\mathcal{X}_q$ in (1) is nonsingular over $\mathbb{F}_q$, and therefore absolutely irreducible. Hence $\mathcal{X}_q$ has genus $(q+1)q/2$.

The set $\{(x,y) \in \mathcal{P}_q^2 : F_q(x,y) = 0\}$ is a subset of $\mathcal{X}_q(\mathbb{F}_{q^3})$. So $\mathcal{X}_q$ has at least $(q+1)(q^2+q+1)$ points with nonzero coordinates in $\mathbb{F}_{q^3}$. We will see that these are not the only ones.

Let $\mathcal{F}_q$ be the *Fermat* curve with defining equation

$$U^{q^2+q+1} + V^{q^2+q+1} + W^{q^2+q+1} = 0$$

over $\mathbb{F}_q$. Let $(u{:}v{:}w)$ be a point of $\mathcal{F}_q$. Further let $x = u^{q+1}w$, $y = v^{q+1}u$ and $z = w^{q+1}v$. Then $x^{q+1}y + y^{q+1}z + z^{q+1}x$ is equal to

$$u^{q^2+2q+2}v^{q+1}w^{q+1} + u^{q+1}v^{q^2+2q+2}w^{q+1} + u^{q+1}v^{q+1}w^{q^2+2q+2}$$

$$= u^{q+1}v^{q+1}w^{q+1}(u^{q^2+q+1} + v^{q^2+q} + w^{q^2+q+1}) = 0$$

So $(x{:}y{:}z)$ is a point of $\mathcal{X}_q$. Define $\varphi(u{:}v{:}w) = (u^{q+1}w{:}v^{q+1}u{:}w^{q+1}v)$. Then

$$\varphi : \mathcal{F}_q \longrightarrow \mathcal{X}_q$$

is a morphism of curves. An $\mathbb{F}_{q^3}$-rational point of $\mathcal{F}_q$ gives under the mapping φ an $\mathbb{F}_{q^3}$-rational point of $\mathcal{X}_q$.

Proposition 4. *The curve $\mathcal{F}_q$ has exactly $(q^2+q+1)(q+1)(q-1)^2$ points that are $\mathbb{F}_{q^3}$-rational.*

Proof. Notice that x^{q^2+q+1} is an element of $\mathbb{F}_q$ for every $x \in \mathbb{F}_{q^3}$, and for every nonzero $a \in \mathbb{F}_q^*$ there are exactly q^2+q+1 solutions of $x^{q^2+q+1} = a$ with $x \in \mathbb{F}_{q^3}$. Hence $\mathcal{F}_q$ intersects the line at inifinity, with equation $W = 0$, in exactly q^2+q+1 points, and these points are $\mathbb{F}_{q^3}$-rational.

Consider the affine equation $U^{q^2+q+1} + V^{q^2+q+1} + 1 = 0$. There are q^2+q+1 solutions of the equation $v^{q^2+q+1} + 1 = 0$ with $v \in \mathbb{F}_{q^3}$, corresponding to $\mathbb{F}_{q^3}$-rational points of the form $(0,v)$.

If for a given $v \in \mathbb{F}_{q^3}$ we have $v^{q^2+q+1} + 1 \neq 0$, then $v^{q^2+q+1} + 1 \in \mathbb{F}_q^*$ and there are q^2+q+1 solutions of $u^{q^2+q+1} + v^{q^2+q+1} + 1 = 0$ with $u \in \mathbb{F}_{q^3}$. Hence we have in total

$$2(q^2+q+1) + (q^2+q+1)[q^3 - (q^2+q+1)] = (q^2+q+1)(q^3-q^2-q+1)$$

$\mathbb{F}_{q^3}$-rational points on $\mathcal{F}_q$, and this number is equal to $(q^2+q+1)(q+1)(q-1)^2$. ∎

Remark 2. The points of $\mathcal{F}_q$ on the line with equation $W = 0$ are mapped under φ to (0:1:0). The points of $\mathcal{F}_q$ on the line with equation $V = 0$ are mapped under φ to (1:0:0). The points of $\mathcal{F}_q$ on the line with equation $U = 0$ are mapped under φ to (0:0:1). In affine coordinates, the map is defined by $\varphi(u,v) = (x,y)$, where $x = u^{q+1}v^{-1}$ and $y = uv^q$. Thus $u^{q^2+q+1} = x^q y$ and $v = u^{q+1}x^{-1}$. Hence $x^q y \in \mathbb{F}_q$ if $u \in \mathbb{F}_{q^3}$, and conversely if $x^q y \in \mathbb{F}_q$ and $x \in \mathcal{P}_q$, there are q^2+q+1 points $(u,v) \in \mathcal{F}_q(\mathbb{F}_{q^3})$ such that $\varphi(u,v) = (x,y)$.

Hence for every given point $(x{:}y{:}z)$ of $\mathcal{X}_q$ there are exactly q^2+q+1 points in the inverse image of $(x{:}y{:}z)$ under φ. Hence φ is unramified and has degree q^2+q+1. Moreover the points of $\varphi^{-1}(x{:}y{:}z)$ are all $\mathbb{F}_{q^3}$-rational if $(x{:}y{:}z)$ is in the image of $\mathcal{F}_q(\mathbb{F}_{q^3})$ under φ. ◇

Lemma 5. *Let $x,y \in \mathbb{F}_{q^3}$ be such that $F_q(x,y)=0$, and let $t=x^q y$. Then, if $y \in \mathcal{P}_q$, then $t^{q+1}+t+1=0$, otherwise $t \in \mathbb{F}_q$.*

Proof. Suppose $x,y \in \mathbb{F}_{q^3}$ and $x^{q+1}y+y^{q+1}+x=0$. Raising to the q-th power gives $x^{q^2+q}y^q+y^{q^2+q}+x^q=0$. Multiplying with y gives

$$(x^q y)^{q+1}+y^{q^2+q+1}+(x^q y) \;=\; 0$$

Let $t=x^q y$. Let $a=y^{q^2+q+1}$. Then $t^{q+1}+t+a=0$. Now $a^{q-1}=y^{q^3-1}$ is zero or one. Thus $a \in \mathbb{F}_q$. If $y \in \mathcal{P}_q$, then $a=1$, so $t^{q+1}+t+1=0$. Otherwise, raise $t^{q+1}+t+a=0$ to the q-th power. This gives $t^{q^2+q}+t^q+a=0$. Multiplying with t gives $t^{q^2+q+1}+t^{q+1}+at=0$. Let $b=t^{q^2+q+1}$. Then $b \in \mathbb{F}_q$. Thus $b+(-t-a)+at=0$. Hence $t=(a-b)/(a-1) \in \mathbb{F}_q$. ■

In what follows, we let ε_q denote the remainder in $\{0,1,2\}$ of $q+1$ modulo 3.

Lemma 6. *The set $\varphi(\mathcal{F}_q(\mathbb{F}_{q^3})) \cap \mathcal{P}_q^2$ has $\varepsilon_q(q^2+q+1)$ points.*

Proof. If $(x,y) \in \varphi(\mathcal{F}_q(\mathbb{F}_{q^3})) \cap \mathcal{P}_q^2$, then there exists a $(u,v) \in \mathcal{F}_q(\mathbb{F}_{q^3})$ such that $\varphi(u,v)=(x,y) \in \mathcal{P}_q^2$. So $x=u^{q+1}v^{-1}$ and $y=uv^q$. Hence $u^{q^2+q+1}=x^q y$. Let $t=x^q y$. Then $t \in \mathbb{F}_q^*$, and $t^{q+1}+t+1=0$ by Lemma 5. Hence $t^2+t+1=0$ and $y=tx^{-q}$ is determined by $x \in \mathcal{P}_q$ and t.

(i). If q is a power of 3, then $(t-1)^2=t^2+t+1=0$. Thus $t=1$.
(ii). If q is not divisible by 3, then $t^3-1=(t-1)(t^2+t+1)=0$. So $t^3=1$ and $t \neq 1$. Hence the order of t in $\mathbb{F}_q^*$ is 3. Thus 3 divides $q-1$, and there are two zeros t_1 and t_2 in $\mathbb{F}_q^*$ of $t^2+t+1=0$.

Therefore the number of zeros $t^2+t+1=0$ in $\mathbb{F}_q$ is equal to ε_q, and there are at most $\varepsilon_q(q^2+q+1)$ points in $\varphi(\mathcal{F}_q(\mathbb{F}_{q^3})) \cap \mathcal{P}_q^2$. Conversely, every choice of $x \in \mathcal{P}_q$ and a zero of $\mathbb{F}_q$ $t^2+t+1=0$ gives a point (x,tx^{-q}) in $\varphi(\mathcal{F}_q(\mathbb{F}_{q^3})) \cap \mathcal{P}_q^2$ by Remark 2. Therefore this set has $\varepsilon_q(q^2+q+1)$ points. ■

Theorem 7. *The number of $\mathbb{F}_{q^3}$-rational points of $\mathcal{X}_q$ is equal to*

$$2q^3+1+(1-\varepsilon_q)(q^2+q+1)$$

Proof. Lemma 5 and Remark 2 imply that $\mathcal{X}_q(\mathbb{F}_{q^3})$ is the union of $\varphi(\mathcal{F}_q(\mathbb{F}_{q^3}))$ and $\mathcal{X}_q(\mathbb{F}_{q^3}) \cap \mathcal{P}_q^2$. The set $\varphi(\mathcal{F}_q(\mathbb{F}_{q^3}))$ has $(q+1)(q-1)^2$ elements, since $\mathcal{F}_q(\mathbb{F}_{q^3})$ has $(q^2+q+1)(q+1)(q-1)^2$ elements, by Proposition 4, while φ has degree q^2+q+1 and the points of $\varphi^{-1}(x{:}y{:}z)$ are all $\mathbb{F}_{q^3}$-rational if $(x{:}y{:}z)$ is in the image of $\mathcal{F}_q(\mathbb{F}_{q^3})$ under φ by Remark 2.

The set $\mathcal{X}_q(\mathbb{F}_{q^3}) \cap \mathcal{P}_q^2$ has $(q^2+q+1)(q+1)$ points by Proposition 2. Hence $\mathcal{X}_q(\mathbb{F}_{q^3})$ has

$$(q+1)(q-1)^2 + (q^2+q+1)(q+1) - \varepsilon_q(q^2+q+1)$$

points by the inclusion/exclusion principle and Lemma 6, which is equal to the desired number. ∎

Remark 3. It was already noticed that $\mathcal{X}_2$, the Klein quartic over $\mathbb{F}_8$, has 24 rational points, and this is optimal for a curve of genus 3. Let $N_q(g)$ be the maixmal number of rational points of a curve of genus g over $\mathbb{F}_q$. Thus $N_8(3) = 24$.

Now $\mathcal{X}_3$ has genus 6 and 55 rational points over $\mathbb{F}_{27}$ by Theorem 7, but it is shown in [11] that $76 \leq N_{27}(6) \leq 88$. The curve $\mathcal{X}_4$ has genus 10 and 108 rational points over $\mathbb{F}_{64}$ by Theorem 7, but $193 \leq N_{64}(10) \leq 225$ by [12]. Hence the number of rational points of $\mathcal{X}_q$ over $\mathbb{F}_{q^3}$ is far from being optimal if $q > 2$.

◇

4. A general construction

In this section the constuction in [18] is discussed. There it was used to get plane curves with many rational points. Here it will be related to the question how to get curves representing symmetric designs, in particular projective planes. Finally a polynomial is given that represents the projective geometry $\mathrm{PG}(m,q)$.

Let $f(T)$ be a univariate polynomial with coefficients in $\mathbb{F}_q$ of degree k. Let $\mathcal{P}$ be a cyclic subgroup of $\mathbb{F}_{q^e}^*$ of order v. Suppose that $f(T)$ has k distinct zeros in $\mathcal{P}$. Let s and t be nonnegative integers. Consider the bivariate polynomial $F(X,Y)$ that is obtained from $X^t f(X^s Y)$ by reducing modulo v the exponents i and j of a monomial $X^i Y^j$ that appears in $X^t f(X^s Y)$.

If s is relatively prime to v, then $(\mathcal{P}, \mathcal{P}, F)$ a 1-(v,k,k) design. An example of this construction has been given in § 2, which gives, in fact, a 2-design with $f(T) = f_q(T) = T^{q+1} + T + 1$ over $\mathbb{F}_q$, $e = 3$, $\mathcal{P} = \mathcal{P}_q$, $k = q+1$, $v = q^2+q+1$, $s = q$ and $t = 1$, and where $F(X,Y)$ is the reduction of $X f_q(X^q Y)$. This is a variation of the following two examples from [18].

Example 2. Let $f(T) = T^{q^2-1} - T^{q-1} + 1$, $e = 3$, $\mathcal{P} = \mathbb{F}_{q^3}^*$, $k = q^2 - 1$, $v = q^3 - 1$, $s = q$ and $t = q - 1$. Then

$$F(X,Y) = X^{q-1} - X^{q^2-1}Y^{q-1} + Y^{q^2-1}$$

and $(\mathcal{P}, \mathcal{P}, F)$ represents a 1-(q^3-1, q^2-1, q^2-1) design. The first member of this family of curves $(q = 2)$ is again the Klein quartic.

If $x_1, x_2 \in \mathcal{P}$ and $x_1 \neq x_2$, then $F(x_1, Y) - F(x_2, Y)$ has degree $q-1$ in Y. In order that the design is a 2-$(q^3-1, q^2-1, q-1)$ design one should have that $bk(k-1) = \lambda v(v-1)$. Thus $(q^2-1)(q^2-2) = (q-1)(q^3-2)$, so $q = 2$. □

Example 3. Let $f(T) = T^q + T + 1$, $e = 2$, $\mathcal{P} = \mathbb{F}_{q^2}^*$, $k = q$, $v = q^2 - 1$, $s = q$ and $t = 0$. Then $F(X,Y) = XY^q + X^qY + 1$ and $(\mathcal{P}, \mathcal{P}, F)$ represents a 1-$(q^2 - 1, q, q)$ design. The corresponding homogeneous polynomial gives

$$XY^q + X^qY + Z^{q+1} = 0$$

defining the *Hermitian curve*. So $Y^q + Y + Z^{q+1} = 0$ is an affine equation of the Hermitian curve. The analogous proof of Proposition 2 that it is a 2-design breaks down, since the counting argument at the end fails. One should have the parameters of a projective plane of order $k - 1$, but $v \neq k^2 - k + 1$. □

These examples are interesting from the point of view of curves with many rational points, but they do not represent 2-designs if $q > 2$.

In order to get more examples of projective planes of order n represented by a polynomial, one might apply the construction to trinomials of the form $f_{n,\alpha,\beta}(T) = T^{n+1} + \alpha T + \beta$ with coefficients in $\mathbb{F}_q^*$. Let $v = n^2 + n + 1$. Let $\mathcal{P}_n = \{x \in \mathbb{F}_{q^e} : x^v = 1\}$. Let $F_{n,\alpha,\beta}(X,Y)$ be the reduction of $Xf_{n,\alpha,\beta}(X^nY)$. Then

$$F_{n,\alpha,\beta}(X,Y) = \alpha X^nY + Y^{n+1} + \beta X.$$

The following proposition says that only the Desarguesian projective planes are obtained in this way.

Theorem 8. *Let p be the characteristic of $\mathbb{F}_q$. Suppose that $f_{n,\alpha,\beta}(T)$ has $n+1$ zeros in $\mathcal{P}_n$ and $\mathcal{P}_n$ has $n^2 + n + 1$ elements. Then the triple $(\mathcal{P}_n, \mathcal{P}_n, F_{n,\alpha,\beta})$ represents a projective plane of order n if and only if n is a power of p, $\beta \in \mathcal{P}_n$ and $\alpha = \beta^{n+1}$. Moreover, all the projective planes obtained in this way are isomorphic to* PG$(2,n)$.

Proof. The proof that the conditions are sufficient to get a representation of a projective plane of order n is similar to the proof of Proposition 2. Let t be a zero of $f_{n,\alpha,\beta}(T)$. Then $-t^{n+1} = \alpha t + \beta$. So

$$(-1)^n t^{n^2+n+1} = (\alpha t + \beta)^n t = \sum_{i=0}^{n} \binom{n}{i} \alpha^i \beta^{n-i} t^{i+1}$$

Now $t^{n^2+n+1} = 1$, since $t \in \mathcal{P}_n$. Substitute $-\alpha t - \beta$ for t^{n+1}. Then

$$0 = ((-1)^{n+1} - \alpha^n\beta) + (\beta^n - \alpha^{n+1})t + \sum_{i=1}^{n-1} \binom{n}{i} \alpha^i \beta^{n-i} t^{i+1}$$

This equation has degree n in t and has $n + 1$ solutions. So all coefficients are zero. Hence $\alpha^n\beta = (-1)^{n+1}$, $\alpha^{n+1} = \beta^n$ and $\binom{n}{i} = 0$ in $\mathbb{F}_q$ for all $0 < i < n$. Thus $\binom{n}{i} \equiv 0 \pmod{p}$ for all $i = 1, 2, \ldots, n$. Therefore n is a power of p by [**21**, Lemma 4.16]. So $\alpha^n\beta = (-1)^{n+1} = 1$ in $\mathbb{F}_n$. Hence

$$\alpha = \alpha\alpha^n\beta = \alpha^{n+1}\beta = \beta^n\beta = \beta^{n+1},$$

and $\beta^{n^2+n+1} = (\beta^{n+1})^n\beta = \alpha^n\beta = 1$. Therefore $\beta \in \mathcal{P}_n$. The isomorphism is obtained by noticing that if $t^{n+1} + t + 1 = 0$ and $\beta \in \mathcal{P}_n$, then $\beta^{-n}t$ is a zero of $T^{n+1} + \beta^{n+1}T + \beta$. ∎

The *projective geometry* $\mathrm{PG}(m,q)$ is the design, where a point P of $\mathrm{PG}(m,q)$ is a 1-dimensional linear subspace of $\mathbb{F}_q^{m+1}$. A block B of $\mathrm{PG}(m,q)$, also called a *hyperplane*, is an m-dimensional linear subspace of $\mathbb{F}_q^{m+1}$. The point P is incident with B if and only if $P \subset B$.

Let $p(m,q)$ be the number of points of $\mathrm{PG}(m,q)$. Define $p(-1,q) = 0$. Then

$$p(m,q) = \frac{q^{m+1}-1}{q-1}$$

and $\mathrm{PG}(m,q)$ is a 2-$(p(m,q), p(m-1,q), p(m-2,q))$ design. Consider the polynomial

$$f_{m,q}(T) = \sum_{i=0}^{m} T^{p(i-1,q)}.$$

over $\mathbb{F}_q$. Let $\mathcal{P}_{m,q} = \{ x \in \mathbb{F}_{q^{m+1}} : x^{p(m,q)} = 1 \}$. Apply the construction to $f_{m,q}(T)$ with $e = m+1$, $\mathcal{P} = \mathcal{P}_{m,q}$, $k = p(m-1,q)$, $v = p(m,q)$, $s = q$ and $t = 1$. Then the substitution $T = X^qY$, and the reduction of $X^{p(m,q)}$ to 1 in $Xf_{m,q}(X^qY)$ gives $F_{m,q}(X,Y)$ where

$$F_{m,q}(X,Y) = \sum_{i=0}^{m-1} X^{p(i,q)}Y^{p(i-1,q)} + Y^{p(m-1,q)}.$$

Theorem 9. *The triple $(\mathcal{P}_{m,q}, \mathcal{P}_{m,q}, F_{m,q})$ represents a design with parameters 2-$(p(m,q), p(m-1,q), p(m-2,q))$, which is isomorphic to the points and hyperplanes of the projective geometry* $\mathrm{PG}(m,q)$.

Proof. Choose dual bases $\xi_0, \ldots, \xi_m$ and $\alpha_0, \ldots, \alpha_m$ of $\mathbb{F}_{q^{m+1}}$. Now define $\sigma(x_0 : \cdots : x_m) = (\sum x_i\xi_i)^{q-1}$, where $(x_0 : \cdots : x_m)$ represents a point of the projective geometry $\mathrm{PG}(m,q)$, that is the line $\{\lambda \sum x_i\xi_i : \lambda \in \mathbb{F}_q\}$ in $\mathbb{F}_{q^{m+1}}$. Define $\tau(a_0 : \cdots : a_m) = (\sum a_i\alpha_i)^{q^m - q^{m-1}}$, where $(a_0 : \cdots : a_m)$ represents the hyperplane of $\mathrm{PG}(m,q)$ with defining equation $\sum a_ix_i = 0$.

The proof is now similar to the proof of Proposition 3. The proof that it is a 2-design is now a consequence of the isomorphism with $\mathrm{PG}(m,q)$. ∎

References

[1] S. BALL, A. BLOKHUIS, AND C. O'KEEFE, A representation of $\mathrm{PG}(2,q)$ in $\mathrm{GF}(q^3)$, unpublished manuscript, May 1997.

[2] R.E. BLAHUT, Algebraic-geometric codes without algebraic geometry, in *Proc. IEEE Inform. Theory Workshop*, Salvador, Brazil, June 1992.
[3] ———, *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley, 1983.
[4] ———, Algebraic coding theory in one and two dimensions, lectures given at the Eindhoven University of Technology, August 20, 21 and 22, 1994.
[5] P.J. CAMERON AND J.H. VAN LINT, *Designs, Graphs, Codes and their Links*, Cambridge, UK: Cambridge University Press, 1991.
[6] P. DEMBOWSKI, *Finite Geometries*, Berlin: Springer-Verlag 1968.
[7] G.-L. FENG AND T.R.N. RAO, Decoding of algebraic geometric codes up to the designed minimum distance, *IEEE Trans. Inform. Theory*, vol. 39, pp. 37–45, 1993.
[8] ———, A simple approach for construction of algebraic-geometric codes from affine plane curves, *IEEE Trans. Inform. Theory*, vol. 40, pp. 1003–1012, 1994.
[9] ———, Improved geometric Goppa codes, Part I: Basic theory, *IEEE Trans. Inform. Theory*, vol. 41, pp. 1678–1693, 1995.
[10] G.-L. FENG, V.K. WEI, T.R.N. RAO, AND K.K. TZENG, Simplified understanding and efficient decoding of a class of algebraic-geometric codes, *IEEE Trans. Inform. Theory*, vol. 40, pp. 981–1002, 1994.
[11] G. VAN DER GEER AND M. VAN DER VLUGT, Quadratic forms, generalized Hamming weights of codes, and curves with many rational points, *J. Number Theory*, vol. 59, pp. 20–36, 1996.
[12] ———, Generalized Hamming weights of codes and curves over finite fields with many rational points, in *Israel Math. Proc.*, vol. 9, pp. 417–432, 1996.
[13] J.P. HANSEN, Codes from the Klein quartic, ideals, and decoding, *IEEE Trans. Inform. Theory*, vol. 33, pp. 923–925, 1987.
[14] T. HØHOLDT, J.H. VAN LINT, AND R. PELLIKAAN, Algebraic geometry codes, in *Handbook of Coding Theory*, V.S. Pless and W.C. Huffman (Editors), Amsterdam: Elsevier, 1998.
[15] D.R. HUGHES AND F.C. PIPER, *Projective Planes*, Berlin: Springer-Verlag 1973.
[16] A. HURWITZ, Über algebraische Gebilde mit eindeutigen Transformationen in sich, *Math. Ann.*, vol. 41, pp. 403–442, 1893.
[17] ———, Über die diophantische Gleichung $x^3y + y^3z + z^3x = 0$, *Math. Ann.*, vol. 65, pp. 428–430, 1908.
[18] J. JUSTESEN, K.J. LARSEN, H. ELBRØND JENSEN, A. HAVEMOSE, AND T. HØHOLDT, Construction and decoding of a class of algebraic geometric codes, *IEEE Trans. Inform. Theory*, vol. 35, pp. 811–821, 1989.
[19] F. KLEIN, Über die Transformation Siebenter Ordnung der Elliptischen Funktionen, *Math. Ann.*, vol. 14, pp. 428–471, 1879.
[20] F.J. MACWILLIAMS AND N.J.A. SLOANE, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland 1977.
[21] J.J. ROTMAN, *An Introduction to the Theory of Groups*, Berlin: Springer-Verlag 1995.
[22] H. STICHTENOTH, *Algebraic Function Fields and Codes*, Berlin: Springer-Verlag 1993.

Improved Hermitian-like Codes over GF(4)

Gui-Liang Feng

CENTER FOR ADVANCED COMPUTER STUDIES
UNIVERSITY OF SOUTHWESTERN LOUISIANA
LAFAYETTE, LA 70504, USA
glf@cacs.usl.edu

Thammavarapu R. N. Rao

CENTER FOR ADVANCED COMPUTER STUDIES
UNIVERSITY OF SOUTHWESTERN LOUISIANA
LAFAYETTE, LA 70504, USA
rao@cacs.usl.edu

ABSTRACT. Hermitian-like codes over GF(4) are introduced. For these Hermitian-like codes, we derive a new lower bound of the minimum distance based on nearly-well-behaving sequences. Applying the new bound, we derive improved Hermitian-like codes. We also prove that the improved Hermitian-like codes over GF(4) exceed the Drinfeld-Vlăduţ bound.

1. Introduction

For the past forty years, finding good codes, especially a sequence of linear codes which meets or exceeds the Gilbert-Varshamov bound, has been a very active area of research. Unfortunately, many known linear codes are asymptotically bad codes. For linear codes, the commonly used notation (n, k, d) means a code of length n, with dimension k and minimum distance d. For such a code, the information rate and relative minimum distance are defined as $R = k/n$ and $\delta = d/n$, respectively. A sequence of linear codes $\{\mathbb{C}_m\}_{m=1}^{\infty}=\{(n_m, k_m, d_m)\}_{m=1}^{\infty}$ over $\mathbb{F}_q$, such that the code length n_m tends to ∞ when $m \to \infty$, is said to be *asymptotically good* if the information rates R_m and also the relative minimum distances δ_m are bounded away from zero. For more than forty years, the best lower bound on R_m for a fixed δ_m has been the *Gilbert-Varshamov* bound.

Supported in part by the National Science Foundation under grant NCR-9505619.

Let $H_q(x)$ be the q-ary entropy function, given by

$$H_q(x) = x \cdot \log_q(q-1) - x \cdot \log_q x - (1-x) \cdot \log_q(1-x)$$

When $q = 2$, the entropy function reduces to the binary entropy function given by the following equation:

$$H_2(x) = -x \cdot \log_2 x - (1-x) \cdot \log_2(1-x).$$

From [**13**, p. 75], we know that $R_{GV}(\delta) = 1 - H_q(\delta)$. This is the *Gilbert-Varshamov curve*. It is also known that $R_{GV}(0) = 1$ and $R_{GV}(q - 1/q) = 0$. The well-known *Gilbert-Varshamov bound* can be described from [**9**, p. 557], as follows.

Theorem 1 (Gilbert-Varshamov bound). *Suppose $0 \leq \delta \leq (q-1)/q$. Then there exists an infinite sequence of (n, k, d) q-ary linear codes with $d/n = \delta$ and rate $R = k/n$ satisfying the inequality:*

$$R \geq 1 - H_q(\delta) \qquad \text{for all } n$$

In [**5**], it was shown that there exist long Goppa codes that meet the Gilbert-Varshamov bound. However, to date, a solution for finding these long Goppa codes has not appeared. The most important development in the theory of error-correcting codes in recent years has been the introduction of methods from algebraic geometric curves for the construction of linear codes. These so-called *algebraic-geometric codes* (AG codes) were introduced by Goppa [**6, 7, 8**].

In 1982, Tsfasman, Vlăduţ and Zink [**14**] obtained an extremely exciting result: they showed the existence of a sequence of AG codes, which exceeds the Gilbert-Varshamov bound. In 1984, Katsman, Tsfasman, and Vlăduţ [**10**] presented a polynomial construction of q-ary codes arising from modular curves $X_0(111)$ over GF(q^2), which are better than the Gilbert-Varshamov bound, when $q \geq 49$. But the construction has a very high complexity. Therefore, a solution for the construction of asymptotically good AG codes which exceed the Gilbert-Varshamov bound, has not yet been found. Pellikaan [**4**] has shown that abundant curves give asymptotically good codes for small relative minimum distance values. They are better than the Tsfasman-Vlăduţ-Zink bound, but are still below the Gilbert-Varshamov bound. Recently, Garcia and Stichtenoth [**11**] calculated the genus g_m of Hermitian-like curves. When $q > 49$, the Hermitian-like codes exceed the Tsfasman-Vlăduţ-Zink bound and the Gilbert-Varshamov bound. Now many researchers are working on an explicit construction of a sequence of codes that exceed the Tsfasman-Vlăduţ-Zink bound [**15, 1, 12, 2**]. In this paper, we present an explicit construction of a sequence of codes over GF(4), whose designed minimum distance is greater than $n - k - g_m + 1$.

The organization of this paper is as follows. In the next section, we briefly review the Hermitian-like curves and the Hermitian-like codes. In § 3, we introduce a new approach to determine a lower bound of minimum distance of algebraic-geometric codes and apply it to the Hermitian-like codes over GF(4). In § 4, we illustrate a class of improved Hermitian-like codes over GF(4), which are asymptotically better than Hermitian-like codes over GF(4). Finally, a few conclusions are presented in § 6.

2. Hermitian-like curves and Hermitian-like codes

It is well known that the Hermitian curve over GF(q^2) is defined by equation

$$Y^{q+1} + X^q + X = 0 \quad (1)$$

This curve has q_3 points or *roots*. For each Y, there are q distinct values of X, such that (Y, X) is a root of (1).

Example 1. Consider the simplest Hermitian curve over GF(2^2), given by:

$$Y^3 + X^2 + X = 0 \quad (2)$$

where $q = 2$. This curve has 8 roots: $(Y, X) = (0,0), (0,1), (1,\alpha), (1,\alpha^2), (\alpha,\alpha), (\alpha,\alpha^2), (\alpha^2,\alpha)$, and (α^2,α^2), where α is the primitive element of GF(2^2). □

A slight change of Hermitian curves gives us Hermitian-like curves. For $Y \neq 0$, let $y = Y$ and $x = X/Y$. That is, $X = xY$. Then (1) becomes

$$y^{q+1} + x^q y^q + xy = 0 \quad (3)$$

Since $Y \neq 0$, we have

$$y^q + x^q y^{q-1} + x = 0 \quad (4)$$

over GF(q^2). The curve in (4) is called a Hermitian-like curve. For each $y \neq 0$, there are q distinct nonzero x's such that all (x, y) are the roots of (4). Thus we have $(q^2-1)q$ roots when $y \neq 0$. If $y = 0$, then obviously (0,0) is a root of (3). Therefore (4) has $n = (q^2 - 1)q + 1$ roots altogether.

Example 2. Consider the simplest Hermitian-like curve over GF(2^2), given by:

$$y^2 + x^2 y + x = 0 \quad (5)$$

where $q = 2$. This curve has 7 roots, namely: $(y, x) = (0,0), (1,\alpha), (1,\alpha^2), (\alpha,\alpha), (\alpha,\alpha^2), (\alpha^2,\alpha)$, and (α^2,α^2). □

Example 3. Consider the simplest Hermitian-like curve over GF(3^2), given by:

$$y^3 + x^3 y^2 + x = 0$$

where $q = 3$. This curve has 25 roots, namely: $(y, x) = (0,0), (\beta^i,\beta^i), (\beta^i,\beta^{i+5}), (\beta^i,\beta^{i+7})$ for $i = 0,2,4,6$, and $(\beta^i,\beta^i), (\beta^i,\beta^{i+1}), (\beta^i,\beta^{i+3})$ for $i = 1,3,5,7$, where β is the primitive element of GF(2^2). □

In order to analyze the asymptotical behavior of codes, the number of roots should be as large as we need. This requirement can be met by extending the curves to high dimensional affine spaces. Equation (4) can be regarded as a Hermitian-like curve in a two-dimensional affine plane. The Hermitian-like

curves in a m-dimensional affine space can be written as a set of equations, with variables $x_1, x_2, \cdots, x_m$, as follows:

$$\begin{cases} x_1^q + x_2^q x_1^{q-1} + x_2 = 0 \\ x_2^q + x_3^q x_2^{q-1} + x_3 = 0 \\ \quad \cdots\cdots \\ x_{m-1}^q + x_m^q x_{m-1}^{q-1} + x_m = 0 \end{cases} \tag{6}$$

over GF(q^2). For each x_i, there are q x_{i+1}'s such that $(\cdots, x_{i+1}, x_i, \cdots)$ are roots of (6), and $(0, 0, \cdots, 0)$ is obviously a root. The number n of roots of this set of equations is therefore $(q^2-1)q^{m-1}+1$.

Example 4. Consider the simplest Hermitian-like curve in an m-dimensional affine space over GF(2^2):

$$\begin{cases} x_1^2 + x_2^2 x_1 + x_2 = 0 \\ x_2^2 + x_3^2 x_2 + x_3 = 0 \\ \quad \cdots\cdots \\ x_{m-1}^2 + x_m^2 x_{m-1} + x_m = 0 \end{cases} \tag{7}$$

It can be easily checked that this curve has $n = 3 \cdot 2^{m-1} + 1$ roots (points), where one is $(0, 0, \ldots, 0)$, the others have nonzero components. □

Example 5. The Hermitian-like curve in an m-dimensional affine space over GF(3^2) is:

$$\begin{cases} x_1^3 + x_2^3 x_1^2 + x_2 = 0 \\ x_2^3 + x_3^3 x_2^2 + x_3 = 0 \\ \quad \cdots\cdots \\ x_{m-1}^3 + x_m^3 x_{m-1}^2 + x_m = 0 \end{cases} \tag{8}$$

It can be easily checked that this curve has $n = 8 \cdot 3^{m-1} + 1$ roots (points). □

The reason that we use Hermitian-like curves instead of Hermitian curves is that the Hermitian curves have too large a genus g compared to the number of roots n. From the viewpoint of algebraic geometry, one must compute the genus g of these curves, and then the minimum distance can be calculated as $r - g + 1$. It was estimated that over GF(2^2), the genus of Hermitian curve in an m-dimensional space is $O((q+1)^m)$. Since the number of roots of this curve is only $O(q^m)$, this bound on the genus does not have a practical meaning. However, Garcia and Stichtenoth [11] have shown that the genus g_m of the Hermitian-like curves is given by:

$$g_m = \begin{cases} q^m + q^{m-1} - q^{\frac{m+1}{2}} - 2q^{\frac{m-1}{2}+1} & \text{if } m \text{ is odd,} \\ q^m + q^{m-1} - \frac{1}{2}q^{\frac{m}{2}+1} - \frac{3}{2}q^{\frac{m}{2}} - q^{\frac{m-2}{2}} + 1 & \text{if } m \text{ is even.} \end{cases} \tag{9}$$

Let $d = r - g_m + 1$. This value is called the *Garcia-Stichtenoth bound* on the minimum distance. Our main result herein is an explicit construction of a sequence of codes over GF(4) which exceed the Garcia-Stichtenoth bound for all m.

Next we will construct Hermitian-like codes. In our previous papers [**2, 3**], we presented improved algebraic-geometric codes. We now briefly review this code construction.

Let $\mathcal{LS} = \{P_1, P_2, \ldots, P_n\}$ be a set of points in an m-dimensional affine space, which is usually the set of roots of an m-dimensional curve. For a given location set $\mathcal{LS}$, each monomial $f = x_m^{i_m} x_{m-1}^{i_{m-1}} \cdots x_2^{i_2} x_1^{i_1}$ is associated with an n-tuple vector $(f(P_1), f(P_2), \cdots, f(P_n))$. This vector is called an *evaluated vector* of monomial f at $\mathcal{LS}$. In the subsequent discussion, the evaluated vector and monomial (or polynomial) are used interchangeably. The monomials are linearly independent, which means that the corresponding vectors of these monomials at $\mathcal{LS}$ are linearly independent. Given a location set $\mathcal{LS}$ and a sequence H of monomials, which are linearly independent, we can build a matrix whose i-th row is the vector $\boldsymbol{h}_i = (\boldsymbol{h}_i(P_1), \boldsymbol{h}_i(P_2), \cdots, \boldsymbol{h}_i(P_n))$. Any r rows of this matrix can be used as a parity-check matrix. The resulting matrix defines a linear code, called an improved AG code.

Example 6. Let us consider the curve in Example 2. The location set for this curve is given by:

$$\mathcal{LS} = \{(0,0), (1,\alpha), (1,\alpha^2), (\alpha,\alpha), (\alpha,\alpha^2), (\alpha^2,\alpha), (\alpha^2,\alpha^2)\}$$

We can find an $H = \{1, x, y, xy, y^2, xy^2, y^3\}$. The corresponding evaluated vectors are:

$$\begin{array}{rcccccccc}
1 & \longleftrightarrow & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
x & \longleftrightarrow & 0 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha^2 \\
y & \longleftrightarrow & 0 & 1 & 1 & \alpha & \alpha & \alpha^2 & \alpha^2 \\
xy & \longleftrightarrow & 0 & \alpha & \alpha^2 & \alpha^2 & 1 & 1 & \alpha^2 \\
y^2 & \longleftrightarrow & 0 & 1 & 1 & \alpha^2 & \alpha^2 & \alpha & \alpha \\
xy^2 & \longleftrightarrow & 0 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 \\
y^3 & \longleftrightarrow & 0 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

□

We also derive a new lower bound of the minimal distance based on the definition of the *weight* of monomials and the concept of a *well-behaving sequence* $H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n\}$ of monomials at the location set $\mathcal{LS}$. The order of the elements in a well-behaving sequence H is controlled by the weight function. Briefly, each variable x_μ is associated with a positive integer $w(x_\mu)$, called the *weight* of x_μ, and the weight of a monomial is calculated as:

$$w(x_m^{i_m} \cdots x_1^{i_1}) = \sum_{\mu=1}^{m} i_\mu w(x_\mu) \tag{10}$$

The weight of $\boldsymbol{h}_i * \boldsymbol{h}_j$ is given by

$$w(\boldsymbol{h}_i * \boldsymbol{h}_j) = w(\boldsymbol{h}_i) + w(\boldsymbol{h}_j) \tag{11}$$

At the same time, $\boldsymbol{h}_i * \boldsymbol{h}_j$ can be written as the linear combination of $\boldsymbol{h}_\mu$'s, among which $\boldsymbol{h}_k$ has the maximum weight. That is,

$$\boldsymbol{h}_i * \boldsymbol{h}_j = \boldsymbol{h}_k + \sum_{\mu} \boldsymbol{h}_\mu \tag{12}$$

where $w(\boldsymbol{h}_\mu) < w(\boldsymbol{h}_k)$. The linear combination is denoted as $\boldsymbol{h}_{i,j}$. We say that $\boldsymbol{h}_{i,j}$ is consistent with $\boldsymbol{h}_k$ and denote $\boldsymbol{h}_{i,j} \sim \boldsymbol{h}_k$. Obviously, $w(\boldsymbol{h}_{i,j}) = w(\boldsymbol{h}_k)$.

But $w(\boldsymbol{h}_i * \boldsymbol{h}_j)$ may not be equal to $w(\boldsymbol{h}_k)$. There are three cases, according to the relative weight of $\boldsymbol{h}_i * \boldsymbol{h}_j$ and $\boldsymbol{h}_k$, as follows:

(1) If $w(\boldsymbol{h}_i * \boldsymbol{h}_j) = w(\boldsymbol{h}_k)$, then $\boldsymbol{h}_{i,j}$ is called a *well-behaving term.*

(2) If $w(\boldsymbol{h}_i * \boldsymbol{h}_j) > w(\boldsymbol{h}_k)$, then $\boldsymbol{h}_{i,j}$ is called a *not-bad term.*

(3) $w(\boldsymbol{h}_i * \boldsymbol{h}_j) = w(\boldsymbol{h}_k)$, then $\boldsymbol{h}_{i,j}$ is called a *bad term.*

A sequence $H = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n\}$ is called a *well-behaving sequence*, if its weights satisfy $w(\boldsymbol{h}_1) < w(\boldsymbol{h}_2) < \cdots < w(\boldsymbol{h}_n) = W_{\max}$, and for $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ satisfying $w(\boldsymbol{h}_i) + w(\boldsymbol{h}_j) \leq W_{\max}$, $\boldsymbol{h}_{i,j}$ is either a well-behaving term or a not-bad term.

In a well-behaving sequence, lower bound of the rank of the *syndrome matrix* $\mathcal{S}$ can be easily estimated by counting the N numbers for all the terms in H. The number N_r of $\boldsymbol{h}_r$ is defined as the number of $\boldsymbol{h}_{i,j}$ such that $w(\boldsymbol{h}_{i,j}) = w(\boldsymbol{h}_r)$ and $\boldsymbol{h}_{i,j}$ is a well-behaving term. The idea is that, in the syndrome matrix, by assuming $s_i = 0$, when $i \leq r$ and $s_{r+1} \neq 0$, we can count the number of nonzero points along the iso-weight line of $w(\boldsymbol{h}_{r+1})$.

In a well-behaving sequence, for all other points inside such a nonzero point (i,j), that is, all (u,v) with $u \leq i$, $v \leq j$ and $(u,v) < (i,j)$, we have $s_{uv} = 0$. This is because $w(\boldsymbol{h}_{u,v}) \leq w(\boldsymbol{h}_u) + w(\boldsymbol{h}_v) < w(\boldsymbol{h}_i) + w(\boldsymbol{h}_j) = w(\boldsymbol{h}_{r+1})$, which means that (u,v) lies on the iso-weight line whose weight is less than $w(\boldsymbol{h}_{r+1})$.

Therefore, the number of well-behaving terms along the iso-weight curve of value $w(\boldsymbol{h}_{r+1})$ gives N_{r+1}, the lower bound on the rank when $s_{r+1} \neq 0$. The minimum distance of the resulting code is thus at least $\min\{N_k : k > r\}$. In the following we give some examples that explain the above results and concepts.

Example 7. Let us consider the simplest Hermitian curve over $\mathrm{GF}(2^2)$, discussed in Example 1. This curve has 8 points. From [**13, 9**], we define $w(Y) = 2$ and $w(X) = 3$. We have the following linearly independent monomial sequence: $H = \{00, 10, 01, 20, 11, 02, 21, 03\}$, where ab denotes $Y^a X^b$. From (2), we have $Y^4 + YX^2 + YX = 0$. Since $Y^4 = Y$, we have $YX^2 = Y + YX$, so that 12 is a linear combination of 10 and 11. Thus 12 is not enclosed in H. The weight sequence, denoted by W, of H is {0, 2, 3, 4, 5, 6, 7, 9 }. We have the syndrome matrix $\mathcal{S}$ shown below, where ab denotes $s_{i,j}$ if $\boldsymbol{h}_i * \boldsymbol{h}_j = Y^a X^b$.

00	10	01	20	11	02	21	03
10	20	11	30	21	12	21	...
01	11	02	21	12	03	...	
20	30	21	40	31	...		
11	21	12	31	...			
02	12	03	...				
21	31	...					
03	...						

In the above matrix, $31 \sim 03$, $12 \sim 11$, $30 \sim 02$, $40 = 10$. From the definitions, we can observe there isn't any bad term, since 40 and 12 are not-bad terms, 30, 31, 03 and others are well-behaving terms. Thus H is a well-behaving sequence. By calculating, $N(00) = N_1 = 1$, $N(10) = N_2 = 2$, $N(01) = N_3 = 2$,

$N(20) = N_4 = 3$, $N(11) = N_5 = 4$, $N(02) = N_6 = 5$, $N(21) = N_7 = 6$, $N(03) = N_8 = 8$. If we take the first 4 monomials to form a parity-check matrix, that is if $r = 4$, then the code defined by this parity-check matrix has minimum distance of at least 4, because $N_k \geq 4$ for $k > 4$. □

Example 8. Let us consider the simplest Hermitian curve over GF(2^2) discussed in Example 2. This curve has 7 points. From [**2, 3**], we can define $w(x) = 1$ and $w(y) = 2$. We have the following linearly independent monomial sequence: $H = \{00, 10, 01, 11, 02, 12, 03\}$, where ab denotes $x^a y^b$, and where $W = \{0, 1, 2, 3, 4, 5, 6\}$. We have the syndrome matrix $\mathcal{S}$ shown below, where ab denotes $s_{i,j}$ if $\boldsymbol{h}_i * \boldsymbol{h}_j = x^a y^b$.

00	10	01	11	02	12	03
10	20	11	21	12	22	...
01	11	02	12	03	...	
11	21	12	22	...		
02	12	03	...			
12	22	...				
03	...					

In the above matrix, from (5), we have $xy^2 + x^3y + x^2 = 0$. Since x^3y and y have the same evaluated vector, $x^2 = xy^2 + y$. Thus $x^2 \sim xy^2$, so that $20 \sim 12$. However, $w(x) + w(x) = 2 < 5 = w(xy^2)$. Thus 20 is a bad term. The sequence H is not a well-behaving sequence. In the above matrix $22 = 03 + 11$, so that $22 \sim 03$; $21 = 02 + 10$, so that $21 \sim 02$. Thus the other terms are all well-behaving terms. If we choose $\widehat{H} = \{00, 01, 11, 02, 12, 03, 13\}$, then $\widehat{W} = \{0, 2, 3, 4, 5, 6, 7\}$ and we have syndrome matrix $\widehat{\mathcal{S}}$ as follows:

00	01	11	02	12	03	13
01	02	12	03	13	...	
11	12	22	13	...		
02	03	13	...			
12	13	...				
03	...					
13	...					

Since $22 = 03 + 11$, so that $22 \sim 03$, it can be easily checked that there are all well-behaving terms in the above matrix. Thus $\widehat{H}$ is a well-behaving sequence. By calculation, we have $N_1 = N(00) = 1$, $N_2 = N(01) = 2$, $N_3 = N(11) = 2$, $N_4 = N(02) = 3$, $N_5 = N(12) = 4$, $N_6 = N(03) = 5$, $N_7 = N(00) = 6$. □

3. Improved minimum distance bound

It can be seen that the bound obtained by the well-behaving sequence sometimes is not tight enough. Some of the conditions are too strong and can be weakened further, which may increase the estimated lower bound. In this section, we illustrate some possible improvements.

The first observation is that the requirement of a well-behaving sequence is only a sufficient condition but not a necessary condition in estimating the N number. In the well-behaving sequence, it is required that each term $\boldsymbol{h}_{s,t}$ is either a well-behaving term or a not-bad term. That is, each term should satisfy $w(\boldsymbol{h}_{s,t}) \geq w(\boldsymbol{h}_k)$, where $\boldsymbol{h}_{s,t} = \boldsymbol{h}_k + \sum_\mu \boldsymbol{h}_\mu$ and $w(\boldsymbol{h}_\mu) < w(\boldsymbol{h}_k)$. This requirement makes sure when calculating N_{r+1}, that for all (i,j) satisfying $\boldsymbol{h}_{i,j} \in L(r+1)$ and $w(\boldsymbol{h}_{i,j}) = w(\boldsymbol{h}_{r+1})$, we have $s_{ij} \neq 0$, while $s_{uv} = 0$ for all $u \leq i$, $v \leq j$, and $(u,v) < (i,j)$.

We can, however, meet this same goal, provided that $w(\boldsymbol{h}_k) < w(\boldsymbol{h}_{r+1})$, even if $w(\boldsymbol{h}_k)$ may be greater than $w(\boldsymbol{h}_{s,t})$. Therefore, we define the *nearly well-behaving sequences* for r as those $H = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_r, \cdots, \boldsymbol{h}_{\widehat{r}}, \cdots, \boldsymbol{h}_n\}$ whose weights satisfy

$$w(\boldsymbol{h}_1) < w(\boldsymbol{h}_2) < \cdots < w(\boldsymbol{h}_r) < \cdots < w(\boldsymbol{h}_{\widehat{r}}) < \cdots < w(\boldsymbol{h}_n) \quad = \quad W_{\max},$$

and for each $\widehat{r} > r$, if $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ satisfy $w(\boldsymbol{h}_i) + w(\boldsymbol{h}_j) \leq w(\boldsymbol{h}_{\widehat{r}})$, $\boldsymbol{h}_{i,j}$ is a well-behaving term or a not-bad term, or $w(\boldsymbol{h}_k) \leq w(\boldsymbol{h}_{\widehat{r}})$.

Having more freedom in the arrangement of the H sequence, the number of useful points along the iso-weight curve of weight $w(\boldsymbol{h}_{\widehat{r}})$, namely $N_{\widehat{r}}$, will, at least, not decrease.

Example 9. Let us consider the simplest Hermitian-like curve over GF(2^2) discussed in Example 8. We know that $H = \{00, 10, 01, 11, 02, 12, 03\}$ is not a well-behaving sequence. However, it is a nearly well-behaving sequence for $r = 6$. In the first matrix of Example 8, only $02 \sim 12$ is a bad term. But $w(12) < w(03)$. From the above definition, it is a nearly well-behaving sequence for $r = 6$. Thus $N_7 = 7$. Using the nearly well-behaving sequence for $r = 6$, the AG code defined by the first 6 monomials as a parity-check matrix has a minimum distance of at least 7. If the well-behaving sequence $\hat{H}$ in Example 8 is used, then the AG code defined by the first 6 monomials as a parity-check matrix has a minimum distance of at least 6. From this example, we can see the advantage of a nearly well-behaving sequences. □

The second observation concerns the properties of algebraic curves used to construct the codes. In Hermitian-like curves, some apparently different monomials with different weights may actually correspond to the same vector. For example, consider the Hermitian-like codes over GF(q^2). If $a_m, \cdots, a_{i+1}, a_{i-1}, \cdots, a_1$ are not all equal to zero, then $x_m^{a_m} \cdots x_i^{q^2-1} \cdots x_1^{a_1}$ and $x_m^{a_m} \cdots x_i^0 \cdots x_1^{a_1}$ are two identical evaluated vectors having different weights. Therefore, if we substitute one monomial with another identical one but with a different weight, then we can get a new monomial sequence H such that H is a well-behaving or nearly

well-behaving sequence. The N numbers and the bound on minimum distance in the end will be improved, because we can control the substitution such that the number of useful points along the iso-weight curves increases.

For example, in case $q = 2$, H is not a well-behaving sequence. But $10 = 13$ (that is, $x = xy^3$). Hence replacing 10 by 13, we get $\widehat{H}$, where $\widehat{H}$ is a well-behaving sequence. We now consider another example.

Example 10. Let us consider the Hermitian-like curve in a 3-dimensional affine space over GF(2^2). It can be easily checked that H is not a well-behaving sequence. If we replace $100, 010, 110, 101, 111$ by $103, 013, 113, 104, 114$, respectively, then we get a new sequence

$$\widehat{H} = \{000, 001, 011, 002, 102, 012, 112, 003, 103, 013, 113, 104, 114\}$$

and the weight sequence $\widehat{W} = \{0, 4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 17, 19\}$. From the following syndrome matrix:

```
000 001 011 002 102 012 112 003 103 013 113 104 114
001 002 012 003 103 013 113 004 104 014 114
011 012 022 013 113 023 123 014 114
002 003 013 004 104 014 114
102 103 113 104 204 114
012 013 023 014 114
112 113 123 114
003 004 014
103 104 114
013 014
113 114
104
114
```

we can check that $\widehat{H}$ is a well-behaving sequence, and we can calculate all the N numbers. Thus $N_1 = 1$, $N_2 = 2$, $N_3 = 2$, $N_4 = 3$, $N_5 = 2$, $N_6 = 4$, $N_7 = 2$, $N_8 = 5$, $N_9 = 4$, $N_{10} = 6$, $N_{11} = 6$, $N_{12} = 8$, and $N_{13} = 10$. □

Another point to be mentioned is that we need to calculate $N_{\widehat{r}}$ for $r < \widehat{r} \leq n$ and pick the smallest one. When using $N_{\widehat{r}}$ to determine a lower bound on the weight of codeword $\boldsymbol{c}$, we assume that $s_{\widehat{r}}(\boldsymbol{c}) \neq 0$, but $s_i(\boldsymbol{c}) = 0$ for $i < \widehat{r}$. Therefore, to improve $N_{\widehat{r}}$, we can substitute, delete, and rearrange the monomials in the sequence before $\boldsymbol{h}_{\widehat{r}}$ and keep $\boldsymbol{h}_j$ in some order for $\widehat{r} < j \leq n$ such that we get two nearly well-behaving sequences for $\hat{r}$, namely $RH_{\widehat{r}}$ and $CH_{\widehat{r}}$. Then, the first row and the first column of a new syndrome matrix $\widetilde{\mathcal{S}}_{\widehat{r}}$ correspond to $RH_{\widehat{r}}$ and $CH_{\widehat{r}}$, respectively. From the matrix $\widetilde{\mathcal{S}}_{\widehat{r}}$, we can calculate the $N_{\widehat{r}}$ number.

The $N_{\widehat{r}}$ number will be improved because we can control the selection of $RH_{\widehat{r}}$ and $CH_{\widehat{r}}$ such that the number of useful points along the iso-weight curves increases. Therefore, the third improvement is that the choices of $RH_{\widehat{r}}$ and $CH_{\widehat{r}}$ for different $\widehat{r}$ are independent of each other. Combining these improvements, we indeed get a better estimation of the minimum distance.

Example 11. Let us consider the Hermitian-like curve in 4-dimensional affine space over GF(2^2):

$$\begin{cases} x_1^2 + x_2^2 x_1 + x_2 = 0 \\ x_2^2 + x_3^2 x_2 + x_3 = 0 \\ x_3^2 + x_4^2 x_3 + x_4 = 0 \end{cases}$$

We have $w(x_4) = 1$, $w(x_3) = 2$, $w(x_2) = 4$, $w(x_1) = 8$. Using the substitution technique, we can find the following well-behaving sequence:

$$H = \{0000, 0001, 0011, 0002, 0102, 0012, 1012, 0112, 1112, 0003, 1003, 0103, 1103, 0013, 1013, 0113, 1113, 1004, 0104, 1104, 1014, 0114, 1114, 1005, 1105\}$$

where $abcd$ denotes $x_4^a x_3^b x_2^c x_1^d$. The corresponding weight sequence is given by:

$$W = \{0, 8, 12, 16, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 39, 41, 43\}$$

and $W_{\max} = 43$. The fist 17 columns of the resulting syndrome matrix are shown below:

0000 0001 0011 0002 0102 0012 1012 0112 1112 0003 1003 0103 1103 0013 1013 0113 1113
0001 0002 0012 0003 0103 0013 1013 0113 1113 0004 1004 0104 1104 0014 1014 0114 1114
0011 0012 0022 0013 0113 0023 1023 0123 1123 0014 1014 0114 1114 0024 1024 0124 1124
0002 0003 0013 0004 0104 0014 1014 0114 1114 0005 1005 0105 1105
0102 0103 0113 0104 0204 0114 1114 0214 1214 0105 1105
0012 0013 0023 0014 0114 0024 1024 0124 1124
1012 1013 1023 1014 1114 1024 2024 1124
0112 0113 0123 0114 0214 0124 1124
1112 1113 1123 1114 1214 1124
0003 0004 0014 0005 0105
1003 1004 1014 1005 1105
0103 0104 0114 0105
1103 1104 1114 1105
0013 0014 0024
1013 1014 1024
0113 0114 0124
1113 1114 1124
1004 1005
0104 0105
1104 1105
0114
1114
1005
1105

The remaining columns have 1004, 0104, 1104, 1014, 0114, 1114, 1005, 1105 in the first row and 1005, 0105, 1105 in the second row, with all other rows being zero. From this matrix, we compute N_1=1, N_2=2, N_3=2, N_4=3, N_5=2, N_6=4, N_7=2, N_8=2, N_9=2, N_{10}=5, N_{11}=2, N_{12}=4, N_{13}=2, N_{14}=6, N_{15}=4, N_{16}=6, N_{17}=4, N_{18}=5, N_{19}=8, N_{20}=6, N_{21}=7, N_{22}=10, N_{23}=10, N_{24}=12, N_{25}=14. □

For this well-behaving sequence, the genus is $g = 15$, which is also equal to the number of missing weights. Let the code $\mathbb{C}$ be defined by a parity-check matrix consisting of the first 23 monomials. Then, using the Riemann-Roch theorem, the minimum distance of $\mathbb{C}$ is at least $23 - 15 + 1 = 9$. Using the N number bound, the minimum distance is of $\mathbb{C}$ is at least $\min\{12, 14\} = 12$.

If the above technique is used to find as large as possible N_{24} and N_{25}, then we can get an even tighter bound on the minimum distance. For example, for the monomial 1005, we get two sequences:

$$RH_{24} = \{0000,0001,0101,1101,0011,0002,0102,0012,1012,1112,1003,1013,1113,1005,\cdots\}$$

$$CH_{24} = \{0000,0101,0011,0002,0102,0012,1012,1112,1003,1013,0113,1113,1004,1005,\cdots\}$$

The corresponding syndrome matrix $\widetilde{S}_{24}$ is given by:

$$\begin{array}{l}
0000\ 0101\ 0011\ 0002\ 0102\ 0012\ 1012\ 1112\ 1003\ 1013\ 0113\ 1113\ 1004\ 1005\ \cdots \\
0001\ 0102\ 0012\ 0003\ 0103\ 0013\ 1013\ 1113\ 1004\ 1014\ 0114\ 1114\ 1005\ \cdots \\
0101\ 0202\ 0112\ 0103\ 0203\ 0113\ 1113\ 1213\ 1004\ 1114\ 0214\ 1214\ \cdots \\
1101\ 1202\ 1112\ 1103\ 1203\ 1113\ 2113\ 2213\ 2004\ 2114\ 1214\ \cdots \\
0011\ 0112\ 0022\ 0013\ 0113\ 0023\ 1023\ 1123\ 1014\ 1024\ \cdots \\
0002\ 0103\ 0013\ 0004\ 0104\ 0014\ 1014\ 1114\ 1005\ \cdots \\
0102\ 0203\ 0113\ 0104\ 0204\ 0114\ 1114\ 1214\ \cdots \\
0012\ 0113\ 0023\ 0014\ 0114\ 0024\ 1024\ \cdots \\
1012\ 1113\ 1023\ 1014\ 1114\ 1024\ \cdots \\
1112\ 1213\ 1123\ 1114\ 1214\ \cdots \\
1003\ 1104\ 1014\ 1005\ \cdots \\
1013\ 1114\ 1024\ \cdots \\
1113\ 1214\ \cdots \\
1005\ \cdots \\
\cdots
\end{array}$$

From this matrix, we can find the new $N_{24} = 14$. For the monomial 1105, we get the following two sequences:

$$\begin{aligned}
RH_{25} = \{&0000,0001,0011,0111,1111,0002,1002,0102,0012,\\
&1012,0112,1112,1003,1103,0013,1013,1113,1104,1105,\cdots\}
\end{aligned}$$

$$\begin{aligned}
CH_{25} = \{&0000,0001,0011,0111,1111,0002,0102,0012,1012,\\
&0112,1112,1003,0103,1103,0013,1013,1113,1104,1105,\cdots\}
\end{aligned}$$

We can thus find the new value $N_{25} = 19$. It follows that this code $\mathbb{C}$ has minimum distance of at least 14.

4. Hermitian-like codes over GF(2^2)

In this section, we discuss the Hermitian-like codes GF(2^2) defined by the Hermitian-like curve (7) in Example 4. There are $3 \cdot 2^{m-1} + 1$ roots (points) on this curve. Indeed, let α be a primitive element of GF(2^2) with $\alpha^2 + \alpha + 1 = 0$. Then the points are $(0, 0, \ldots, 0)$ and $(r_1, r_2, \ldots, r_m)$, where $r_1, r_2, \ldots, r_m$ are defined as follows. For each $r_i = 1$, we have $r_{i+1} = \alpha, \alpha^2$; for each $r_i = \alpha$, we have $r_{i+1} = 1, \alpha$; for each $r_i = \alpha^2$, we have $r_{i+1} = 1, \alpha^2$. Finally, $r_1 = 1, \alpha, \alpha^2$.

Let $a_m a_{m-1} \cdots a_1$ denote the monomial $x_m^{a_m} x_{m-1}^{a_{m-1}} \cdots x_1^{a_1}$. We can define

$$w(x_i) = 2^{m-i} \tag{13}$$

From the equation of the curve, we have:

$$\cdots 21 \cdots = \cdots 02 \cdots + \cdots 10 \cdots \tag{14}$$

and

$$\cdots (i+2)(j+1) \cdots = \cdots i(j+2) \cdots + \cdots (i+1)j \cdots \tag{15}$$

Among the points of the curve, there is exactly one point $(0, 0, \ldots, 0)$ having 0 components and there are no other points having any 0 component. On the other hand, since $x_i^3 = 1$ for $x_i \neq 0$, we have:

$$a_m \cdots a_i \cdots a_1 = a_m \cdots (a_i + 3) \cdots a_1 \tag{16}$$

for $a_m \cdots a_i \cdots a_1 \neq 0 \cdots 0 \cdots 0$. For example, $00103 = 00100$, $20031 = 20001$, $00104 = 00101$, $20034 = 20001$, $0004 = 0001$, and so on. But $0003 \neq 0000$. Using this result and the equation (7), we have the following reductions:

$$\cdots 22 \cdots = \cdots 00 \cdots + \cdots 11 \cdots \tag{17}$$

and

$$\cdots 20 \cdots = \cdots 12 \cdots + \cdots 01 \cdots \tag{18}$$

In what follows we will be interested in the monomials $a_m \cdots a_i \cdots a_1$, where $a_i \in \{0, 1\}$ for $2 \leq i \leq m$ and $a_1 \geq 0$. We call them *standard* monomials. The following lemma can be easily proved.

Lemma 2. *The product of any two standard monomials $a_m \cdots a_1$ and $b_m \cdots b_1$ can be uniquely expressed as a linear combination of the standard monomials $c_m^\mu \cdots c_1^\mu$ repeatedly using* (15), (17), *and* (18), *as follows:*

$$a_m \cdots a_1 \cdot b_m \cdots b_1 = \sum_\mu c_m^\mu \cdots c_1^\mu \tag{19}$$

Example 12. Let $10111 \cdot 10104 = 20215$. Using (18), we get $12215 + 01215$. Using (17) for the first term, we get $10015 + 11115 + 01215$. Using (15) for the last term, we get $10015 + 11115 + 01105 + 01025$. Using (15) for the last term again, we get $10015 + 11115 + 01105 + 01006 + 01014$. □

Comparing this result with the results in [5] and [6], we find that the new approach can be used to construct more efficient Hermitian-like codes over $\mathrm{GF}(2^2)$.

In the following we derive an algorithm to determine the good terms for $\boldsymbol{h}_r$. Then we select the pairs of standard monomials, whose products are good terms for $\boldsymbol{h}_r$ and do not produce any bad terms.

Lemma 3. *Let $1 \cdots 1$ denote k consecutive 1's. Then $21 \cdots 15 \sim 10 \cdots 15$.*

Proof. We prove this lemma using mathematical induction. When $k = 1$, $215 = 025 + 105 = 006 + 014 + 105$. Since $006 = 003, 215 = 003 + 104 + 105$. $w(105) > w(003)$, $w(014)$. Thus $215 \sim 015$ and the lemma is true.

Assume the lemma is true for k, and consider the case of $k+1$. Then $211\cdots15 = 101\cdots15 + 021\cdots15$. Using the assumption, $021\cdots15 \sim 010\cdots15$. On the other hand, $w(010\cdots15) < w(101\cdots15)$, and we have $211\cdots15 \sim 101\cdots15$. Thus the lemma is also true for $k+1$. ■

Lemma 4. $a_m\cdots a_{p+1}01\cdots15 \sim a_m\cdots a_{p+1}20\cdots15$.

Proof. This lemma is equivalent to $01\cdots15 \sim 20\cdots15$. First we consider the case of $k=1$. Since $015 = 205+125 = 205+106+114$ and $106 = 103$, we have $015 \sim 205$ the lemma is true. Now assume that the lemma is true for k, and consider the case of $k+1$. We have $011\cdots15 = 201\cdots15+121\cdots15$. Using Lemma 3, the second term is consistent with $110\cdots15$. Since $w(110\cdots15) < w(201\cdots15)$, we have $011\cdots15 \sim 210\cdots15$ and the lemma is true for $k+1$. ■

Lemma 5. $\cdots011\cdots12\cdots \sim \cdots201\cdots12\cdots$.

Proof. Repeatedly using $20 = 01+12$ and in the last step using $22 = 11+00$, we have:

$$\begin{aligned}
\cdots201\cdots12\cdots &= \cdots011\cdots12\cdots+\cdots121\cdots12\cdots \\
&= \cdots011\cdots12\cdots+\cdots110\cdots12\cdots+\cdots102\cdots12\cdots \\
&= \cdots \\
&= \cdots011\cdots12\cdots+\cdots110\cdots12\cdots+\cdots+\cdots101\cdots22\cdots \\
&= \cdots011\cdots12\cdots+\cdots110\cdots12\cdots+\cdots+\cdots101\cdots11\cdots \\
&\quad +\cdots101\cdots00\cdots
\end{aligned}$$

Among the right side terms, $\cdots110\cdots12\cdots$ has the maximal weight and its weight is equal to the weight of $\cdots201\cdots12\cdots$. The proof is thus complete. ■

Using Lemmas 3, 4, and 5, we have the following good terms for $\boldsymbol{h}_r$.

Theorem 6. *Let* $a_m,\cdots,a_p \in \{0,1\}$. *Then:*

$$\begin{aligned}
a_m\cdots a_p00\cdots011\cdots12a_q\cdots a_25 &\sim a_m\cdots a_p00\cdots201\cdots12a_q\cdots a_25 \\
&\sim \cdots \\
&\sim a_m\cdots a_p002\cdots101\cdots12a_q\cdots a_25 \\
&\sim a_m\cdots a_p021\cdots101\cdots12a_q\cdots a_25 \\
&\sim a_m\cdots a_p21\cdots101\cdots12a_q\cdots a_25
\end{aligned}$$

As an application of Theorem 6, we have for example: $0011000115 \sim 0011002015 \sim 0011021015 \sim 0011211015 \sim 0201211015 \sim 2101211015$.

Now we consider the problem of how to find a row sequence and a column sequence such that they are nearly-well-behaving sequences with $\boldsymbol{h}_r$, for a given standard monomial $\boldsymbol{h}_r$ with $a_2 = 1$ and $a_1 = 5$. First we give a formula for the standard monomials of the row sequence and the column sequence. Then we prove that these sequences are nearly-well-behaving sequences for given $\boldsymbol{h}_r$.

Example 13. 11015 $\sim$ 11205. For 11205, we select row monomials: $**10@$, and column monomials: $**10@$. Then 00000 and 11015 are added into the row and column sequence, and 11012 and 00003 are deleted from the row and column sequence. For this example, the row sequence is:

$$\{00000, 00102, 10102, 01102, 11102, 00103, 10103, 01103, 11103, 11015\}$$

and the column sequence is:

$$\{00000, 00102, 10102, 01102, 11102, 00103, 10103, 01103, 11103, 11015\}.$$

They form the following syndrome matrix:

00000	00102	10102	01102	11102	00103	10103	01103	11103	11015
00102	00204	10204	01204	11204	00205	10205	01205	11205	
10102	10204	20204	11204	21204	10205	20205	11205		
01102	01204	11204	02204	12204	01205	11205			
11102	11204	21204	12204	22204	11205				
00103	00205	10205	01205	11205					
10103	10205	20205	11205						
01103	01205	11205							
11103	11205								
11015									

□

Example 14. 001110015 $\sim$ 001110205 $\sim$ 001112105 $\sim$ 020112105 $\sim$ 210112105. We are interested only in 001112105, 020112105 and 210112105. We have the following selection for the row and column sequences:

	001112105	$\sim$	020112105	$\sim$	210112105
Row	$00***110@$		$010**110@$		$110**110@$
Column	$00***100@$		$010**100@$		$100**100@$
#	16		8		8

000000000, 001110015 are added in to the two sequences, respectively. We obtain the row and column sequences with size $16 + 8 + 8 + 2 = 34$. Thus $N(001110015) \geq 34$.

□

In the following we introduce a selection to find the nearly-well-behaving row and column sequences. Once the sequences are selected, we can calculate the size of the row and column sequences, which gives the lower bound of $N(\boldsymbol{h}_r)$.

For each $\boldsymbol{h}_r$, with $a_2a_1 = 15$, we can divide the $a_m \cdots a_3 15$ into some partitions L_μ and K_μ of size l_μ and k_μ, respectively, for $\mu = 1, 2, \ldots, p$. L_μ contains consecutive 1's and K_μ contains consecutive 0's. The index μ is increased from right to left and k_p may be 0 and the other k_μ and $l_\mu \geq 1$.

Example 15. Let us consider the standard monomial:

$$11100001100010011115$$

We have $p = 4$; $L_1 = 1111$, $K_1 = 00$; $L_2 = 1$, $K_2 = 000$; $L_3 = 11$; $K_3 = 0000$; $L_4 = 111$, $K_4 = \varnothing$; and $l_1 = 4$, $k_1 = 2$; $l_2 = 1$, $k_2 = 3$; $l_3 = 2, k_3 = 4$; $l_4 = 3, k_4 = 0$.

□

Let $\boldsymbol{h}_r$ be the above standard monomial. Let $\boldsymbol{h}'$ be such monomial obtained through the following changes: for each $\mu \neq p$, L_μ is changed to L'_μ by changing the leftmost 1 to 0; K_μ is changed to K'_μ by changing the leftmost 0 to 2 and the other 0 to 1. Then obviously $\boldsymbol{h}' \sim \boldsymbol{h}_r$. Thus we have:

$$\boldsymbol{h}' = 00\cdots011\cdots5 \sim 00\cdots201\cdots5 \sim \cdots \sim 21\cdots101\cdots5. \quad (20)$$

These monomials are called *useful monomials* of $\boldsymbol{h}_r$. For example, let us consider the standard monomial $\boldsymbol{h}_r = 11100001100010011115$, as in Example 15. Then the monomial 11121110121102101115 is the unique useful monomial of $\boldsymbol{h}_r$. Now, if $\boldsymbol{h}_r = 00111000011000100111115$, then we have:

$$00111211101211021011115 \sim 02011211101211021011115 \sim 21011211101211021011115$$

We have three useful monomials of 00111000011000100111115, namely 0011121 110121102101115, 02011211101211021011115, and 21011211101211021011115.

Now consider a quasi-standard monomial $c_m\cdots5$. If $\cdots c_jc_{j-1}\cdots c_i2\cdots$, where $c_s = 1$ for $s = i, i+1, \ldots, j$, then the j-th location is called a double location. If $\cdots c_jc_{j-1}\cdots c_i0\cdots$, where $c_s = 1$ for $s = i, i+1, \ldots, j$, then the j-th location is called a single location. If $c_j = 0, 2$, then the j-th location is called 0-location or 2-location, respectively. With this notation, we have the following selection rule.

> **Selection rule:** For each useful monomial of $\boldsymbol{h}_r$, we select the following monomials as row monomials: at double locations there is a $*$; at single locations there is a 1; at 0 or 2 locations there is a 0 or 1; at the first location, there is a @. We select the following monomials as column monomials: at double locations there is a $*$; at single locations there is a 0; at 0 or 2 locations there is a 0 or 1; at the first location, there is a @.

Example 15 (continued). For $\boldsymbol{h}_r = 00111000011000100111115$, we select the row monomials and column monomials, respectively, as follows:

00111211101211021011115	~	02011211101211021011115	~	21011211101211021011115
00 * * * 11110 * 1110110 * * * @		010 * *11110 * 1110110 * * * @		110 * *11110 * 1110110 * * * @
00 * * * 10000 * 1000100 * * * @		010 * *10000 * 1000100 * * * @		100 * *10000 * 1000100 * * * @

□

Theorem 7. *Given a monomial $\boldsymbol{h}_r$ with $a_2a_1 = 15$, the size $N^*(\boldsymbol{h}_r)$ of the row and columns obtained by the above rules, isgiven by*

$$N^*(\boldsymbol{h}_r) = 2\left(2^{\sum_{\mu=1}^{p}(l_\mu-1)}(k_p+2)\right)+2 \quad (21)$$

Example 13 (continued). We have $p = 2$, $L_1 = 1$, $K_1 = 0$; $L_2 = 11$, $K_2 = \varnothing$; and $l_1 = 1$, $k_1 = 1$; $l_2 = 2$, $k_2 = 0$. Thus it follows from Theorem 7 that

$$N^*(11015) = 2(2^{(1-1)+(2-1)}(0+2))+2 = 10$$

□

Example 14 (continued). We have $p = 2$, $L_1 = 1$, $K_1 = 00$; $L_2 = 111$, $K_2 = 00$; and $l_1 = 1$, $k_1 = 2$; $l_2 = 3$, $k_2 = 2$. Thus by Theorem 7 we have

$$N^*(001110015) = 2(2^{(1-1)+(3-1)}(2+2)) + 2 = 34$$

□

Example 15 (continued). It follows from Theorem 7 that

$$N^*(11100001100010011115) = 2(2^{(4-1)+(1-1)+(2-1)+(3-1)}(0+2)) + 2 = 258$$

□

Now we prove the following main theorem:

Theorem 8. *The row sequence and column sequence obtained by the above algorithm are nearly-well-behaving sequences for h_r.*

Proof. Let $r_m \cdots r_1$ and $c_m \cdots c_1$ be row and column monomials respectively, and $w(r_m \cdots r_1) + w(c_m \cdots c_1) \leq w(h_r)$, and they are not h_1 and h_r. Let $p_m \cdots p_1 = r_m \cdots r_1 \cdot c_m \cdots c_1$, where $p_i = r_i + c_i$ for $m \geq i \geq 1$. Thus let $m - k_p \geq i$, if ith location is 0 and 2 location, then $p_i = 0$ or 2, respectively; if it is a single location, then $p_i = 1$; if it is a double location, then $p_i = 0, 1$, or 2. On the last k_p locations, namely at $p_m \cdots p_{m-k_p+1}$, if $p_\mu = 2$, then $p_{\mu-1} = \cdots = p_{m-k_p+1} = 1$. Thus this product is consistent with h_p, so $w(h_p) \leq w(h_r)$. This completes the proof of the theorem. ■

5. Codes exceeding the Garcia-Stichtenoth bound

Now we are interested in the following standard monomial $t_m \cdots t_{l+1}01 \cdots 15$, where $l > 2$. We want to find a lower bound of its N number.

From the above algorithm, $t_m \cdots t_{l+1}01 \cdots 15$ and $t_m \cdots t_{l+1}20 \cdots 15$ are its equivalent quasi-standard monomials. We select the monomials

$$\begin{gathered} 00 \cdots 00 \\ t_m \cdots t_{l+1} 0 a_{l-1} \cdots a_2 a_1 \\ t_m \cdots t_{l+1} 10 a'_{l-2} \cdots a'_2 a'_1 \\ t_m \cdots t_{l+1} 01 \cdots 15 \end{gathered}$$

as rows, and select

$$\begin{gathered} 00 \cdots 00 \\ 0 \cdots 0 b_{l-1} \cdots b_2 b_1 \\ 0 \cdots 10 b'_{l-2} \cdots b'_2 b'_1 \\ t_m \cdots t_{p+1} 01 \cdots 15 \end{gathered}$$

as columns, where $a_i + b_i = a'_i + b'_i = 1$ for $2 \leq i \leq l-2$, $a_{l-1} + b_{l-1} = 1$, $a_1 + b_1 = a'_1 + b'_1 = 5$, $a_1, a'_1, b_1, b'_1 \in \{2, 3\}$, $a_i, b_i, a'_i, b'_i \in \{0, 1\}$. Thus each row and each column has $2^l + 2^{l-1}$ terms. Then it can be easily seen that

this syndrome matrix is a nearly well-behaving sequence. The number of well-behaving terms consistent with $t_m \cdots t_{l+1}01 \cdots 15$ is equal to $2^l + 2^{l-1}$. Thus from Theorem 7, we have the following theorem.

Theorem 9.

$$N(t_m \cdots t_{p+1}01 \cdots 15) \geq 2^l + 2^{l-1} + 2$$

Example 16. Let us consider 10115. In this case $m = 5, l = 3$. Thus 10115 and 12015 are equivalent quasi-standard monomials. Select 00000, 10112, 10012, 10102, 10002, 10113, 10013, 10103, 10003, 11012, 11002, 11013, 11003, 10115 to be rows and select 00000, 00112, 00012, 00102, 00002, 00113, 00013, 00103, 00003, 01012, 01002, 01013, 01003, 10115 to be columns. We have the following syndrome matrix:

00000	10002	11002	10102	10012	11012	10112	10003	11003	10103	10013	11013	10113	10115
00002	10004	11004	10104	10014	11014	10114	10005	11005	10105	10015	11015	10115	
01002	11004	12004	11104	11014	12014	11114	11005	12005	11105	11015	12015		
00102	10104	11104	10204	10114	11114	10214	10105	11105	10205	10115			
00012	10014	11014	10114	10024	11024	10124	10015	11015	10115				
01012	11014	12014	11114	11024	12024	11124	11015	12015					
00112	10114	11114	10214	10124	11124	10224	10115						
00003	10005	11005	10105	10015	11015	10115							
01003	11005	12005	11105	11015	12015								
00103	10105	11105	10205	10115									
00013	10015	11015	10115										
01013	11015	12015											
00113	10115												
10115													

There are $14 = 2^l + 2^{l-1} + 2$ well-behaving terms consistenting with 10115. □

If we choose all monomials except $t_m \cdots t_{l+1}t_l1 \cdots 15$ as parity-check matrix, then this matrix defines an Hermitian-like code (n, k, d) over GF(2^2), where $n = 3{\cdot}2^{m-1} + 1$, $k = 2^{m-l+1}$, and $d \geq 2^l + 2^{l-1} + 2$. Let us consider the special case: $l = \frac{m}{2}$ where m is even. Thus

$$k + d \geq 2^{\frac{m}{2}+1} + 2^{\frac{m}{2}} + 2^{\frac{m}{2}-1} + 2 = 3{\cdot}2^{\frac{m}{2}} + 2^{\frac{m}{2}-1} + 2 \qquad (22)$$

On the other hand, for $q = 2$ and from (9), the Garcia-Stichtenoth bound is

$$g_m = \begin{cases} 2^m + 2^{m-1} - 2^{\frac{m+1}{2}} - 2^{\frac{m+1}{2}} + 1 & \text{if } m \text{ is odd} \\ 2^m + 2^{m-1} - 2^{\frac{m}{2}-1} - 2^{\frac{m+2}{2}} + 1 & \text{if } m \text{ is even} \end{cases} \qquad (23)$$

Thus we have

$$n - g_m = \begin{cases} 2^{\frac{m+1}{2}+1} + 1 & \text{if } m \text{ is odd} \\ 3{\cdot}2^{\frac{m}{2}} + 1 & \text{if } m \text{ is even} \end{cases} \qquad (24)$$

When m is even, if we construct the algebraic-geometric code with parameters (n, k, d^*), then from Reimann-Roch theorem we have $d^* = n - k - g_m + 1$, that is $d^* + k - 1 \geq n - g_m$. Comparing (22) and (24), we have $d + k - 1 > d^* + k - 1$. These codes thus exceed the Garcia-Stichtenoth bound.

We point out that this is only a special case. Using Theorem 7, we can construct many codes exceeding the Garcia-Stichtenoth bound.

6. Conclusions

In this paper, we have improved the minimum distance bound for Hermitian-like codes. We also explicitly constructed a sequence of Hermitian-like codes over GF(4) which exceed the Garcia-Stichtenoth bound for low rates. Theorem 7 can be further improved. However, these codes are not asymptotically good. Further research should be done in applying this approach to the Hermitian-like codes over $\mathrm{GF}(q^2)$ for $q > 2$.

Acknowledgment. The authors are deeply grateful to Xinwen Wu, Junmei Zhu, and Dahon Xie for helpful discussion and comments concerning this work. The authors would like to thank R.E. Blahut, Ruud Pellikaan, and Tom Høholdt for many discussions and valuable suggestions on the topic of this paper.

References

[1] M. DE BOER, Some numerical results and ideas, unpublished manuscript, 1996.

[2] G.-L. FENG AND T.R.N. RAO, A simple approach for construction of algebraic-geometric codes from affine plane curves, *IEEE Trans. Inform. Theory*, vol. 40, pp. 1003–1012, 1994.

[3] ———, Improved geometric Goppa codes, Part I: Basic theory, *IEEE Trans. Inform. Theory*, vol. 41, pp. 1678–1693, 1995.

[4] A. GARCIA AND H. STICHTENOTH, A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlăduţ bound, *Inventiones Math.*, vol. 121, pp. 211–222, 1995.

[5] V.D. GOPPA, Rational representation of codes and (L, g)-codes, *Problemy Peredachi Informatsii*, vol. 7, pp. 223–229, 1971.

[6] ———, Codes associated with divisors, *Problemy Peredachi Informatsii*, vol. 13, pp. 33–39, 1977.

[7] ———, Codes on algebraic curves, *Soviet Math. Doklady*, vol. 24, pp. 75–91, 1981.

[8] ———, Algebraic-geometric codes, *Math. USSR Izvestiya*, vol. 21, pp. 75–91, 1983.

[9] F.J. MACWILLIAMS AND N.J.A. SLOANE, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland 1977.

[10] R. PELLIKAAN, On the gonality of curves, abundant codes and decoding, in *Coding Theory and Algebraic Geometry*, H. Stichtenoth and M.A. Tsfasman (Editors), *Lect. Notes Math.*, vol. 1518, pp. 132–145, Berlin: Springer-Verlag 1991.

[11] ———, On the existence of order functions, in *Proc. 2-nd Conference on Design, Codes and Finite Geometry*, Shanghai, China, May 1996.

[12] ———, Asymptotically good sequences of curves and codes, in *Proc. 34-th Allerton Conf. Comm., Control, and Computing*, Monticello, IL., pp. 276–285, October 1996.

[13] M.A. TSFASMAN, S.G. VLĂDUTS, AND T. ZINK, Modular curves, Shimura curves, and Goppa codes better than the Varshamov-Gilbert bound, *Math. Nachrichten*, vol. 104, pp. 13–28, 1982.

[14] S.G. VLĂDUTS, G.L. KATSMAN, and M.A. TSFASMAN, Modular curves and codes with polynomial complexity of construction, *Probl. Peredachi Inform.*, vol. 20, pp. 47–55, 1984.

[15] C. VOSS AND T. HØHOLDT, An explicit construction of a sequence of codes attaining the Tsfasman-Vlăduţ-Zink bound, *IEEE Trans. Inform. Theory*, vol. 43, pp. 128–135, 1997.

The BM Algorithm and the BMS Algorithm

Shojiro Sakata

Department of Computer Science and Mathematics
University of Electro-Communications
Chofu-shi, Tokyo 182, Japan
sakata@cs.uec.ac.jp

Abstract. We present a brief overview of the Berlekamp-Massey (BM) algorithm and of the Berlekamp-Massey-Sakata (BMS) algorithm, with an emphasis on a close relationship between them, which can be summarized as follows: *BM algorithm + Gröbner basis = BMS algorithm.* Furthermore, we describe various forms of these algorithms, all of which are related to fast decoding of algebraic codes including algebraic-geometric (AG) codes. As a byproduct, we obtain a recursive BMS algorithm which is an extension of Blahut's recursive BM algorithm. Throughout this paper, we recall the assertions of R.E. Blahut on the importance of fast decoding for good codes, particularly as described in his books, which have spurred recent developments of novel decoding algorithms of AG codes.

1. Introduction

The Berlekamp-Massey-Sakata (BMS) algorithm [**5, 6**] has been said to be an extension or generalization of the Berlekamp-Massey (BM) algorithm [**3, 4, 1, 2**]. However, one might wonder if this extension is a natural one or there is no other natural generalization. In this paper, we present a brief survey of both the BM algorithm and the BMS algorithm from the same point of view and emphasize some closer relationship between them, which can be summarized by the following formula: *BM algorithm + Gröbner basis = BMS algorithm.* As another generalization, we review a BMS algorithm over polynomial modules [**7, 10**], and finally develop a recursive form of the BMS algorithm over polynomial modules, which is an extension of Blahut's recursive BM algorithm [**1**].

This short note is organized as follows. In § 2, we provide a streamlined version of the BM algorithm and compare it with the conventional form [**3, 4, 1, 2**]. Then in § 3, we introduce a multidimensional generalization of the BM algorithm which is the BMS algorithm for Gröbner bases of ideals in the multivariate polynomial ring. In Sections 4 and 5, we present a BMS algorithm over polynomial modules [**7, 10**] and a recursive form of that algorithm. § 6 contains some

concluding remarks about applications of the BMS algorithm to decoding algebraic geometric (AG) codes [**7, 10, 9, 8, 11**]. Throughout this note, we recall R.E. Blahut's emphasis on importance of fast decoding for good codes, particularly as described in his books [**1, 2**], which have spurred recent developments of novel decoding algorithms of algebraic geometric codes. Specifically, in § 5 we have added an example motivated by these ideas.

2. The BM algorithm

The following is a streamlined version of the Berlekamp-Massey algorithm.

Algorithm 1:

Step 1 (initialization): $p := 0;\ f := 1;\ s := 0;$

$g := 0;\ c := -1;\ d_g := 0;$

Step 2 (discrepancy): $d_f := \sum_{r=0}^{s} f_r u_{r+p-s};$

Step 3 (updating): **if** $d_f \neq 0$ **then**

if $p \leq s + c$ **then** $f := f - (d_f/d_g)x^{s+c-p}g;$

else

begin $r := p - c;\ h := x^{r-s}f - (d_f/d_g)g;$

$g := f;\ c := p - s;\ d_g := d_f;\ f := h;\ s := r;$

end;

Step 4 (termination): $p := p + 1;$

if $p < q$ **then go to Step 2 else stop.**

Although the BM algorithm was devised by Berlekamp to find the error locators of BCH codes, it actually finds the shortest linear feedback shift register capable of generating a given finite one-dimensional (1D) array $u = (u_p)$, $0 \leq p < q$, over a finite field $\mathbb{F}_q$, as it is explained by Berlekamp himself [**3**] and by Massey [**4**]. Blahut [**1, 2**] gave a concise form of the BM algorithm.

Blahut's BM algorithm:

Step 1 (initialization): $p := 0;\ f := 1;\ s := 0;\ g := 0;$

Step 2 (discrepancy): $\Delta := \sum_{r=0}^{p} f_r u_{p-r};$

Step 3 (updating):

$$\delta := \begin{cases} 1 & \text{if } \Delta \neq 0 \wedge 2s \leq p, \\ 0 & \text{otherwise;} \end{cases}$$

$$\begin{pmatrix} f \\ g \end{pmatrix} := \begin{pmatrix} 1 & -\Delta x \\ \Delta^{-1}\delta & (1-\delta)x \end{pmatrix} \begin{pmatrix} f \\ g \end{pmatrix};$$

$$s := \delta(p-s+1) + (1-\delta)s;$$

Step 4 (termination): **if** $p < q$ **then go to Step 2 else stop.**

If we take one arithmetic operation over the finite field as a unit of computation, the BM algorithm has computational complexity of order $\mathcal{O}(q^2)$, and is well-known for this efficiency. Precisely, the BM problem which is solved by Algorithm 1 is as follows. Here, $\mathbb{Z}_0$ is the set of nonnegative integers and $q \in \mathbb{Z}_0$.

> **Given:** a 1D array $u = (u_p)$, $0 \leq p \leq q-1$;
>
> **Find:** a monomial polynomial $f = \sum_{r=0}^{s} f_r x^r$ with minimal degree $\deg(f) = s \in \mathbb{Z}_0$ such that the corresponding linear recurrence is satisfied by the given array u:

$$\sum_{r=0}^{s} f_r u_{r+p-s} = 0, \qquad s \leq p < q \tag{1}$$

From (1), the conventional expression of the polynomial $f = \sum_{r=0}^{s} f_r x^r$ and the corresponding linear recurrence

$$\sum_{r=0}^{s} f_r u_{p-r} = 0, \qquad s \leq p < q$$

are derived by the following change of notation: $f_r := f_{s-r}$, $0 \leq r \leq s$. This equation is derived also from the polynomial equation (called the key equation):

$$f(x)u(x) \equiv w(x) \qquad (\mathrm{mod}\, x^q)$$

where the polynomial f is denoted as $f(x)$, $u(x) := \sum_{r=0}^{q-1} u_r x^r$, and $w(x)$ is a polynomial of degree less than s. (Our recurrence (1) can be rewritten as

$$\sum_{r=0}^{s} f_r u_{r+p} = 0, \qquad 0 \leq p < q-s \tag{2}$$

where the meaning of the points p is different in expressions (1) and (2). The former p and the latter p are considered to represent the absolute and relative coordinates on the array u, respectively, where the latter p depends on the degree s of f.)

One of the merits of our notation is that our error locator polynomial f has as the locators just the zeros, but not the inverses of zeros (as the conventional error locator polynomial has). Furthermore, the degree $s = \deg(f)$ of

the solution polynomial f is written explicitly in the equation and its meaning becomes clear from the expression. Throughout the iterations of Algorithm 1, the identity $s = c + 1$ is kept invariant, which is an important fact on understanding the implications of the BM algorithm. The quantity $c = \mathsf{span}(g)$ called the span of the 'auxiliary polynomial' g represents the limit of the *excluding point set* $\Delta(u^t) = \{r \in \mathbb{Z}_0 : 0 \le r \le c\}$ at the iteration of the point t, where $u^t := (u_r : 0 \le r < t)$ is a part of the given array u and there is no polynomial f such that the linear recurrence (1) is satisfied for all p, $s = \deg(f) \le p < t$, and $s \in \Delta(u^t)$. On the other hand, there is a 'minimal' polynomial f with $\deg(f) = s$ for which (1) is satisfied at all p, $s \le p < t$. According to the basic lemma of the BM algorithm, the span of a polynomial f is given as the difference of the order of f and the degree of f: $\mathsf{span}(f) = \mathsf{ord}(f) - \deg(f)$, where $\mathsf{ord}(f)$ is defined to be the point $q \in \mathbb{Z}_0$ such that the linear recurrence (1) is satisfied at all p, $\deg(f) \le p < q$, and (1) is not satisfied at $p = q$.

Before proceeding to the multidimensional case in the next section, we remark that, for a given perfect 1D array $u = (u_p)$, $p \in \mathbb{Z}_0$, the subset

$$\mathsf{Val}(u) := \left\{ f \in \mathbb{F}_q[x] : \sum_{r=0}^{\deg(f)} f_r u_{r+p} = 0,\ p \in \mathbb{Z}_0 \right\}$$

is an ideal of the ring $\mathbb{F}_q[x]$. Since the ring $\mathbb{F}_q[x]$ is Euclidean, the ideal $\mathsf{Val}(u)$ is generated by a single polynomial, which is a minimal polynomial of u, because it has the minimum degree among the polynomials of $\mathsf{Val}(u)$.

3. The BMS algorithm

At the appearance of the BMS algorithm [5], it was not devised for decoding of multidimensional codes. It was to find a kind of 'simplest' m-dimensional (mD) linear feedback shift register capable of generating a given finite mD array over $\mathbb{F}_q$. In the mD space, or rather in the mD integral lattice $\mathbb{Z}_0^m$ which is the m-fold Cartesian product of $\mathbb{Z}_0$, we need an admissible total order $\le_T$ to define a finite array in $\mathbb{Z}_0^m$ and the degree of an m-variate polynomial $f \in \mathbb{F}_q[\boldsymbol{x}] := \mathbb{F}_q[x_1, \cdots, x_m]$. As the total order $\le_T$, we often take the total degree order or the weighted degree order. Then, a finite array u is defined to be a mapping from a subset $\mathsf{Supp}(u)$ of $\mathbb{Z}_0^m$ into $\mathbb{F}_q$. In particular, an array with $\mathsf{Supp}(u) = \Sigma^{\boldsymbol{q}} := \{\boldsymbol{p} \in \mathbb{Z}_0^m : \boldsymbol{p} <_T \boldsymbol{q}\}$ is denoted as $u = u^{\boldsymbol{q}}$. A polynomial $f \in \mathbb{F}_q[\boldsymbol{x}]$ is written as

$$f = \sum_{\boldsymbol{r} \in \mathsf{Supp}(f)} f_{\boldsymbol{r}} \boldsymbol{x}^{\boldsymbol{r}},$$

where $f_{\boldsymbol{r}} \in \mathbb{F}_q$, $\boldsymbol{x}^{\boldsymbol{r}} := x_1^{r_1} \cdots x_m^{r_m}$ for $\boldsymbol{r} \in \mathbb{Z}_0^m$, $\mathsf{Supp}(f) := \{\boldsymbol{r} \in \mathbb{Z}_0^m : f_{\boldsymbol{r}} \ne 0\}$. Notice that the degree of f is defined as $\deg(f) := \max_T \mathsf{Supp}(f)$ which is the maximum element of $\mathsf{Supp}(f)$ with respect to $\le_T$.

On the other hand, to consider a 'simplest' polynomial, we need a partial order $\le_P$ in $\mathbb{Z}_0^m$, where we can interpret minimal with respect to $\le_P$ as 'simplest'.

Although the total order $\leq_T$ and the partial order $\leq_P$ are quite the same in the 1D lattice $\mathbb{Z}_0$, we must distinguish between them clearly in $\mathbb{Z}_0^m$ for $m \geq 2$. Furthermore, we need in general a plural number of polynomials as a basis of an ideal in the m-variate polynomial ring $\mathbb{F}_q[\boldsymbol{x}]$. The separation between the total order and the partial order and inclusion of polynomial sets force us to introduce Gröbner bases in our theory. With respect to the partial order $\leq_P$, it is natural to consider subsets of $\mathbb{Z}_0^m$ such as $\Sigma_{\boldsymbol{s}} := \{\boldsymbol{p} \in \mathbb{Z}_0^m : \boldsymbol{s} \leq_P \boldsymbol{p}\}$ and $\Gamma_{\boldsymbol{c}} := \{\boldsymbol{p} \in \mathbb{Z}_0^m : \boldsymbol{p} \leq_P \boldsymbol{c}\}$ and a partition of the whole set $\mathbb{Z}_0^m$: $\mathbb{Z}_0^m = \Sigma_S \cup \Gamma_C$, where $\Sigma_S = \cup_{\boldsymbol{s} \in S} \Sigma_{\boldsymbol{s}}$, $\Gamma_C = \cup_{\boldsymbol{c} \in C} \Gamma_{\boldsymbol{c}}$ and $\Sigma_S \cap \Gamma_C = \varnothing$ for a pair of finite sets of points $S, C \subset \mathbb{Z}_0^m$ both of which are 'nondegenerate' with respect to $\leq_P$, i.e., if $\boldsymbol{p} \leq_P \boldsymbol{q}$ for either $\boldsymbol{p}, \boldsymbol{q} \in S$ or $\boldsymbol{p}, \boldsymbol{q} \in C$, then $\boldsymbol{p} = \boldsymbol{q}$.

Now, an mD linear recurrence corresponding to a polynomial f with $\deg(f) = \boldsymbol{s}$ is given by:

$$\sum_{\boldsymbol{r} \in \mathsf{Supp}(f)} f_{\boldsymbol{r}} u_{\boldsymbol{r}+\boldsymbol{p}-\boldsymbol{s}} = 0, \qquad \boldsymbol{s} \leq_P \boldsymbol{p} <_T \boldsymbol{q} \tag{3}$$

which should be satisfied by the given mD array $u = u^{\boldsymbol{q}}$. The expression (3) reduces straightforwardly to (1) in case $m = 1$. Using this notation, our BMS problem is stated as follows:

Given: an mD array $u = u^{\boldsymbol{q}} = (u_{\boldsymbol{p}})$, $\boldsymbol{0} \leq_T \boldsymbol{p} <_T \boldsymbol{q}$;

Find: a minimal polynomial set F of u such that:

(1). any $f \in F$ is 'valid' for u in the sense that the corresponding linear recurrence (3) is satisfied;

(2). F is 'minimal' in the sense that there is no valid polynomial g for u with $\deg(g) \in \Delta(F)$, where $S = \{\deg(f) : f \in F\}$ is nondegenerate and accompanied by $\Delta(F) = \Gamma_C$ such that $\mathbb{Z}_0^m = \Sigma_S \cup \Gamma_C$ is a partition, that is $\Delta(F) := \mathbb{Z}_0^m \setminus \Sigma_S$.

A minimal polynomial set F of u is not unique, but the subset $\Delta(F)$ is unique for u. Therefore, $\Delta(F)$ is denoted as $\Delta(u)$ and it is called the *excluded point set* of u. In the mD case, the concepts of 'order' and 'span' of a polynomial are defined in the same way as in the 1D case, but with some more implications:

The order of a polynomial f is the point $\boldsymbol{q} \in \mathbb{Z}_0^m$ such that the linear recurrence (3) is valid at all $\boldsymbol{p}$, $\boldsymbol{s}$ $(:= \deg(f)) \leq_P \boldsymbol{p} <_T \boldsymbol{q}$, and not at $\boldsymbol{p} = \boldsymbol{q}$; The span of f is $\mathsf{span}(f) := \mathsf{ord}(f) - \deg(f)$. In this context, the basic lemma states that there exists no polynomial f such that $\mathsf{ord}(f) >_T \boldsymbol{q}$ and $\deg(f) \leq_P \mathsf{span}(g)$ for some g with $\mathsf{ord}(g) \leq_T \boldsymbol{q}$.

The above problem can be solved by the following BMS algorithm, which is composed of iterations at the points $\boldsymbol{p} <_T \boldsymbol{q}$, moving successively from the current point $\boldsymbol{p}$ to the next point $\boldsymbol{p} \oplus 1$, according to the total order $\leq_T$. At each iteration, a minimal polynomial set F of the current $u^{\boldsymbol{p}}$, together with an auxiliary polynomial set G, is updated for $u^{\boldsymbol{p} \oplus 1}$, where G is associated to the point set C in the sense that $C = \{\mathsf{span}(g) : g \in G\}$. The set $\Delta(u^{\boldsymbol{p}})$ is

nondecreasing, so that $\Delta(u^{\boldsymbol{p}}) \subseteq \Delta(u^{\boldsymbol{p}\oplus 1})$. The updated set $\Delta(u^{\boldsymbol{p}\oplus 1}) = \Gamma_{C^+}$ accompanies the nondegenerate point set S^+ such that $\Sigma_{S^+} \cup \Gamma_{C^+} = \mathbb{Z}_0^m$.

In the algorithm that follows, we use the the following notation. For $B \subset \mathbb{Z}_0^m$, $\mathrm{Min}_P B$ is the set of all minimal points in B with respect to $\leq_P$. For two points $\boldsymbol{p} = (p_1, \cdots, p_m)$ and $\boldsymbol{q} = (q_1, \cdots, q_m) \in \mathbb{Z}_0^m$, we define:

$$\max(\boldsymbol{p}, \boldsymbol{q}) \ := \ (\max\{p_1, q_1\}, \cdots, \max\{p_m, q_m\}) \in \mathbb{Z}_0^m$$

For $\boldsymbol{s} = \deg(f)$, $\boldsymbol{p} = \mathsf{ord}(f)$, $\boldsymbol{c} = \mathsf{span}(g)$, we define $\boldsymbol{s}^+ := \max(\boldsymbol{s}, \boldsymbol{p} - \boldsymbol{c})$ and

$$h(f,g) \ := \ \boldsymbol{x}^{\boldsymbol{s}^+ - \boldsymbol{s}} f - (d_f/d_g)\boldsymbol{x}^{\boldsymbol{s}^+ - (\boldsymbol{p}-\boldsymbol{c})} g$$

Algorithm 2:

Step 1 (initialization): $\boldsymbol{p} := \boldsymbol{0}$; $F := \{1\}$; $G := \varnothing$; $S := \{\boldsymbol{0}\}$; $C := \varnothing$;

Step 2 (discrepancy): $F_N := \{f \in F : \mathsf{ord}(f) = \boldsymbol{p}\}$; $S_N := \{\deg(f) : f \in F_N\}$;

$F_{NN} := \{f \in F : \deg(f) \leq_P \boldsymbol{p} - \boldsymbol{s}', \exists \boldsymbol{s}' \in S_N\}$;

$S_{NN} := \{\deg(f) : f \in F_{NN}\}$;

Step 3(a) (updating): for each $f \in F_N \setminus F_{NN}$

begin $\boldsymbol{s}^+ := \boldsymbol{s}$;

$f^+ := h(f,g)$, where $g \in G$ such that $\mathsf{span}(g) \geq_P \mathsf{span}(f)$;

end;

for each $f \in F_{NN}$

begin $\boldsymbol{s} := \deg(f)$;

$\widehat{S} := \mathrm{Min}_P \{\max(\boldsymbol{s}, \boldsymbol{p} - \boldsymbol{c}) : \boldsymbol{c} \in C \cap \Gamma_{\boldsymbol{p}}\}$;

$\check{S} := \mathrm{Min}_P \, \Sigma_{\boldsymbol{s}} \setminus \Gamma_{\boldsymbol{p}}$; $S^+ := \mathrm{Min}_P \, \widehat{S} \cup \check{S}$;

for each $\boldsymbol{s}^+ \in S^+$

if $\boldsymbol{s}+ \in \widehat{S}$ then $f^+ := (f,g)$, where $g \in G$

such that $\mathsf{span}(g) = \boldsymbol{c}$, $\boldsymbol{s}^+ = \max(\boldsymbol{s}, \boldsymbol{p} - \boldsymbol{c})$,

else $f^+ := \boldsymbol{x}^{\boldsymbol{s}^+ - \boldsymbol{s}} f$;

end;

Step 3(b) (updating):

$F := (F \setminus F_N) \cup \{$all f^+ obtained in Step 3(a)$\}$;

$G_{NN} := \{g \in G : \mathsf{span}(g) <_P \mathsf{span}(f), {}^{\exists} f \in F_{NN}\}$;

$G := (G \setminus G_{NN}) \cup F_{NN}$;

Step 4 (termination): $\boldsymbol{p} := \boldsymbol{p} \oplus 1$; if $\boldsymbol{p} <_T \boldsymbol{q}$ then go to Step 2 else stop

---

In the mD case, there are several complications. The computational complexity of Algorithm 2 is $\mathcal{O}(\rho r^2)$, where $r = \#\Sigma^{\boldsymbol{q}}$ and $\#F \le \rho$ for some integer ρ provided that the cardinality $\#F$ of F is assumed to be upperbounded. Otherwise, the BMS algorithm is quite similar to the BM algorithm in the skeleton. Of course, there are several intrinsic differences between both of them. For example, the 1D relationship $s = c + 1$ is replaced by the partition $\Sigma_S \cup \Gamma_C = \mathbb{Z}_0^m$ in the mD case. Summarizing all of them, we can have the simple formula that "BM algorithm + Gröbner basis = BMS algorithm," which has various implications in it. For a perfect array u with $\mathsf{Supp}(u) = \mathbb{Z}_0^m$, the subset

$$\mathsf{Val}(u) := \left\{ f \in \mathbb{F}_q[\boldsymbol{x}] : \sum_{\boldsymbol{r} \in \mathsf{Supp}(f)} f_{\boldsymbol{r}} u_{\boldsymbol{r}+\boldsymbol{p}} = 0, \boldsymbol{p} \in \mathbb{Z}_0^m \right\}$$

is an ideal of the m-variate polynomial ring $\mathbb{F}_q[\boldsymbol{x}]$, which is generated by a minimal polynomial set of u, i.e., a Gröbner basis of the ideal. Thus, the BMS problem is just one of the basic problems of the Gröbner basis theory, and the BMS algorithm gives its solution as a minimal polynomial set of a certain finite part $u^{\boldsymbol{q}}$ of the perfect array u.

4. A BMS algorithm over polynomial modules

As a variation or extension of the BMS algorithm we can devise a BMS algorithm over polynomial modules. The former is to find a Gröbner basis of an ideal in the m-variate polynomial ring $\mathbb{F}_q[\boldsymbol{x}] = \mathbb{F}_q[x_1, \cdots, x_m]$, whereas the latter is to find a Gröbner basis of a submodule in the polynomial module R^ρ which is the ρ-fold Cartesian product of the univariate polynomial ring $R := \mathbb{F}_q[x]$ for a fixed integer ρ. The module R^ρ is an R-module in the sense that, for $\boldsymbol{f} = (f_i)_{1 \le i \le \rho} \in R^\rho$ and $f \in R$, the multiplication $f\boldsymbol{f} := (ff_i)_{1 \le i \le \rho} \in R^\rho$ is defined. In this context, we can consider the following situation. We are given an array matrix $\boldsymbol{u} = [u_j^i]$, $1 \le i, j \le \rho$, each component of which is a 1D array $u_j^i = (u_{jk}^i)$, $k \in \mathsf{Supp}(u_j^i)$, $1 \le i, j \le \rho$, where $\mathsf{Supp}(u_j^i) \subset \mathbb{Z}_0$, $1 \le i, j \le \rho$, and we denote $\mathsf{Supp}(\boldsymbol{u}) := \{(i, j, k) : k \in \mathsf{Supp}(u_j^i)\}$. Let $P_1 := \{i \in \mathbb{Z}_0 : 1 \le i \le \rho\}$ and $P_2 := \{j \in \mathbb{Z}_0 : 1 \le j \le \rho\}$. Over the direct product $\boldsymbol{\Sigma} := P_1 \times P_2 \times \mathbb{Z}_0$, a total order $\le_T$ is defined such that:

(1). $(i, j, k) \ne_T (i, j', k')$ if $(j, k) \ne (j', k')$, where $(i, j, k) =_T (i', j', k')$ means both $(i, j, k) \le_T (i', j', k')$ and $(i, j, k) \ge_T (i', j', k')$, but two or more elements of $\boldsymbol{\Sigma}$ can happen to be equal with resect to $\le_T$, in other words $(i, j, k) =_T (i', j', k')$, if $i \ne i'$;

(2). It satisfies a kind of admissibility condition:

- (a) If $k < k'$ then $(i, j, k) <_T (i, j, k')$;
- (b) If $(i, j, k) \le_T (i', j', k)$ for some k, then $(i, j, k) <_T (i', j', k)$ for any $k \in \mathbb{Z}_0$.

We call each element $\boldsymbol{f} = (f_i)_{1 \le i \le \rho}$ of the polynomial module R^ρ a polynomial vector. According to the total order $\le_T$ over $\boldsymbol{\Sigma}$, the number and degree of

a polynomial vector $\boldsymbol{f}$ is defined as follows: $\mathsf{num}(\boldsymbol{f}) = i$ and $\deg(\boldsymbol{f}) = s_i$ if $(i, \rho, s_i) >_T (i', \rho, s_{i'})$ for any $i' \neq i$, $1 \leq i' \leq \rho$, where $s_i := \deg(f_i)$ is the usual degree of the univariate polynomial f_i, $1 \leq i \leq \rho$. (In both left-hand and right-hand sides of the above inequality, the second components ρ can be replaced by any common $j, 1 \leq j \leq \rho$, provided that j is fixed.)

Now, we consider a compound linear recurrence corresponding to a polynomial vector $\boldsymbol{f} = (f_j)_{1 \leq j \leq \rho}$ with $\mathsf{num}(\boldsymbol{f}) = i$ and $\deg(\boldsymbol{f}) = s(:= s_i)$

$$\sum_{\lambda=1}^{\rho} \sum_{\kappa=0}^{s_j} f_{\lambda,\kappa} u^{\lambda}_{j,\kappa+k-s} = 0, \qquad 1 \leq j \leq \rho, s \leq k$$

which should be satisfied by the given array matrix $\boldsymbol{u} = [u^i_j]$, $1 \leq i, j \leq \rho$, where $f_j = \sum_{\kappa=0}^{s_j} f_{j,\kappa} x^{\kappa}$, $1 \leq j \leq \rho$. For a perfect array vector $\boldsymbol{u}$ with $\mathsf{Supp}(\boldsymbol{u}) = \boldsymbol{\Sigma}$,

$$\mathsf{Val}(\boldsymbol{u}) :=$$

$$\left\{ \boldsymbol{f} = (f_j) \in R^{\rho} : \sum_{\lambda=1}^{\rho} \sum_{\kappa=0}^{s_j} f_{\lambda,\kappa} u^{\lambda}_{j,\kappa+k-s} = 0, 1 \leq j \leq \rho, s := \deg(\boldsymbol{f}) \leq k \right\}$$

is an R-submodule of R^{ρ}, which has a Gröbner basis composed of ρ polynomial vectors $\{\boldsymbol{f}^1, \cdots, \boldsymbol{f}^{\rho}\}$, where $\mathsf{num}(\boldsymbol{f}^i) = i$, $1 \leq i \leq \rho$. We denote such a ρ-tuple of polynomial vectors simply as a polynomial matrix $\boldsymbol{F} = [f^i_j]$, $1 \leq i, j \leq \rho$, where $\boldsymbol{f}^i = (f^i_j)_{1 \leq j \leq \rho} \in R^{\rho}$, $1 \leq i \leq \rho$.

On decoding a one-point AG code, we get an array vector $\boldsymbol{u}$ from the syndromes. In this situation [**7, 10**], the total order $\leq_T$ over $\boldsymbol{\Sigma}$ is determined according to the pole numbers $o(f)$ of algebraic functions f on the curve defining the one-point AG code, where we have the ordered set of pole numbers

$$\mathcal{O} = \{o_i : i \in \mathbb{Z}_0\} = \{o_0(= 0) < o_1(= \rho) < o_2 < \cdots\}$$

which is an additive monoid with the identity element $o_0 = 0$. For a couple of fixed ρ-tuples of pole numbers $\{\bar{o}^{(i)} : 1 \leq i \leq \rho\}$ and $\{o^{(i)} : 1 \leq i \leq \rho\}$ such that $\bar{o}^{(i)} \equiv o^{(i)} \equiv i - 1 \bmod \rho, 1 \leq i \leq \rho$, a subset

$$\bar{\mathcal{O}} := \cup_{i=1}^{\rho} \{\bar{o}^{(i)} + o_j : j \in \mathbb{Z}_0\} = \{\bar{o}_l : l \in \mathbb{Z}_0\} = \{\bar{o}_0 < \bar{o}_1 < \bar{o}_2 < \cdots\}$$

of $\mathcal{O}$ is introduced, where each element $(i, j, k) \in \boldsymbol{\Sigma}$ corresponds to the pole number $\bar{o}_l = \bar{o}^{(i)} + o^{(j)} + k\rho \in \bar{\mathcal{O}}$, $1 \leq i, j \leq \rho$, $k \in \mathbb{Z}_0$, and the total order $\leq_T$ over $\overline{\boldsymbol{\Sigma}}$ is defined by the condition that $(i, j, k) \leq_T (i', j', k')$ if and only if $\bar{o}^{(i)} + o^{(j)} + k\rho \leq \bar{o}^{(i')} + o^{(j')} + k'\rho$. This correspondence determines a couple of partial mappings from the direct product $\mathbb{Z}_0 \times P_1$ into $P_2 = \{j : 1 \leq j \leq \rho\}$ and into $\mathbb{Z}_0 : j(l, i) = j$ and $k(l, i) = k$ for $(l, i) \in \mathbb{Z}_0 \times P_1$ if $\bar{o}_l = \bar{o}^{(i)} + o^{(j)} + k\rho \in \bar{\mathcal{O}}$. Thus, a finite array vector $\boldsymbol{u}^l$ is defined to have

$$\mathsf{Supp}(\boldsymbol{u}^l) = \boldsymbol{\Sigma}^l := \{(i, j, k) : \bar{o}^{(i)} + o^{(j)} + k\rho < \bar{o}_l\}$$

For a given finite array vector $\boldsymbol{u}^l$ we now define a minimal polynomial matrix $\boldsymbol{F} = [f^i_j]_{1 \leq i,j \leq \rho}$ of $\boldsymbol{u}^l$. This matrix is defined by the following conditions.

(a). Each component polynomial vector $\boldsymbol{f}^i = (f^i_j)_{1\le j\le \rho}$ with $\mathsf{num}(\boldsymbol{f}^i) = i$ and $\deg(\boldsymbol{f}^i) = s^i (:= s^i_i)$ is 'valid' for the array vector $\boldsymbol{u}$ in the sense that, for any $(i,j,k) \in \boldsymbol{\Sigma}^l$ such that $k \ge s^i$,

$$\boldsymbol{f}^i[\boldsymbol{u}]_{(j,k)} := \sum_{\lambda=1}^{\rho}\sum_{\kappa=0}^{s^i_j} f^i_{\lambda,\kappa} u^{\lambda}_{j,\kappa+k-s^i} = 0$$

(b). Each polynomial vector $\boldsymbol{f}^i$ has the minimum degree s^i among the valid polynomial vectors $\boldsymbol{f}$ with $\mathsf{num}(\boldsymbol{f}) = i$ for $\boldsymbol{u}$.

Using the above notation, the present BMS problem is stated as follows:

Given: an array vector $\boldsymbol{u} = \boldsymbol{u}^r$ for a fixed $r \in \mathbb{Z}_0$;
Find: a minimal polynomial matrix $\boldsymbol{F} = [f^i_j]$ of $\boldsymbol{u}$.

To give a BMS algorithm to solve this problem, we need some concepts which are similar but slightly different from those of the previous BMS algorithm. In the restriction

$$\overline{\Sigma}^{(i)} := \{(j,k) : (i,j,k) \in \boldsymbol{\Sigma}\}$$

of $\boldsymbol{\Sigma}$, the order $\mathsf{ord}(\boldsymbol{f})$, the span $\mathsf{span}(\boldsymbol{f})$, and the span number $\mathsf{snum}(\boldsymbol{f})$, with respect to the given array vector $\boldsymbol{u}$, of a polynomial vector $\boldsymbol{f}$ with $\mathsf{num}(\boldsymbol{f}) = i$ and $\deg(\boldsymbol{f}) = s$ are defined as follows:

Definition 1. $\mathsf{ord}(\boldsymbol{f}) = (\bar{\jmath},\bar{k}) \in \overline{\Sigma}^{(i)}$ *if* $\boldsymbol{f}[\boldsymbol{u}]_{(j,k)} = 0$ *for any* $(j,k) \in \overline{\Sigma}^{(i)}$ *such that* $(i,j,k) <_T (i,\bar{\jmath},\bar{k})$, $k \ge s$, *and* $\boldsymbol{f}[\boldsymbol{u}]_{(\bar{\jmath},\bar{k})} \ne 0$.

Definition 2. $\mathsf{span}(\boldsymbol{f}) := k'$ *and* $\mathsf{snum}(\boldsymbol{f}) = j'$ *if* $(j',k') = \mathsf{ord}(\boldsymbol{f}) - (0, \deg(\boldsymbol{f}))$.

Definition 3. $d_{\boldsymbol{f}} := \boldsymbol{f}[\boldsymbol{u}]_{(\bar{\jmath},\bar{k})}$ *is called the discrepancy of* $\boldsymbol{f}$, *if* $(\bar{\jmath},\bar{k}) = \mathsf{ord}(\boldsymbol{f})$.

At each iteration of the algorithm, we either keep or update polynomial matrices $\boldsymbol{F} := [f^i_j]$ and $\boldsymbol{G} := [g^i_j]$ with their component polynomial vectors

$$\boldsymbol{f}^i = (f^i_j)_{1\le j\le\rho} \qquad \boldsymbol{g}^i = (g^i_j)_{1\le j\le\rho}$$

such that $\mathsf{num}(\boldsymbol{f}^i) = i$, $\deg(\boldsymbol{f}^i) = s^i$, $\mathsf{snum}(\boldsymbol{g}^i) = i$, and $\mathsf{span}(\boldsymbol{g}^i) = c^i$ for $1 \le i \le \rho$. At every iteration, the identity $\#\Delta(\boldsymbol{F}) = \#\Delta(\boldsymbol{G})$ is kept invariant for

$$\#\Delta(\boldsymbol{F}) := \sum_{1\le i\le\rho}(s^i - 1) \qquad \#\Delta(\boldsymbol{G}) := \sum_{1\le i\le\rho} c^i$$

We denote $F := \{\boldsymbol{f}^i : 1 \le i \le \rho\}$ and $G := \{\boldsymbol{g}^i : 1 \le i \le \rho\}$. Further

(1) $\boldsymbol{f}^i \leftarrow \boldsymbol{g}^{\bar{\imath}}$ if $s^i \ge k(l,i) - c^{\bar{\imath}}$;

and

(2) $\boldsymbol{f}^i \to \boldsymbol{g}^{\bar{\imath}}$ if $s^i < k(l,i) - c^{\bar{\imath}}$, where $\bar{\imath} = j(l,i)$ for $(l,i) \in \mathbb{Z}_0 \times P_1$, and $s^i = \deg(\boldsymbol{f}^i)$, $c^{\bar{\imath}} = \mathsf{span}(\boldsymbol{g}^{\bar{\imath}})$.

Algorithm 3:

Step 1: $l := 0$; $s^i := 0$, $1 \le i \le \rho$; $f_i^i := 1$, $1 \le i \le \rho$; $f_j^i := 0$, $i \ne j$, $1 \le i, j \le \rho$; $c^i := \varnothing$, $1 \le i \le \rho$; $g_j^i := \varnothing$, $1 \le i, j \le \rho$;

Step 2: for each i, $1 \le i \le \rho$, such that $j(l,i) \ne \varnothing$, compute

$$d_{\boldsymbol{f}^i} := \boldsymbol{f}^i[\boldsymbol{u}]_{(\bar{\imath},k)}$$

for $(i, \bar{\imath}, k) \in \boldsymbol{\Sigma}^l$ such that $\bar{\imath} := j(l,i), k = k(l,i)$; Further

$$F_N := \{\boldsymbol{f}^i \in F : \mathrm{ord}(\boldsymbol{f}^i) = (\bar{\imath}, k)\};$$

Step 3 (updating):

(a) for each pair $\boldsymbol{f}^i \in F_N$ and $\boldsymbol{g}^{\bar{\imath}}$ such that $\bar{\imath} = j(l,i)$ and $\boldsymbol{f}^i \leftarrow \boldsymbol{g}^{\bar{\imath}}$,

$$\boldsymbol{f}^i := \boldsymbol{f}^i - \frac{d_{\boldsymbol{f}^i} d_{\boldsymbol{g}^{\bar{\imath}}}}{x}^{s^i - (k(l,i) - c^{\bar{\imath}})} \boldsymbol{g}^{\bar{\imath}};$$

(b) for each pair $\boldsymbol{f}^i \in F_N$ and $\boldsymbol{g}^{\bar{\imath}}$ such that $\bar{\imath} = j(l,i)$ and $\boldsymbol{f}^i \to \boldsymbol{g}^{\bar{\imath}}$

$$\boldsymbol{f}^{+i} := x^{(k(l,i) - c^{\bar{\imath}}) - s^i} \boldsymbol{f}^i - \frac{d_{\boldsymbol{f}^i}}{d_{\boldsymbol{g}^{\bar{\imath}}}} \boldsymbol{g}^{\bar{\imath}}, \ \boldsymbol{g}^{\bar{\imath}} := \boldsymbol{f}^i; \boldsymbol{f}^i := \boldsymbol{f}^{+i};$$

(c) for each $\boldsymbol{f}^i \in F_N$ such that $\boldsymbol{g}^{\bar{\imath}} = \varnothing$ with $\bar{\imath} = j(l,i)$,

$$\boldsymbol{f}^{+i} := x^{k(l,i) - s^i + 1} \boldsymbol{f}^i, \ \boldsymbol{g}^{\bar{\imath}} := \boldsymbol{f}^i; \ \boldsymbol{f}^i := \boldsymbol{f}^{+i};$$

Step 4: $l := l + 1$; if $l < r$ then go to Step 2 else stop.

For this algorithm the basic lemma is stated as follows:

Lemma 1. *Suppose that* $\boldsymbol{f}^i \in F$ *with* $\mathrm{num}(\boldsymbol{f}) = i$ *and* $\deg(\boldsymbol{f}) = s^i$ *fails to be valid at the iteration* l, *and let* $\bar{\imath} := j(l,i)$. *Then, we have the following facts:*

(a). *There is a polynomial vector* $\boldsymbol{h}$ *with* $\mathrm{num}(\boldsymbol{h}) = i$ *and* $\deg(\boldsymbol{h}) = \deg(\boldsymbol{f})$ *which is valid for* $\boldsymbol{u}^{l+1}$ *if and only if there exists* $\boldsymbol{g} \in G$ *with* $\mathrm{snum}(\boldsymbol{g}) = \bar{\imath}$ *and* $\mathrm{span}(\boldsymbol{f}) \le \mathrm{span}(\boldsymbol{g})$;

(b). *There is no polynomial vector* $\boldsymbol{h}$ *with* $\mathrm{num}(h) = i$ *and* $\deg(\boldsymbol{h}) < k(l,i) - c\bar{\imath}$ *which is valid for* $\boldsymbol{u}^{l+1}$ *if* $c^{\bar{\imath}} < k(l,i) - s^i$ *for* $\boldsymbol{g} \in G$ *with* $\mathrm{snum}(\boldsymbol{g}) = \bar{\imath}$ *and* $\mathrm{span}(\boldsymbol{g}) = c^{\bar{\imath}}$.

In the case of $\rho = 1$, this algorithm reduces to the BM algorithm. The identity $\#\Delta(\boldsymbol{F}) = \#\Delta(\boldsymbol{G})$ corresponds to $s = c + 1$ in the BM algorithm. Although Algorithm 3 has computational complexity of $\mathcal{O}(\rho r^2)$, one might want to have

a much faster algorithm by making a recursive version which is a multidimensional extension of the recursive BM algorithm by Blahut [1].

5. A recursive BMS algorithm

We can modify the BMS algorithm over polynomial modules (Algorithm 3) to get not only an accelerated version but also a recursive BMS algorithm along the same line as Blahut's recursive BM algorithm, which has asymptotically better complexity. Of course, it is of a theoretical interest.

From now on we denote $\boldsymbol{f}^i, \boldsymbol{g}^{\bar{\imath}}, s^i, c^{\bar{\imath}}$ at the beginning of Step 3 of the l-th iteration of Algorithm 3 as $\boldsymbol{f}^i(l), \boldsymbol{g}^{\bar{\imath}}(l), s^i(l), c^{\bar{\imath}}(l)$, respectively, and let

$$\boldsymbol{f}^i(l+1) := \boldsymbol{f}^{+i} \qquad \text{for } \boldsymbol{f}^i = \boldsymbol{f}^i(l)$$

Now, we define another couple of mappings from $\mathbb{Z}_0 \times P_1$ into P_2 and into $\mathbb{Z}_0$: $\check{\jmath}(l,i) = j$ and $\check{k}(l,i) = k$ for $(l,i) \in \mathbb{Z}_0 \times P_1$ if $\bar{o}_l = \bar{o}^{(i)} + k\rho - (\rho - j)$.

Remark. Notice that $\check{\jmath}(l,i) = j(l,i)$; $\check{k}(l+1,i) = \check{k}(l,i) + 1$ if $j(l,i) = \rho$ and $\check{k}(l+1,i) = \check{k}(l,i)$ otherwise.

Further $\check{k}(l,i)$ is monotonically increasing with respect to l for a fixed i, though $k(l,i)$ does not always have such monotonicity. At the l-th iteration of Algorithm 3 we have two sets of polynomial vectors $\boldsymbol{f}^i(l)$ with

$$\deg(\boldsymbol{f}^i(l)) \;=\; s^i(l), \qquad 1 \le i \le \rho$$

and $\boldsymbol{g}^{\bar{\imath}}(l)$ with

$$\mathsf{span}(\boldsymbol{g}^{\bar{\imath}}(l)) \;=\; c^{\bar{\imath}}(l), \qquad 1 \le \bar{\imath} \le \rho$$

For them, we introduce the degree-raised polynomial vectors with

$$\check{s}^i(l) \;:=\; \deg(\check{\boldsymbol{f}}^i(l)) = \check{k}(l,i) \qquad\qquad \check{\boldsymbol{f}}^i(l) \;:=\; x^{\check{k}(l,i)-s^i(l)}\boldsymbol{f}^i(l)$$

Their component polynomials are $\check{f}^i_j(l) = \sum_{k=\check{k}(l,i)-s^i(l)}^{\check{k}(l,i)} \check{f}^i_{jk}(l)x^k$, where

$$\check{f}^i_{jk}(l) \;=\; f^i_{j,k+\check{k}(l,i)-s^i(l)}(l), \qquad 0 \le k \le s^i(l)$$

When the polynomial vectors $\boldsymbol{f}^i(l)$ and $\boldsymbol{g}^{\bar{\imath}}(l)$ are updated, the raised polynomial vectors are updated as follows:

$$\check{\boldsymbol{f}}^i(l+1) := \quad x^{\check{k}(l+1,i)-\check{k}(l,i)}(\check{\boldsymbol{f}}^i(l) - \frac{d\boldsymbol{f}^i(l)}{d\boldsymbol{g}^{\bar{\imath}}(l)}\check{\boldsymbol{g}}^{\bar{\imath}}(l)), \tag{4}$$

$$\check{\boldsymbol{g}}^{\bar{\imath}}(l+1) := \qquad\qquad \check{\boldsymbol{g}}^{\bar{\imath}}(l) \text{ or } \check{\boldsymbol{f}}^i(l), \tag{5}$$

where (5) depends on the case. (The rightmost term of (4), $\boldsymbol{g}^{\bar{\imath}}(l) = \boldsymbol{f}^{i'}(l')$ for some pair of $l' < l$ and $1 \le i' \le \rho$, where $\bar{\imath} = j(l,i) = j(l',i')$.)

Consequently, at the l-th iteration, we have two $\rho \times \rho$ polynomial matrices $\check{\boldsymbol{F}}(l) := [\check{f}^i_j(l)]_{1\le i,j\le\rho}$, $\check{\boldsymbol{G}}(l) = [\check{g}^i_j(l)]_{1\le i,j\le\rho}$ and a ρ-tuple of discrepancies

$$\Delta^i(l) \;:=\; d_{\boldsymbol{f}^i(l)} \;=\; \sum_{\lambda=1}^{\rho} \sum_{\kappa=k(l,i)-s(l,i)}^{k(l,i)} \check{f}^i_{\lambda\kappa}(l)u^{\lambda}_{\bar{\imath},\kappa}, \qquad 1 \le i \le \rho$$

Summarizing them, we have a discrepancy vector $\boldsymbol{\Delta}(l) := [\Delta^i(l)]_{1\le i\le \rho}$ of elements of the symbol field $\mathbb{F}_q$ and a $2\rho \times \rho$ matrix of polynomials

$$\begin{bmatrix} \check{\boldsymbol{F}}(l) \\ \check{\boldsymbol{G}}(l) \end{bmatrix}$$

The updating and/or keeping of polynomial matrices at the l-th iteration are rewritten as

$$\begin{bmatrix} \check{\boldsymbol{F}}(l+1) \\ \check{\boldsymbol{G}}(l+1) \end{bmatrix} = \Phi(l) \cdot \begin{bmatrix} \check{\boldsymbol{F}}(l) \\ \check{\boldsymbol{G}}(l) \end{bmatrix}$$

where $\boldsymbol{\Phi}(l)$ is a $2\rho \times 2\rho$ polynomial matrix. The above equation is made up of the following formulas

$$\begin{bmatrix} \check{\boldsymbol{f}}^i(l+1) \\ \check{\boldsymbol{g}}^{\bar{\imath}}(l+1) \end{bmatrix} = \begin{bmatrix} \epsilon^i(l) & -\Delta^i(l)\epsilon^i(l) \\ \frac{1}{\Delta^i(l)}\delta^i(l) & 1-\delta^i(l) \end{bmatrix} \begin{bmatrix} \check{\boldsymbol{f}}^i(l) \\ \check{\boldsymbol{g}}^{\bar{\imath}}(l) \end{bmatrix} \tag{6}$$

where $\bar{\imath} = j(l,i)$, $\epsilon^i(l) := x^{\check{k}(l+1,i)-\check{k}(l,i)} (= x \text{ or } 1)$, and $\delta^i(l) := 1$ if $\Delta^i(l) \neq 0$; $\delta^i(l) := 0$ if $\Delta^i(l) = 0$, $1 \le i \le \rho$. The iteration formula (6) is quite similar to that of Blahut's BM algorithm with some difference.

To derive a recursive algorithm, we consider the matrix products

$$\boldsymbol{\Psi}(l) := \prod_{k=1}^{l} \boldsymbol{\Phi}(k) \quad \text{and} \quad \boldsymbol{\Psi}(l,l') := \prod_{k=l'+1}^{l} \boldsymbol{\Phi}(k)$$

for $l' < l$. Then, we have $\boldsymbol{\Psi}(l) = \boldsymbol{\Psi}(l,l') \cdot \boldsymbol{\Psi}(l')$. On the other hand, we have

$$\boldsymbol{\Delta}(l) = \boldsymbol{\Psi}_{11}(l,l')\boldsymbol{\Delta}_1(l') + \boldsymbol{\Psi}_{12}(l,l')\boldsymbol{\Delta}_2(l')$$

and $\boldsymbol{\Delta}_1(l') = \boldsymbol{\Psi}_{11}(l') * \boldsymbol{u}$, $\boldsymbol{\Delta}_2(l') = \boldsymbol{\Psi}_{21}(l') * \boldsymbol{u}$, where $*$ implies a convolution, and the $\rho \times \rho$ matrices $\boldsymbol{\Psi}_{11}(l,l')$, $\boldsymbol{\Psi}_{12}(l,l')$ are submatrices of

$$\boldsymbol{\Psi}(l,l') = \begin{bmatrix} \boldsymbol{\Psi}_{11}(l,l') & \boldsymbol{\Psi}_{12}(l,l') \\ \boldsymbol{\Psi}_{21}(l,l') & \boldsymbol{\Psi}_{22}(l,l') \end{bmatrix}$$

while $\boldsymbol{\Psi}_{11}(l')$, $\boldsymbol{\Psi}_{21}(l')$ are submatrices of

$$\boldsymbol{\Psi}(l') = \begin{bmatrix} \boldsymbol{\Psi}_{11}(l') & \boldsymbol{\Psi}_{12}(l') \\ \boldsymbol{\Psi}_{21}(l') & \boldsymbol{\Psi}_{22}(l') \end{bmatrix}$$

The complexity of computation comes from a number of polynomial multiplications, which are equivalent to convolutions. To reduce those loads, we construct a recursive algorithm by 'divide and conquer' along the line similar to [1]. The largest possible length (degree) of polynomials which appear at the end of the BMS algorithm is $\frac{r}{\rho}$. For simplicity, let r and ρ be 2^σ and 2^τ, respectively. Then, the recursive algorithm works by splitting the whole problem of 2^σ iterations into two problems, each of $2^{\sigma-1}$ iterations, which are linked together by $8\rho^3$ convolutions of length $2^{\sigma-\tau-1}$ contained in the polynomial matrix multiplication $\boldsymbol{\Psi}(2^\sigma) = \boldsymbol{\Psi}(2^\sigma, 2^{\sigma-1}) \cdot \boldsymbol{\Psi}(2^{\sigma-1})$. Each of the two half problems is, in turn, split into two problems. This process continues until problems of only one iterations are reached (Of course, since we cannot have any convolution

of length less than 1, we must be stucked at a convolution of length 1, i.e., one multiplication of elements of $\mathbb{F}_q$ during the last several stages of the above splitting process). Since the half problems can be made to look the same as the original problem, a recursive implementation can be realized. Thus, the recursive algorithm, which we call procedure rBMS, can be designed similar to [1]. The procedure rBMS uses itself, and it has a push-down stack, where it can temporarily store a copy of data at one level of the recursion when it proceeds to the next level. The next level behaves the same way, pushing down yet another set of data, and so on. Eventually the problem segment has only one iteration and cannot be divided in half. In total, the recursion will go σ levels deep. The η-th level will be visited 2^η times. At the η-th level is the matrix product $\Psi(2^{\sigma-\eta}, 2^{\sigma-\eta-1}) \cdot \Psi(2^{\sigma-\eta-1})$ and the discrepancy vector calculation, both of which require $8\rho^3$ convolutions of length $2^{\sigma-\tau-\eta-1}$. In total at level η there are $\rho^3 2^{\eta+4}$ convolutions of length $2^{\sigma-\tau-\eta-1}$. The computational complexity of $\rho^3 2^{\eta+4}$ convolutions of length $2^{\sigma-\tau-\eta-1}$ is greatest when $\eta = 0$, and thus the total complexity of the recursive BMS algorithm is $\mathcal{O}(\rho^3\sigma)C(2^{\sigma-\tau})$, where $C(d)$ is the number of multiplications required to calculate a convolution of polynomials of length (degree) d. In [1], it is shown that the asymptotic complexity of $C(d)$ is $\mathcal{O}(d2^{\log^* d})$, where $\log^* d$ is the minimum number of iterations in $\log_2(\log_2(\cdots(\log_2 d)\cdots)) \leq 1$. The term $2^{\log^* d}$ is less than $\log d$. Summarizing the above arguments, we have the total complexity approximate to $\mathcal{O}(\rho^2 r(\log r)(\log \frac{r}{\rho}))$. In case of codes from an algebraic curve in the N-dimensional projective space, where $\mathcal{O}(\rho) = \mathcal{O}(r^{1-\frac{2}{N+1}})$, the complexity is almost $\mathcal{O}(r^{3-\frac{4}{N+1}})$, which is less than $\mathcal{O}(\rho r^2) = \mathcal{O}(r^{3-\frac{2}{N+1}})$ of the original BMS algorithm. In the above we does not use the Strassen fast method of matrix multiplication, by which we have complexity $\rho^{\log_2 7}$ instead of ρ^3. If we invoke it, we have a better complexity $\mathcal{O}(\rho^{\log_2 \frac{7}{2}} r(\log r)(\log \frac{r}{\rho}))$.

6. Concluding remarks

In this paper we have reviewed the BM algorithms and the BMS algorithms in several forms, and some intrinsic relationships among them. Those algorithms have been found to be useful to get faster decoding methods of algebraic codes, including BCH codes, RS codes, and AG codes [**9**, **8**, **11**]. The main role of all these algorithms is in finding the error locators. Thus, from the aspects of decoding, the BMS algorithm can be regarded as quite a natural generalization of the BM algorithm, except for some extra tasks of majority voting for finding the unknown syndrome values to decode AG codes up to the Feng-Rao designed distance. Some other problems of decoding, i.e., soft decision decoding [**11**], parallelization [**10**], etc. can be treated similarly. From these considerations, we make sure again that R. E. Blahut's unifying view on various algebraic methods towards fast decoding [**2**] is very important to have good and practically significant error correcting codes. As a byproduct, we have given a recursive BMS algorithm which can be obtained from his approach [**1**].

References

[1] R.E. BLAHUT, *Theory and Practice of Error Control Codes*, Chapter 11: Fast Algorithms, pp. 308–346, Reading, MA: Addison Wesley 1983.

[2] ———, *Algebraic Methods for Signal Processing and Communications Coding*, New York: Springer-Verlag 1992.

[3] E.R. BERLEKAMP, *Algebraic Coding Theory*, New York: McGraw-Hill 1968.

[4] J.L. MASSEY, Shift-register synthesis and BCH decoding, *IEEE Trans. Inform. Theory*, vol. 15, pp. 122–127, 1969.

[5] S. SAKATA, Finding a minimal set of linear recurring relations capable of generating a given finite 2D cyclic codes and their duals, *J. Symbolic Computation*, vol. 5, pp. 1321–1337, 1988.

[6] ———, Extension of the BM algorithm to N dimensions, *Information and Computation*, vol. 84, pp. 207–239, 1990.

[7] ———, A vector version of the BMS algorithm for implementing fast erasure-and-error decoding of one-point AG codes, submitted to *Appl. Algebra Engrg. Comm. Comput.*

[8] S. SAKATA, H. ELBRØND JENSEN, AND T. HØHOLDT, Generalized Berlekamp-Massey decoding of algebraic-geometric codes up to half the Feng-Rao bound," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1762–1769, 1995.

[9] S. SAKATA, J. JUSTESEN, Y. MADELUNG, H. ELBRØND JENSEN, AND T. HØHOLDT, Fast decoding of algebraic geometric codes up to the designed distance, *IEEE Trans. Inform. Theory*, vol. 41, pp. 1672-1677, 1995.

[10] S. SAKATA AND M. KURIHARA, A systolic array architecture for implementing a fast parallel decoding algorithm of one-point AG codes, submitted to *IEEE Trans. Inform. Theory.*

[11] S. SAKATA, D. LEONARD, H. ELBRØND JENSEN, AND T. HØHOLDT, Fast erasure-and-error decoding of AG codes up to the Feng-Rao bound, submitted to *IEEE Trans. Inform. Theory.*

Lee Weights of Codes from Elliptic Curves

José Felipe Voloch

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF TEXAS
AUSTIN, TX 78712, USA
voloch@math.utexas.edu

Judy L. Walker

DEPARTMENT OF MATHEMATICS AND STATISTICS
UNIVERSITY OF NEBRASKA
LINCOLN, NE 68588-0323, USA
jwalker@math.unl.edu

ABSTRACT. In [**15**], the second author defined algebraic geometric codes over rings. This definition was motivated by two recent trends in coding theory: the study of algebraic geometric codes over finite fields, and the study of codes over rings. In that paper, many of the basic parameters of these new codes were computed. However, the Lee weight, which is very important for codes over the ring $\mathbb{Z}/4\mathbb{Z}$, was not considered. In [**14**], this weight measure, as well as the more general Euclidean weight for codes over $\mathbb{Z}/p^l\mathbb{Z}$, is considered for algebraic geometric codes arising from elliptic curves. In this paper, we will focus on the specific case of codes over $\mathbb{Z}/4\mathbb{Z}$ and we will show how everything works in an explicit example.

1. Introduction

The study of linear codes over finite rings has received much attention lately. This is due in large part to the paper of Hammons, Kumar, Calderbank, Sloane, and Solé [**2**], in which it is shown that certain nonlinear binary codes are actually the images of linear codes over the ring $\mathbb{Z}/4\mathbb{Z}$ under the Gray map. This idea has prompted many authors (see [**1**] or [**11**], for example) to consider the question of how to construct linear codes over $\mathbb{Z}/4\mathbb{Z}$, with the hope that these codes might have good binary images under the Gray map.

In [**15**], a new method of constructing linear codes over $\mathbb{Z}/4\mathbb{Z}$ is proposed. The idea there is to generalize the construction of algebraic geometric codes over finite fields ([**12**], [**13**]) to allow the use of a local Artin ring to play the role of the finite field. In that paper, the length, dimension (rank), and minimum

The first author was supported in part by NSA Grant #MDA904-97-1-0037 and TARP Grant #ARP-006. The second author was supported in part by NSF Grant #DMS-9709388.

Hamming distance of these new codes are computed. However, the crucial Lee weight, which is the same as the Hamming weight of their binary images under the Gray map, is not treated there.

It turns out that computing the minimum Lee weight of an algebraic geometric code is very difficult. In [**16**], it is shown how the notion of Lee weight for codes over $\mathbb{Z}/4\mathbb{Z}$ generalizes to a weight measure called Euclidean weight for codes over $\mathbb{Z}/p^l\mathbb{Z}$, where p is any prime and $l \geq 1$ is an integer. Also in that paper, the minimum Euclidean weight of an algebraic geometric code over $\mathbb{Z}/p^l\mathbb{Z}$ is expressed in terms of an exponential sum.

In [**14**], a bound on this sum is found in the special case that the curve over which the code is defined is a certain type of elliptic curve. In this paper, we will examine in detail the case of [**14**] where $p = l = 2$, so that the code in question is a code over $\mathbb{Z}/4\mathbb{Z}$. To illustrate our results, we work a specific example in detail.

2. Galois and Witt rings, and algebraic geometric codes over them

A common method of constructing long codes over small fields is to construct a code over an extension field and then consider the trace code. A similar approach can be taken to construct codes over the ring $\mathbb{Z}/4\mathbb{Z}$ by first constructing codes over a *Galois ring* $\mathrm{GR}(4,m)$, and then considering the trace code. Recall (cf. [**2**]) that the Galois ring $\mathrm{GR}(4,m)$ is simply $(\mathbb{Z}/4\mathbb{Z})[x]/(f)$, where $f \in (\mathbb{Z}/4\mathbb{Z})[x]$ is monic, irreducible, and a divisor of the polynomial $x^{2^m-1}-1$. More generally, the Galois ring $\mathrm{GR}(p^l,m)$ is $(\mathbb{Z}/p^l\mathbb{Z})[x]/(f)$, where $f \in (\mathbb{Z}/p^l\mathbb{Z})[x]$ is monic and irreducible and divides the polynomial $x^{p^m-1}-1$. Notice that $\mathrm{GR}(p^l,1) \simeq \mathbb{Z}/p^l\mathbb{Z}$. Recall that the *Teichmüller set* of $\mathrm{GR}(p^l,m)$ is the multiplicative lifting of the field $\mathbb{F}_{p^m}$ living inside $\mathrm{GR}(p^l,m)$.

It is often more convenient to think of the ring $\mathrm{GR}(4,m)$ as the ring $W_2(\mathbb{F}_{2^m})$ of 2-dimensional *Witt vectors* over the field $\mathbb{F}_{2^m}$. As a set, this ring looks like vectors of length 2 with coordinates in $\mathbb{F}_{2^m}$, and the operations are given as follows:

$$\begin{aligned}(a_0,a_1)+(b_0,b_1) &= (a_0+b_0, a_1+b_1+a_0b_0)\\ (a_0,a_1)\cdot(b_0,b_1) &= (a_0b_0, a_0^2b_1+b_0^2a_1)\end{aligned}$$

This ring may be thought of as the quotient modulo 4 of $W(\mathbb{F}_{2^m})$, the ring of infinite length Witt vectors over $\mathbb{F}_{2^m}$ (see [**5**]). Also, one can consider Witt rings over fields of characteristic other than 2, or of finite lengths other than 2.

Notice that the map

$$W_2(\mathbb{F}_2) \mapsto \mathbb{Z}/4\mathbb{Z} \tag{1}$$

given by

$$\begin{aligned}(0,0) &\mapsto 0\\ (1,0) &\mapsto 1\\ (0,1) &\mapsto 2\\ (1,1) &\mapsto 3\end{aligned}$$

is an isomorphism of rings. More generally, $W_l(\mathbb{F}_p) \simeq \mathbb{Z}/p^l\mathbb{Z}$ for any prime p and any positive integer l. Similarly, $W_2(\mathbb{F}_{2^m}) \simeq \mathrm{GR}(4,m)$ for all m and more

generally $W_l(\mathbb{F}_{p^m}) \simeq \mathrm{GR}(p^l, m)$. The Teichmüller set of $\mathrm{GR}(p^l, m)$ corresponds to the vectors in $W_l(\mathbb{F}_{p^m})$ which are zero beyond the first coordinate.

The map $F : W_l(\mathbb{F}_{p^m}) \to W_l(\mathbb{F}_{p^m})$ given by $(a_0, \ldots, a_{l-1}) \mapsto (a_0^p, \ldots, a_{l-1}^p)$ is an additive endomorphism of order m. The trace map $T : W_l(\mathbb{F}_{p^m}) \to W_l(\mathbb{F}_p) \simeq \mathbb{Z}/p^l\mathbb{Z}$ is given by $T(a) = a + F(a) + F^2(a) + \cdots + F^{m-1}(a)$.

The generalization of algebraic geometric codes which we will describe below will make sense for any *local Artin* ring A. Recall that a local ring is one with only one maximal ideal. $\mathrm{GR}(p^l, m)$ is local with its unique maximal ideal being the one generated by p. In $W_l(\mathbb{F}_{p^m})$, this ideal consists of the elements having zeros in the first coordinate. An Artin ring is one in which every descending chain of ideals is eventually stable; since any finite ring obviously satisfies this property, $\mathrm{GR}(p^l, m) \simeq W_l(\mathbb{F}_{p^m})$ is an Artin ring.

Next, we describe how to generalize the construction of algebraic geometric codes over finite fields ([**13**], [**12**]) to give codes over local Artin rings such as $W_l(\mathbb{F}_{p^m}) \simeq \mathrm{GR}(p^l, m)$. The set-up for the generalized construction is as follows:

Let A be a local Artin ring, and let $\mathbf{X}$ be a curve defined over A (i.e., a connected irreducible scheme over Spec$A$ which is smooth and of relative dimension one). One may think of $\mathbf{X}$ as being defined by polynomial equations with coefficients in A. In this case, A-points on $\mathbf{X}$ are solutions in A to the equations defining $\mathbf{X}$. Letting $\mathfrak{m}$ be the maximal ideal of A, we get a curve X over the field $A/\mathfrak{m}$ by reducing the equations defining $\mathbf{X}$ modulo $\mathfrak{m}$. By reducing the coordinates of an A-point of $\mathbf{X}$ modulo $\mathfrak{m}$, we get a $A/\mathfrak{m}$-rational point of X. We say two A-points of $\mathbf{X}$ are *disjoint* if their reductions modulo $\mathfrak{m}$ give two distinct $A/\mathfrak{m}$-rational points of X. Let $\mathcal{Z}$ be a set of disjoint A-points on $\mathbf{X}$. Associated to a line bundle $\mathcal{L}$ on $\mathbf{X}$ is an A-module $\Gamma(\mathbf{X}, \mathcal{L})$ of rational functions on $\mathbf{X}$, and $\mathcal{L}$ may easily be chosen so that it makes sense to evaluate the functions in $\Gamma(\mathbf{X}, \mathcal{L})$ at the points of $\mathcal{Z}$.

With A, $\mathbf{X}$, $\mathcal{Z} = \{Z_1, \ldots, Z_n\}$, and $\mathcal{L}$ as above, the algebraic geometric code $\mathbb{C}_A(\mathbf{X}, \mathcal{Z}, \mathcal{L})$ is defined to be the image of the evaluation map

$$\begin{aligned} \Gamma(\mathbf{X}, \mathcal{L}) &\to A^n \\ f &\mapsto (f(Z_1), \ldots, f(Z_n)) \end{aligned}$$

Some of the properties of these codes are summarized in the following theorem. We refer the reader to [**15**] for the proofs.

Theorem 1. *Let $\mathbf{X}$, X, $\mathcal{L}$, and $\mathcal{Z} = \{Z_1, \ldots, Z_n\}$ be as above. Let g denote the genus of X, and suppose $2g - 2 < \deg \mathcal{L} < n$. Set $\mathbb{C} = \mathbb{C}(\mathbf{X}, \mathcal{Z}, \mathcal{L})$. Then $\mathbb{C}$ is a linear code of length n over A, and is free as an A-module. The dimension (rank) of $\mathbb{C}$ is $k = \deg \mathcal{L} + 1 - g$, and the minimum Hamming distance of $\mathbb{C}$ is at least $n - \deg \mathcal{L}$. Further, under the additional assumption that A is Gorenstein, the class of algebraic geometric codes is closed under taking duals. In particular, there exists a line bundle $\mathcal{E}$ such that $\mathbb{C}^\perp = \mathbb{C}(\mathbf{X}, \mathcal{Z}, \mathcal{E})$.*

Remark. The rings $W_l(\mathbb{F}_p) \simeq \mathbb{Z}/p^l\mathbb{Z}$ and $W_l(\mathbb{F}_{p^m}) \simeq \mathrm{GR}(p^l, m)$ are Gorenstein, so everything in the above theorem applies to these rings in particular.

In practice, one is usually most interested in codes over the ring $\mathbb{Z}/4\mathbb{Z}$. Such a code can be obtained in two ways using the construction above. First, one can simply set $A = \mathbb{Z}/4\mathbb{Z}$ and construct an algebraic geometric code over $\mathbb{Z}/4\mathbb{Z}$ directly. The drawback of this is that the codes obtained in this manner will be short. A second method is to set $A = W_2(\mathbb{F}_{2^m}) = \mathrm{GR}(4, m)$, construct an algebraic geometric code over A, and then apply the trace map $T : A \to \mathbb{Z}/4\mathbb{Z}$ coordinatewise to the code to get a code over $\mathbb{Z}/4\mathbb{Z}$. Since $T : \mathbb{Z}/4\mathbb{Z} \to \mathbb{Z}/4\mathbb{Z}$ is the identity map, we need only consider the second of these two constructions.

3. Lee weight

Theorem 1 gives the length, dimension, and minimum Hamming distance of an algebraic geometric code over a ring. However, for codes over $\mathbb{Z}/4\mathbb{Z}$, the relevant weight measure is the Lee weight. The reason for this is that the Gray map is an isometry

$$\Big\{(\mathbb{Z}/4\mathbb{Z})^n, \text{Lee weight}\Big\} \;\to\; \Big\{(\mathbb{F}_2)^{2n}, \text{Hamming weight}\Big\}$$

so that the minimum Hamming weight of the binary image of a $\mathbb{Z}/4\mathbb{Z}$-code is the Lee weight of the code over $\mathbb{Z}/4\mathbb{Z}$. Our goal is to find the minimum Lee weight of (trace codes of) algebraic geometric codes over $\mathbb{Z}/4\mathbb{Z}$.

Recall that the Lee weight is defined on $\mathbb{Z}/4\mathbb{Z}$ by $w_L(0) = 0$, $w_L(1) = w_L(3) = 1$, and $w_L(2) = 2$. The Lee weight of a vector in $(\mathbb{Z}/4\mathbb{Z})^n$ is the sum of the Lee weights of the coordinates of the vector. Our first task is to find an "algebraic" expression for Lee weight, so that it will be easier to study.

Notice that for each $x \in \mathbb{Z}/4\mathbb{Z}$, $w_L(x)$ is exactly half the square of the distance in the complex plane between $1 = i^0$ and i^x, where $i = \sqrt{-1}$. Using a little trigonometry, we see that $w_L(x) = 1 - \cos(\frac{\pi}{2}x) = 1 - \mathrm{Re}(e^{\pi i x/2})$. Therefore, for a vector $\mathbf{x} = (x_1, \ldots, x_n)$ in $(\mathbb{Z}/4\mathbb{Z})^n$, we have

$$w_L(\mathbf{x}) \;=\; \sum_{j=1}^{n} \Big(1 - \mathrm{Re}(e^{\pi i x_j/2})\Big) \;\geq\; n - \left|\sum_{j=1}^{n} e^{\pi i x_j/2}\right|$$

If the vector $\mathbf{x}$ is actually a codeword in a trace code of an algebraic geometric code, then we have

$$\mathbf{x} \;=\; (T(f(Z_1)), \ldots, T(f(Z_n)))$$

where $Z_1, \ldots, Z_n$ are disjoint A-points on some curve $\mathbf{X}$ over $A = W_2(\mathbb{F}_{2^m}) \simeq \mathrm{GR}(4, m)$ and f is a global section of a line bundle on $\mathbf{X}$. In this case, we have

$$w_L(\mathbf{x}) \;\geq\; n - \left|\sum_{j=1}^{n} e^{\pi i T(f(Z_j))/2}\right|$$

So we see that finding a lower bound on the minimum Lee weight of the trace code of an algebraic geometric code amounts to finding an upper bound on the modulus of an exponential sum of the form

$$\sum_{j=1}^{n} e^{\pi i T(f(Z_j))/2} \tag{2}$$

In the case of the projective line, sums similar to this have received much attention recently; see [4] and [6]. In both cases, the A-points on $\mathbb{P}^1$ over which the sum is taken are not arbitrary but instead are the Teichmüller points of A. An analogous concept exists for certain elliptic curves over rings, and it is here where we will focus our attention for the remainder of this paper.

4. Ordinary elliptic curves and canonical lifts

An elliptic curve over a field of characteristic 2 is called *ordinary* if its group of 2-torsion points has order 2. (Otherwise, its group of 2-torsion points is trivial, and the curve is called *supersingular.*) Every elliptic curve has an isogeny called the *Frobenius*; on points, the Frobenius takes (x, y) to (x^2, y^2). A special case of Serre-Tate theory (see [7] or [3]) implies that every ordinary elliptic curve E over a finite field k of characteristic 2 has a *canonical lift* to an elliptic curve over the ring of infinite length Witt vectors $W(k)$. By reducing modulo 4, we get that E has a canonical lift to an elliptic curve $\mathbf{E}$ over $W_2(k)$. The existence of a canonical lift is in fact equivalent to the existence of an injective homomorphism $\tau : E(k) \to \mathbf{E}(W_2(k))$, called the *elliptic Teichmüller lift.* $W_2(k)$-points Z on $\mathbf{E}$ which are of the form $Z = \tau(P)$ for some k-rational point P on E are the analogies of the Teichmüller points on $\mathbb{P}^1$.

Our first question, then, is: How can we tell if an elliptic curve $\mathbf{E}$ over $W_2(k)$ is actually the canonical lift of an ordinary elliptic curve over k? The answer to this question is found in the next proposition, which is true for any characteristic $p > 0$ and is proven in [14]. Before we can state it, however, we need one more definition. Let $\mathbf{X}$ be a scheme over $W_l(k)$, where k is a field and $l \geq 0$. The *Greenberg transform* $G(\mathbf{X})$ is the variety over k formed by writing out the equations for $\mathbf{X}$ in terms of their Witt components. For example, if $\mathbf{X}$ is the elliptic curve over $W_2(\mathbb{F}_2)$ defined by the equation $\mathbf{y}^2 + \mathbf{xy} = \mathbf{x}^3 + 1$, then $G(\mathbf{X})$ is the variety over $\mathbb{F}_2$ defined by the equations $y_0^2 + x_0 y_0 = x_0^3 + 1$ and $x_0 y_0^3 + x_0^2 y_1 + x_1 y_0^2 = x_0^3 + x_0^4 x_1$. Notice that the k-rational points of $G(\mathbf{X})$ are in one-to-one correspondence with the $W_l(k)$-points of $\mathbf{X}$.

Proposition 2. *Let k be a perfect field of characteristic $p > 0$ and E/k an ordinary elliptic curve. If $\mathbf{E}$ is the canonical lift of E to $W_2(k)$, then*

$$\deg x_1 < 3p, \qquad \deg y_1 < 4p$$

Conversely, let $\mathbf{E}$ be any elliptic curve defined over $W_2(k)$ with reduction E. Assume that the projection given by reduction from $G(\mathbf{E})$ to E admits a sec-

tion τ in the category of k-schemes over $E \setminus \{O\}$ (where O is the origin for the group law on E) given by

$$(x_0, y_0) \mapsto (\mathbf{x}, \mathbf{y}) = ((x_0, x_1), (y_0, y_1))$$

where x_1, y_1 are regular away from O and satisfy $\deg x_1 < 3p, \deg y_1 < 4p$. Then τ is regular at O, $\mathbf{E}$ is the the canonical lift of E and τ is the elliptic Teichmüller lift.

5. A simplification in the case of characteristic 2

Let k be a finite field of characteristic 2 and let E be the elliptic curve over k defined by the Weierstrass equation $y^2 + xy = x^3 + a$, for some $a \in k$. E is the reduction modulo 2 of the curve $\mathbf{E}$ over $W_2(k)$ with equation

$$\mathbf{y}^2 + \mathbf{xy} = \mathbf{x}^3 + (a, a^2)$$

and it is now easy to check that the map $\tau : E(k) \to \mathbf{E}(W_2(k))$ given by $(x_0, y_0) \mapsto ((x_0, a), (y_0, (x_0^2 + x_0)y_0 + x_0^3 + ax_0^2 + a))$ satisfies the hypotheses of Proposition 2. Therefore, we know that $\mathbf{E}$ is the canonical lift of E and τ is the elliptic Teichmüller lift on points.

For an integer $r \geq 1$, we may consider the line bundle $\mathcal{L} = \mathcal{O}_{\mathbf{E}}(r\mathbf{O})$ on $\mathbf{E}$. Global sections of this line bundle must have their only pole at $\mathbf{O}$ and that pole must have order at most r. Since $\mathbf{x}$ has a pole of order 2 at $\mathbf{O}$ and $\mathbf{y}$ has a pole of order 3 at $\mathbf{O}$, this means that elements of $\Gamma(\mathbf{E}, \mathcal{L})$ are of the form $A + B\mathbf{y}$, where A and B are polynomials in $\mathbf{x}$ of degrees at most $\lfloor \frac{r}{2} \rfloor$ and $\lfloor \frac{r-3}{2} \rfloor$ respectively. Using the map τ above, we see that for $P \in E(k)$ and $f \in \Gamma(\mathbf{E}, \mathcal{L})$, we have

$$f(\tau(P)) = (f_0(P), f_1(P))$$

as a Witt vector, where f_0 and f_1 are rational functions on E which have poles only at O and those poles are of orders at most r and $2r + 1$ respectively. In other words, $f_0 \in \Gamma(E, \mathcal{O}_E(rO))$ and $f_1 \in \Gamma(E, \mathcal{O}_E((2r + 1)O))$.

Thus, for this particular curve $\mathbf{E}$ and a line bundle of the form $\mathcal{L} = \mathcal{O}_{\mathbf{E}}(r\mathbf{O})$, the sum 2 is the same as the sum

$$\sum_{j=1}^{n} e^{\pi i T(f_0(P), f_1(P))/2} \tag{3}$$

Notice that this sum no longer involves $\mathbf{E}$, but instead is expressed solely in terms of data on the curve E, which is defined over the field k.

Further, we can actually say something much stronger. It is true that *every* ordinary elliptic curve over a field k of characteristic 2 is is isomorphic over $\bar{k}$ to a curve with Weierstrass equation $y^2 + xy = x^3 + a$; see [**10**], Propositions A.1.1 and A.1.2. Since the degree of a line bundle is invariant under base change, this means that for every ordinary elliptic curve in characteristic 2, finding a bound on a sum of the form 2 is equivalent to finding a bound on a sum of the form 3.

6. The bound

We need one more result in order to bound the modulus of our sums 2 and 3. This result can be proven much more generally (see [14]), but we will state it in only the specific case we need here. First, we set up some notation.

Let E be an elliptic curve over the field $\mathbb{F}_{2^m}$, and let K denote the field of rational functions on E. Let $f_0, f_1 \in K$, and assume $f_i \in \Gamma(E, \mathcal{O}_E(r_i O))$ where O is the origin of E and $r_i \geq 1$. Set $\mathbf{f} = (f_0, f_1) \in W_2(K)$. Let $\mathcal{P} = E(k) \setminus \{O\}$, and for $P \in \mathcal{P}$, define $\mathbf{f}(P) = (f_0(P), f_1(P)) \in W_2(\mathbb{F}_{2^m})$. Let F be the additive endomorphism defined in § 2, and let $T : W_2(\mathbb{F}_{2^m}) \to W_2(\mathbb{F}_2) \simeq \mathbb{Z}/4\mathbb{Z}$ be the trace map which was also defined in that section.

Theorem 3. *With notation as defined above, assume that* $\mathbf{f}$ *is not of the form* $\mathbf{f} = F(\mathbf{g}) - \mathbf{g} + \mathbf{c}$ *for any* $\mathbf{g} \in W_2(K)$ *and* $\mathbf{c} \in W_2(\mathbb{F}_{2^m})$. *Then*

$$\left| \sum_{P \in \mathcal{P}} e^{\pi i T(\mathbf{f}(P))/2} \right| \leq (1 + \max\{2r_0, r_1\}) 2^{\frac{m}{2}}$$

A proof of a more general version of this theorem can be found in [14], and most of the steps can actually be found in either [8] or [9]. The basic idea is to consider the degree of the Artin L-function of the Artin-Schreier-Witt cover of $\mathbf{E}$ defined by $F(\mathbf{t}) - \mathbf{t} = \mathbf{f}$. Notice that if $\mathbf{f}$ is of the excluded form, then $T(\mathbf{f}(P))$ would be a constant vector.

By combining this with the argument in § 5, we get the following result.

Theorem 4. *Let* $\mathbf{E}$ *be an elliptic curve over* $W_2(\mathbb{F}_{2^m})$ *which is the canonical lift of an elliptic curve* E *over* $\mathbb{F}_{2^m}$, *and let* $\mathcal{Z} = \{\tau(P) : P \in E(\mathbb{F}_{2^m}) \setminus \{O\}$, *where* $\tau : E(\mathbb{F}_{2^m}) \to \mathbf{E}(W_2(\mathbb{F}_{2^m}))$ *is the elliptic Teichmüller lift. Let* $f \in \Gamma(\mathbf{E}, \mathcal{O}_{\mathbf{E}}(r\mathbf{O}))$ *for some integer* $r \geq 1$. *Then*

$$\left| \sum_{Z \in \mathcal{Z}} e^{\pi i T(f(Z))/2} \right| \leq (2r + 2) 2^{\frac{m}{2}}$$

unless $f \circ \tau \in W_2(\mathbb{F}_{2^m})$ *is of the form* $F(\mathbf{g}) - \mathbf{g} + \mathbf{c}$ *for some Witt vector* $\mathbf{g}$ *of rational functions on* E *and some* $\mathbf{c} \in W_2(\mathbb{F}_{2^m})$.

We can now translate this back into a statement about codes.

Theorem 5. *Let* $\mathbf{E}$ *and* $\mathcal{Z}$ *be as above, and let* $\mathcal{L} = \mathcal{O}_{\mathbf{E}}(r\mathbf{O})$ *for some* $r \geq 1$. *Set* $\mathbb{C} = \mathbb{C}_{\mathbb{Z}/4\mathbb{Z}}(\mathbf{E}, \mathcal{Z}, \mathcal{L})$. *Let* $n = \#\mathcal{Z}$ *denote the length of* $\mathbb{C}$. *Then the minimum Lee weight of the code* $T(\mathbb{C})$ *satisfies*

$$w_L(T(\mathbb{C})) \geq n - (2r + 2) 2^{\frac{m-3}{2}}$$

Proof. Let f be a global section of $\mathcal{L}$. If $f \circ \tau$ is not of the form excluded by Theorem 4, then the Lee weight of the trace of the codeword corresponding to f satisfies the desired inequality by that theorem. Otherwise, the trace of the codeword corresponding to f is constant, so its Lee weight is 0, n, or $2n$. ∎

7. An example

Let $\mathbf{E}$ be the curve over $W_2(\bar{\mathbb{F}}_2)$ defined by $\mathbf{y}^2 + \mathbf{xy} = \mathbf{x}^3 - \mathbf{x}^2 - 2\mathbf{x} - 1$, so that its reduction modulo 2 is the curve E over $\bar{\mathbb{F}}_2$ defined by the equation $y^2 + xy = x^3 + x^2 + 1$. (Although E is isomorphic over $\bar{\mathbb{F}}_2$ to the curve with equation $y^2 + xy = x^3 + 1$, we work with E rather than this latter curve because E has more $\mathbb{F}_8$-rational points.) It is easy to check that if (x_0, y_0) is a point on E, then $((x_0, 1), (y_0, x_0^2(1 + y_0)))$ is a point on $\mathbf{E}$. Further, the map $\tau : E(\bar{\mathbb{F}}_2) \to \mathbf{E}(W_2(\bar{\mathbb{F}}_2))$ given by $(x_0, y_0) \mapsto ((x_0, 1), (y_0, x_0^2(1 + y_0)))$ satisfies the hypotheses of Proposition 2, so we conclude that $\mathbf{E}$ is the canonical lift of E. We construct a code over $W_2(\mathbb{F}_8)$ using $\mathbf{E}$. Applying the trace map then gives us a code over $\mathbb{Z}/4\mathbb{Z}$, and applying the Gray map gives us a binary code.

Write $\mathbb{F}_8 = \mathbb{F}_2[t]/(t^3 + t + 1)$. In addition to the origin O, there are 13 finite $\mathbb{F}_8$-rational points on E. They are the elements of the set:

$$\begin{aligned}\mathcal{P} = \{&(0,1), (t^2, 1+t), (t^2, 1+t+t^2), (t, 1+t^2), (t, 1+t+t^2),\\ &(t+t^2, 1+t^2), (t+t^2, 1+t), (1+t^2, 0), (1+t^2, 1+t^2),\\ &(1+t, 0), (1+t, 1+t), (1+t+t^2, 0), (1+t+t^2, 1+t+t^2)\}\end{aligned}$$

Applying the map τ described above, one gets the origin $\mathbf{O}$ of $\mathbf{E}$ and the following thirteen points of $\mathbf{E}$ defined over $W_2(\mathbb{F}_8)$:

$$\begin{aligned}\mathcal{Z} = \{&((0,1),(1,0)), ((t^2,1),(1+t,1+t+t^2)), ((t^2,1),(1+t+t^2,t)),\\ &((t,1),(1+t^2,t+t^2)), ((t,1),(1+t+t^2,1+t^2)),\\ &((t+t^2,1),(1+t^2,1+t)), ((t+t^2,1),(1+t,t^2)),\\ &((1+t^2,1),(0,1+t+t^2)), ((1+t^2,1),(1+t^2,1)),\\ &((1+t,1),(0,1+t^2)), ((1+t,1),(1+t,1)),\\ &((1+t+t^2,1),(0,1+t)), ((1+t+t^2,1),(1+t+t^2,1))\}\end{aligned}$$

We wish to consider the code $\mathbb{C} = \mathbb{C}_{W_2(\mathbb{F}_8)}(\mathbf{E}, \mathcal{Z}, \mathcal{L})$, where $\mathcal{L} = \mathcal{O}_{\mathbf{E}}(3\mathbf{O})$. The degree of $\mathcal{L}$ is 3, so by the Riemann-Roch theorem [**15**], the rank of the $W_2(\mathbb{F}_8)$-module $\Gamma(\mathbf{E}, \mathcal{L})$ is 3. It is easy to check that $\{1, \mathbf{x}, \mathbf{y}\}$ is a basis for this module, so a (3×13) generator matrix for $\mathbb{C}$ is constructed simply by evaluating these three functions at each of the thirteen points in $\mathcal{Z}$.

In fact, we are interested in the trace code $T(\mathbb{C})$ of $\mathbb{C}$, which will be a code over $W_2(\mathbb{F}_2) = \mathbb{Z}/4\mathbb{Z}$. By Theorem VIII.1.6 of [**12**], we know that the rank of $T(\mathbb{C})$ (as a $\mathbb{Z}/4\mathbb{Z}$-module) is at most 7; we will show that it is exactly 7.

Since $\{1, t, t^2\}$ is a basis for $W_2(\mathbb{F}_8)$ as a $W_2(\mathbb{F}_2)$-module, we know that

$$\{1, t, t^2, \mathbf{x}, t\mathbf{x}, t^2\mathbf{x}, \mathbf{y}, t\mathbf{y}, t^2\mathbf{y}\}$$

is a basis for $\Gamma(\mathbf{E}, \mathcal{L})$ as a $W_2(\mathbb{F}_2)$-module. Some subset of this basis, when evaluated at the points in $\mathcal{Z}$, will give us a set of codewords whose traces form a basis for the trace code $T(\mathbb{C})$. Since $T(1)$, $T(t)$, and $T(t^2)$ are all proportional, we can throw away t and t^2, and look at the 7×13 matrix whose rows are

$(T(f(Z_1)),\ldots,T(f(Z_{13})))$, where $Z_1,\ldots,Z_{13}$ are the points in $\mathcal{Z}$ and f runs over the set $\{1,\mathbf{x},t\mathbf{x},t^2\mathbf{x},\mathbf{y},t\mathbf{y},t^2\mathbf{y}\}$. If this matrix has full rank (7), then it is the generator matrix for $T(\mathbb{C})$.

The matrix described above works out to be (after replacing elements of $W_2(\mathbb{F}_2)$ with their usual representatives in the ring $\mathbb{Z}/4\mathbb{Z}$, as given in 1):

$$\begin{pmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 & 1 & 1 \\ 0 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 3 & 3 \\ 3 & 3 & 1 & 1 & 3 & 3 & 1 & 2 & 3 & 2 & 3 & 2 & 3 \\ 2 & 0 & 3 & 1 & 1 & 1 & 2 & 2 & 3 & 0 & 2 & 2 & 1 \\ 2 & 1 & 1 & 2 & 1 & 0 & 3 & 0 & 2 & 2 & 1 & 2 & 3 \end{pmatrix}$$

Since the submatrix consisting of the last 7 columns of this matrix has determinant 3 (mod 4), this is indeed a generator matrix for the trace code.

By Theorem 5, the minimum Lee weight of the code with the above generator matrix is at least 5. One can check that in fact it is exactly 5. By appending a check digit which is the sum of the first three coordinates, we get a code over $\mathbb{Z}/4\mathbb{Z}$ of length 14, rank 7, and minimum Lee weight 6. The generator matrix is:

$$\begin{pmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 3 & 3 & 2 \\ 0 & 1 & 1 & 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 & 1 & 1 & 2 \\ 0 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 3 & 3 & 0 \\ 3 & 3 & 1 & 1 & 3 & 3 & 1 & 2 & 3 & 2 & 3 & 2 & 3 & 3 \\ 2 & 0 & 3 & 1 & 1 & 1 & 2 & 2 & 3 & 0 & 2 & 2 & 1 & 1 \\ 2 & 1 & 1 & 2 & 1 & 0 & 3 & 0 & 2 & 2 & 1 & 2 & 3 & 0 \end{pmatrix}$$

Applying the Gray map gives a binary code of length 28 with $2^{14} = 16384$ codewords and minimum Hamming distance 6. The best code with this length and number of codewords has minimum Hamming distance 8.

It is interesting to note that the first four rows and the odd-numbered columns of the generator matrix above define the $[7,4]$ Hamming code over $\mathbb{Z}/4\mathbb{Z}$ (see [**2**]). Since only the functions 1 and $\mathbf{x}$ are used here, this gives a construction of this code as a trace code of an algebraic geometric code over $\mathbb{P}^1$. For a more direct construction of the $[7,4]$ Hamming code over $\mathbb{Z}/4\mathbb{Z}$ as an algebraic geometric code, see [**17**].

References

[1] A.R. CALDERBANK AND G.M. MCGUIRE, Construction of a $(64, 2^{37}, 12)$ code via Galois rings, *Designs, Codes, and Cryptography*, vol. 10, pp. 157–165, 1997.

[2] A.R. HAMMONS JR., P.V. KUMAR, A.R. CALDERBANK, N.J.A. SLOANE, AND P. SOLÉ, The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes, *IEEE Trans. Inform. Theory*, vol. 40, pp. 301–319, 1994.

[3] N.M. KATZ, Serre-Tate local moduli, in *Algebraic Surfaces*, (Orsay, 1976–1978), *Lecture Notes Math.*, vol. 868, pp. 138–202, Berlin: Springer-Verlag 1981.
[4] P.V. KUMAR, T. HELLESETH, AND A.R. CALDERBANK, An upper bound for some exponential sums over Galois rings and applications, *IEEE Trans. Inform. Theory*, vol. 41, pp. 456–468, 1995.
[5] S. LANG, *Algebra*, Reading, MA: Addison-Wesley 1974.
[6] W-C.W. LI, Character sums over p-adic fields, unpublished manuscript, 1997.
[7] J. LUBIN, J-P. SERRE, AND J. TATE, Elliptic curves and formal groups, in *Proc. Woods Hole Summer Institute Alg. Geometry*, Woods Hole, MA, 1964.
[8] H.L. SCHMID, Zur arithmetik der zyklischen p-Körper, *Crelles J.* vol. 176, pp. 161–167, 1936.
[9] ———, Kongruenzzetafunktionen in zyklischen Körpern, *Abh. Preuss. Akad. Wiss. Math.-Nat. Kl.*, vol. 1941, 30pp., 1942.
[10] J. SILVERMAN, *The Arithmetic of Elliptic Curves*, New York: Springer-Verlag 1986.
[11] P. SOLÉ, A quaternary cyclic code and a family of quadriphase sequences with low correlation, *Lect. Notes Comp. Science*, vol. 388, pp.193–201, New York: Springer 1989,
[12] H. STICHTENOTH, *Algebraic Function Fields and Codes*, Berlin: Springer-Verlag 1993.
[13] M.A. TSFASMAN and S.G. VLĂDUTS, *Algebraic-Geometry Codes*, Dodrecht: Kluwer Academic 1991.
[14] J.-F. VOLOCH AND J.L. WALKER, Euclidean weights of codes from elliptic curves over rings, submitted for publication.
[15] J.L. WALKER, Algebraic geometric codes over rings, *J. Pure and Appl. Alg.*, to appear.
[16] ———, *Algebraic geometric codes over rings*, Ph.D. thesis, University of Illinois, 1996.
[17] ———, The Nordstrom-Robinson code is algebraic geometric, *IEEE Trans. Inform. Theory*, vol. 43, pp. 1588–1593, 1997.

Curves, Codes and Cryptography

Ian F. Blake

HEWLETT-PACKARD LABORATORIES
1501 PAGE MILL ROAD
PALO ALTO, CA 94304, USA
ifblake@shannon.hpl.hp.com

1. Introduction

The use of curves in algebraic geometries to construct codes [**13, 14, 15**], and asymptotically good codes [**30, 11, 12, 39**], has been a dramatic development in coding theory of the past two decades. These constructions represent a natural, although not obvious, extension of the notions of Reed-Solomon (RS), Bose-Chaudhuri-Hocquenghem (BCH) and Goppa codes. Moving coding theory into such a rich and elegant setting has proven to be productive and enlightening, with many new and interesting directions being pursued.

The use of notions of algebraic geometry in cryptography, and in public key systems in particular, has been more recent but no less dramatic. In this context, the main role of these concepts has been with the construction and presentation of certain abelian groups associated with elliptic and hyperelliptic curves.

While the use to which algebraic geometries have been put in the two subjects is quite different, and indeed the uses in cryptography are quite specific and limited, it is nonetheless of interest to consider the two subjects side-by-side. Our interest in this note is to consider the specific use of elliptic and hyperelliptic curves in the two subjects.

The next section introduces the elements of algebraic geometry needed for the sequel. This includes an introduction to the Hasse-Weil theorem, the Riemann-Roch theorem and the notion of divisors as well as the elements of coding theory and cryptography that will be used subsequently. The treatment of these deep and beautiful results will necessarily be abbreviated with many important results only briefly mentioned. No proofs are given for any of the developments.

In § 3 we consider codes that are derivable from elliptic curves, and their properties, as well as features of elliptic curves that have made them so important for public key cryptosystems. In § 4 we pursue a similar agenda for hyperelliptic curves. While the results for this case are perhaps less prominent than those for the elliptic case, they represent a useful extension that may yet prove to be as important. Some comments in § 5 conclude the presentation.

2. Preliminaries

Let $\mathbb{F}_q$ be the finite field with q elements and $\overline{\mathbb{F}}_q$ its algebraic closure. Let $f(x,y) \in \mathbb{F}_q[x,y]$ be an irreducible bivariate polynomial over $\mathbb{F}_q$ and

$$\mathcal{C} \;=\; \{(u,v) \in \overline{\mathbb{F}}_q^2 : f(u,v) = 0\}$$

That is, $\mathcal{C}$ is the set of solutions, or points, of $f(x,y) = 0$ over $\overline{\mathbb{F}}_q$. In addition, there is the point at infinity, denoted by $\mathcal{O}$, not visible as a solution when using affine coordinates, as in this chapter. The subset of points with coordinates in $\mathbb{F}_q$ are referred to as the rational points of the curve. If there are no points of the curve where both formal derivatives vanish, the curve is referred to as nonsingular, and our interest is in irreducible nonsingular curves. Associated with a curve is an invariant g, the genus of the curve, which is a positive integer. For any given curve, algorithms exist for its determination, although in many cases its computation can prove difficult. An excellent reference on curves in algebraic geometries is the book [**10**]. Treatments of the application of algebraic geometry to coding theory include [**37, 28**] and [**41**].

The genus of a curve is related to the dimension of certain subspaces associated with quantities called divisors. A *divisor* of the curve $\mathcal{C}$ is a finite formal sum

$$D \;=\; \sum_{P_i \in \mathcal{C}} m_i P_i, \qquad\qquad m_i \in \mathbb{Z}$$

where only a finite number of the integers m_i are nonzero. The degree of the divisor is $\sum_i m_i$. Denote by $\boldsymbol{D}$ the set of all divisors of $\mathcal{C}$, which is an abelian group. Denote by $\boldsymbol{D}^0$ the set of divisors of degree 0, which is a subgroup of $\boldsymbol{D}$. Write $D \geq 0$ if $m_i \geq 0$ for all i. The greatest common divisor (gcd) of divisors $D_i = \sum_i m_i P_i$ and $D' = \sum_i m'_i P_i$ is defined to be

$$\gcd(D, D') \;=\; \sum_i \min(m_i, m'_i) P_i \;-\; k\mathcal{O}$$

where $k = \sum_i \min(m_i, m'_i)$. Thus the gcd is a divisor of degree 0. To a bivariate polynomial $g(u,v) \in \overline{\mathbb{F}}_q[u,v]$ can be associated the divisor

$$(g) \;=\; \sum_i m_i P_i - k\mathcal{O} \;\in\; \boldsymbol{D}^0$$

where k is chosen so the divisor has degree 0, and the integer m_i represents the degree of the zero ($m_i > 0$) or pole ($m_i < 0$) the polynomial has at the point P_i. Associated with the rational function $g(u,v)/h(u,v)$ is the divisor $(g) - (h)$. A divisor of this form is referred to as a principal divisor. Clearly all such principal divisors have degree 0 and they form a subgroup $\boldsymbol{P}$ of $\boldsymbol{D}^0$.

The factor group $\boldsymbol{D}^0/\boldsymbol{P}$ is referred to as the jacobian of the curve and will play a role in the formulation of cryptosystems using hyperelliptic curves. Notice that notions of divisors discussed above hold also for curves and their corresponding divisors defined with polynomials of a single variable.

The determination of the number of rational points on a curve of genus g is often a challenging problem. It will be shown how the zeta function of the curve [**37**], defined as

$$\mathcal{Z}(t) = \sum_{n \geq 0} A_n t^n$$

contains information on the numbers of rational points on the curve over $\mathbb{F}_q$ and its extensions. Here A_n is the number of positive divisors of the field of degree n. It can be shown that this function is of the form

$$\mathcal{Z}(t) = \frac{L(t)}{(1-t)(1-qt)}$$

where $L(t)$ is referred to as the L polynomial of the curve and

$$\begin{aligned} L(t) &= \sum_{i=0}^{2g} a_i t^i \in \mathbb{Z}[t], \qquad a_0 = 1, a_{2g} = q^g, a_{2g-i} = q^{g-i} a_i \\ &= \prod_{i=0}^{2g} (1 - \alpha_i t) \end{aligned}$$

It can be shown that $|\alpha_i| = q^{1/2}$, for $i = 1, 2, \cdots, 2g$, an important result, often referred to as the Riemann hypothesis for finite fields. As a consequence of the result, it follows that

$$|N_1 - (q+1)| \leq 2gq^{1/2}$$

a result referred to as the Hasse-Weil theorem, where N_1 is the number of rational points of the curve. Serre [**34, 35**] has sharpened this upper bound to

$$|N_1 - (q+1)| \leq g[2q^{1/2}]$$

where $[x]$ is the integer part of x. As noted, the zeta function contains information on the number of points on the curve over extensions of the base field, $\mathbb{F}_q$, over which the equation is defined. If N_r denotes the number of points of the curve with coordinates in $\mathbb{F}_{q^r}$ then it can be shown that

$$\mathcal{Z}(t) = \exp\left(\sum_{r=1}^{\infty} N_r \frac{t^r}{r}\right)$$

A consequence of the above discussion is that

$$N_r = q^r + 1 - \sum_{i=1}^{2g} \alpha_i^r.$$

Thus the number of solutions (of the curve defined over $\mathbb{F}_q$) over $\mathbb{F}_{q^r}$ is entirely determined by N_1, N_2, $\cdots$, N_g, the number of points on the curve over the fields $\mathbb{F}_{q^i}$ for $i = 1, 2, \cdots, g$. For elliptic curves which have genus 1, the number of points over extension fields is uniquely defined by the number of points over the base field. This result will have useful consequences for the elliptic curve cryptosystem described later.

For a given divisor G on a curve C, define a subset of rational functions on the curve by:

$$L(G) = \{f \in \mathbb{F}_q(x,y) : (f) + G \geq 0\} \cup \{O\}$$

Equivalently, $\mathbb{F}_q(x,y)$ could be replaced with the set of rational functions modulo the defining bivariate polynomial of the curve, denoted $\mathbb{F}_q(C)$. Clearly $L(G)$ is a finite dimensional vector space over $\mathbb{F}_q$. Denote its dimension by $l(G)$. A consequence of the Riemann-Roch theorem [**10**] is then that

$$l(G) \geq \deg(G) + 1 - g$$

where g is the genus of the curve. Indeed for a divisor G with $\deg(G) > 2g-2$, this equation will hold with equality.

For the remainder of this section we consider those concepts from coding and cryptography that will be used in the next two sections. To construct the class of codes that will be of interest, let C be a nonsingular irreducible curve over $\mathbb{F}_q$ of genus g and let $P_1, P_2, \cdots, P_n$ be the rational points on C. Further let $D = P_1 + P_2 + \cdots + P_n$ be a divisor. Let G be a divisor with support disjoint from D and assume that $2g - 2 < \deg(G) < n$.

Define the linear code $\mathbb{C}(D,G)$ over $\mathbb{F}_q$ as the image of the linear map [**37, 28**], as follows:

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q^n \\ f &\longmapsto (f(P_1), f(P_2), \cdots, f(P_n)) \end{aligned}$$

This is often referred to as the evaluation construction of codes from curves in the literature.

The parameters of the code $\mathbb{C}(D,G)$ are established by using the properties discussed above. The kernel of the map is the set $L(G-D)$ and, for the restrictions on the parameters noted,

$$k = \dim \mathbb{C}(D,G) = \dim L(G) - \dim(G-D) = \deg(G) - g + 1.$$

The minimum distance of $\mathbb{C}(D,G)$ is $d \geq d^*$, where $d^* = n - \deg(G)$ is the designed minimum distance. Notice the similarity of this construction to that of RS codes, in that both use the evaluation of a set of functions at points on the curve. For RS codes the curve is the projective line and the set of functions used for the evaluation is the set of polynomials of degree less than some integer, the dimension of the resulting code. Notice also that while RS codes achieve the Singleton bound $d = n - k + 1$ with equality (i.e., they are MDS), the codes $\mathbb{C}(D,G)$ have minimum distance $n - \deg \mathbb{C} = n - k + 1 - g$ and are g away from from the Singleton bound, sometimes referred to as having a defect of g.

The other standard construction for codes, the residue construction, uses the idea of differentials of the curve, and will not be considered.

The notion from public key cryptography that will be required for our presentation is that of a one-way function, on which cryptographic protocols, such as the Diffie-Hellman key exchange, depend. Other protocols, such as digital signatures, authentication and others [**26**], follow readily, depending on the function

in use. The particular one-way function of interest here is that of the discrete logarithm. In a multiplicative cyclic group,

$$G = \langle g \rangle, \qquad \text{with } g^N = 1$$

the discrete logarithm problem is:

$$\text{given } g \text{ and } y = g^x, \text{ find } x$$

and x is referred to as the discrete logarithm of y to the base g. For many groups G this is a difficult problem. For example, in the multiplicative group of the finite fields $\mathbb{F}_{2^m}$ or $\mathbb{F}_p$, p a prime, the order of complexity of the discrete logarithm problem in both time and space, is $L(1/3, c, N)$ where

$$L(v, c, N) = \exp\{c(\log N)^v (\log\log N)^{1-v}\}$$

for some constant c where N is the largest prime divisor of the multiplicative group order [5]. For a prime field, the constant c is approximately 1.6. This complexity function represents subexponential growth in $\log N$. Establishing this complexity for the discrete logarithm problem, uses a technique known as index calculus, which is facilitated by the group actually having two operations, residing as it does in a finite field. To ensure the discrete logarithm problem is as difficult as possible, the cyclic group order $q-1$, should have as large a prime divisor as possible.

It is interesting to note that the complexity of factoring integers of size N with the number field sieve, has precisely the same form i.e., is of the form $L(1/3, c, N)$ where c is approximately 1.9. This is usually taken as the complexity of breaking the RSA public key cryptosystem by integer factorization using the number field sieve factoring technique.

If the cyclic group is chosen so that the discrete logarithm problem is computationally infeasible, by the complexity order shown above, then the following example of a cryptographic protocol, the Diffie-Hellman key exchange, in pervasive usage in security systems, is as follows [26]. A network of users establishes a common cyclic group of fixed and known order N with generator g. User A, wishing to communicate with user B, chooses a random integer a, $1 \leq a < N$, and transmits g^a to B over the network. By assumption on the difficulty of computing discrete logarithms, it is infeasible for an observer on the network, knowing g and g^a, to compute a. Similarly, user B generates a random b and transmits g^b to A. Both parties are in a position to compute g^{ab} while an observer on the network cannot. Bits from the representation of g^{ab} are then used as a key for a symmetric key cryptosystem.

The essence of the application of curves, either elliptic or hyperelliptic, to the discrete logarithm problem, is to replace the groups previously used with additive groups defined from the curves. The presentation of these groups and the group operations, is the focus of the cryptographic aspects of the paper. It will be shown in the next section that using such a group derived from an elliptic curve gives a complexity, by the best algorithm known, that is exponential in $\log N$. The implications of this statement will be commented upon later.

3. Codes and cryptosystems from elliptic curves

It can be shown [**24**] that an elliptic curve over $\mathbb{F}_q$ can always be placed in the form:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \qquad a_i \in \mathbb{F}_q \tag{1}$$

The set of solutions of the equation over $\overline{\mathbb{F}}_q$, together with the point at infinity $\mathcal{O}$, is the curve $E(\overline{\mathbb{F}}_q)$. The set of points with coordinates in $\mathbb{F}_q$, the rational points, is denoted $E(\mathbb{F}_q)$. All such curves have genus 1.

The maximum number of points such a curve of genus 1 can have over $\mathbb{F}_q$ is completely determined [**34, 42, 43**]. From the Hasse-Weil theorem,

$$q + 1 - 2\lfloor\sqrt{q}\rfloor \leq |E(\mathbb{F}_q)| \leq q + 1 + 2\lfloor\sqrt{q}\rfloor$$

For $q = p^a$ if a is an odd integer with $a \geq 3$ and p divides $[2q^{1/2}]$, then q is called exceptional. Then

$$N_q(1) = \begin{cases} q + 1 + [2q^{1/2}], & q \text{ nonexceptional} \\ q + [2q^{1/2}], & q \text{ exceptional} \end{cases}$$

There also exists necessary and sufficient conditions [**42**] for the existence of a curve with exactly N points when the characteristic does not divide $(N - q - 1)$.

If $p = \mathsf{char}(\mathbb{F}_q)$, the curve is called supersingular if $p \mid t$ and non-supersingular otherwise. In terms of the coefficients in equation (1), two fundamental quantities for the curve can be defined [**24**, p. 19], the discriminant Δ, and the j-invariant $j(E)$. The curve is nonsingular iff $\Delta \neq 0$ and, if two curves are isomorphic they have the same j-invariant. Elliptic curves are well classified in terms of their isomorphism classes and their group structure [**36, 24**] but such information is more than is required for our objectives.

From § 2, to determine the number of points on the curve over $\mathbb{F}_{q^k}$, when the curve is defined over $\mathbb{F}_q$, it is sufficient to determine the number of points over $\mathbb{F}_q$, since the curves are of genus 1. Thus if $t = q + 1 - |E(\mathbb{F}_q)|$ then

$$|E(\mathbb{F}_{q^k})| = q^k + 1 - \alpha^k - \beta^k$$

where α and β are solutions of the equation $1 - tx + qx^2 = 0$.

We form codes from the elliptic curves using the evaluation construction of the previous section [**7, 6, 8, 40**]. Let $P_1, P_2, \cdots, P_n$ be the rational points of the elliptic curve over $\mathbb{F}_q$ and $\mathcal{O}$ the point at infinity. Choose the divisor D as $D = P_1 + P_2 + \cdots, P_n$ and let $G = m\mathcal{O}$ for some positive integer m. The code $\mathbb{C}(D, G)$ over $\mathbb{F}_q$ obtained this way has parameters $(n, m, d \geq n - m)$ and

$$|n - (q + 1)| \leq \lfloor 2\sqrt{q}\rfloor$$

Almost all of these codes will have a minimum distance of $d = n - m$, falling short of the Singleton bound by 1. Similarly the curve parameters can be chosen so that the code length is $q + 1 + 2\lfloor\sqrt{q}\rfloor$ or one shorter.

In the case when $G = m\mathcal{O}$, for codes over fields of characteristic 2, it is known [7] that $L(m\mathcal{O})$ has a basis of the polynomials

$$\{1, x, x^2, \cdots, x^{\delta}, y, yx, \cdots, yx^{\widehat{\delta}}\}, \quad \text{where } \delta = \lfloor \tfrac{m}{2} \rfloor \text{ and } \widehat{\delta} = \lfloor \tfrac{m-2}{2} \rfloor$$

to give a code dimension of $\delta + \widehat{\delta} + 2 = \lfloor m/2 \rfloor + \lfloor (m-2)/2 \rfloor + 2$ which is $m+1$ if m is even and m if m is odd. The minimum distance of the code is $n - m$.

The weight enumerator of codes which meet the Singleton bound (the MDS codes) is uniquely specified by the code parameters. Since for (most) codes from elliptic curves, we have $d = n - k$, it is not surprising that some information is available for such codes. The subject has been considered by Katsman and Tsfasman in [17] and in the book [38]. Define the weight enumerators

$$W_C(x) = \sum_{i=0}^{n} A_{n-i} x^i$$

or

$$W_C(x, y) = x^n + y^d \sum_{i=0}^{n-d} A_{d+i} y^i x^{n-d-i}$$

Then, by the MacWilliams identities, we have:

$$W_{C^\perp}(x, y) = q^{-k} W_C(x + (q-1)y, x - y)$$

By applying these relationships to the case of an $(n, k, d = n - k)$ code, it is found [38, p. 302] that:

$$W_C(x) = x^n + \sum_{i=0}^{k-1} \binom{n}{i} (q^{k-i} - 1)(x-1)^i + B_k (x-1)^k$$

It is known that $B'_{n-k} = B_k$, where the prime indicates weights in the dual code. For $n = 2k$, we have $W_C(x) = W_{C^\perp}(x)$ and the codes are formally self dual. Furthermore, if $B_k = 0$ the code is MDS. In general it can be shown that

$$0 < B_k = B'_{n-k} \leq \binom{n}{k} (q-1), \qquad d = n - k$$

and the determination of B_k uniquely specifies the weight enumerator of the code, and hence of its dual. It is possible to characterize B_k as $(q-1)M$ where M is an integer with a combinatorial interpretation in terms of curve parameters [38, p. 303]. More recent investigations of the structure of "codes with small defect" include [3, 9] and [1].

To examine the use of elliptic curves in cryptography, it is noted they have the feature that, for any such curve, it is possible to define an addition of points on the curve. This is a unique feature of elliptic curves and the source of cryptographic interest in them, first proposed independently by Koblitz [19] and Miller [27]. For elliptic curves over the rationals or reals, a geometric picture is available to illustrate the procedure of this point addition. However, the geometric notions, and mathematical equations that reflect them, carry over in much the same manner to finite geometries. The addition results from the observation that the straight line through any two points P, Q, intersects the

curve in a unique third point R. The "addition" of P and Q is then defined as R', where $R + R' = \mathcal{O}$ with $\mathcal{O}$ being the point at infinity. Thus the points $E(\mathbb{F}_q)$ of the elliptic curve over $\mathbb{F}_q$, can be viewed as having the structure of an abelian group under this operation of point addition. This structure serves as the basis of elliptic curve cryptosystems [**24, 18**].

To make the discussion concrete, consider only curves over fields of characteristic 2, namely over $\mathbb{F}_{2^n}$ for some positive integer n. In this case, it is possible to show that any such elliptic curve is isomorphic to a curve of the form either

$$y^2 + xy = x^3 + a_2x^2 + a_6$$

or one of the form:

$$y^2 + a_3y = x^3 + a_4x + a_6$$

depending on whether the j-invariant of the curve is non-zero or zero, respectively. We consider only nonsingular curves. There are precisely $2(q-1)$ non-isomorphic such curves, corresponding to a_2 having either trace 0 or 1 and a_6 being nonzero [**24**]. Notice that for such curves if $P = (x, y)$ is on the curve, then so is $P' = (x, x+y)$. Indeed $P' + P = \mathcal{O}$. For $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ let the sum of the points be $P + Q = (x_3, y_3)$ where:

$$x_3 = \begin{cases} \left(\frac{y_1+y_2}{x_1+x_2}\right)^2 + \frac{y_1+y_2}{x_1+x_2} + x_1 + x_2 + a_2 & P \neq Q, P \neq -Q \\ x_1^2 + \frac{a_6}{x_1^2} & P = Q \end{cases}$$

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

These addition formulas correspond to the notion of drawing a line through the points P and Q and observing that, for elliptic curves, a line through two points always intersects the curve in a unique third point. The sum of this intersection point with the point at infinity is the sum of the two points, $P+Q$. This addition of points is the group operation of the curve and clearly the group is commutative and additive (as opposed to the multiplicative groups considered to this point). When $Q = P$ the line through the points is the tangent line to the point. The order of a point P is the least positive integer N such that $N \cdot P = \mathcal{O}$, the point at infinity. Over fields of characteristic two there is a unique point of order two, the point $(0, \sqrt{a_6})$, since the operation of taking square roots in a field of characteristic 2 is one-to-one.

The discrete logarithm problem in the additive group of points on the curve is then: given a point P of large prime order N and a point multiple

$$c \cdot P = P + \cdots + P \qquad (c \text{ times})$$

determine the integer c. No subexponential algorithm is known to attack this problem. The most efficient algorithm known is the so called square root attack, or baby-step giant step [**26**], described as follows.

For a given point P of order N (that is, $N \cdot P = \mathcal{O}$ is the identity of the group, the point at infinity) and $Q = k \cdot P$ it is desired to find the integer k. A table of pairs is first constructed, $(j, j \cdot P)$ for $0 \leq j < m$ where $m = \lceil \sqrt{N} \rceil$ where the entries are sorted or hashed on the second entry, which are typically pairs of binary n-tuples when the curve is over a finite field of order 2^n. The table takes $O(m)$ storage and $O(m \log N)$ comparisons to sort and $O(m)$ group multiples to construct. To compute the logarithm of Q, compute $m \cdot P$ and set $\gamma = Q = k \cdot P$. For i from 0 to $m - 1$, if $\gamma = j \cdot P$ is the second component of entry j in the table then $k = im + j$. Otherwise set

$$\gamma \ = \ \gamma - m \cdot P$$

and repeat. Each step of the algorithm takes $O(m)$ group multiplications and $O(m)$ table look-ups. The complexity can also be expressed as $\sim \exp(.5 \log(N))$ which is fully exponential in $\log(N)$, as compared to the subexponential complexity of the discrete logarithm problem in finite fields.

The Diffie-Hellman key exchange algorithm in the elliptic curve setting is as follows: the public information available on the network is a point P on a given elliptic curve, the point of order N; user A generates a random integer a with $1 < a < N$ and transmits $a \cdot P$. User B does similarly and both parties derive $a \cdot b \cdot P$, from which a common key is established. Again, the security of the system resides in the difficulty of computing discrete logarithms for elliptic curves.

It is possible to compare the security of the RSA algorithm with that of the elliptic curve algorithm, assuming the algorithms discussed are the best available. It turns out that, for the same level of security, an RSA key of length 1024 bits is equivalent to an elliptic curve key size of approximately 160 bits, using the complexity measures described above. This leads to potentially greater efficiencies of implementation for the elliptic curve system.

A problem for the elliptic curve system is the generation of a curve and a point of "large" prime order, where large in this context is an integer on the order of the field size. One method to achieve this is to define a curve over a relatively small finite field and, choosing a curve at random, determine the number of solutions over this base field, by exhaustive search if necessary. Then the number of points over any extension field can be computed by the method outlined using the zeta function. This random method works well in practice although more complicated deterministic methods are known [**22**]. Considerable effort has also been made in the direct point counting methods [**32, 33**].

4. Codes and cryptosystems from hyperelliptic curves

As a generalization of elliptic curves, consider the hyperelliptic curves, defined by an equation of the form:

$$y^2 + h(x)y \ = \ k(x), \qquad h(x), k(x) \in \mathbb{F}_q[x]$$

where $h(x)$ is a polynomial of degree at most g and $k(x)$ is monic of degree exactly $2g + 1$. We require the curve to be smooth i.e., have no singular points

$(x, y) \in \overline{\mathbb{F}}_q^2$ where both of the partial derivatives $2y+h(x) = 0$ and $h'y-k'(x) = 0$ vanish. In such a case, the genus of the curve is g. Notice that for elliptic curves, $h(x)$ is at most a linear polynomial and $k(x)$ a monic cubic, and the curve is of genus 1. Our main references for this section are [**21**] and [**25**].

If $\mathsf{char}(K) \neq 2$, then the change of variables

$$x \mapsto x, \qquad y \mapsto y - (h(x)/2)$$

transforms the equation to

$$y^2 = f(x), \qquad \deg f(x) = 2g + 1$$

In this case, if $P = (x, y)$ is a point on the curve, then $(x, -y - h(x))$ is also a point on the curve, the sum of P and the point at infinity. If $\mathsf{char}(K) = 2$ and (x, y) is on the curve, then so also is $(x, y + h(x))$. The number of rational points on the curve N is bounded by the result of the Hasse-Weil theorem:

$$|N - (q + 1)| \leq g\lfloor 2\sqrt{q} \rfloor$$

We use the standard evaluation construction of codes from such curves. The following example [**40, 20**] is instructive. Consider the curve

$$y^2 + y = x^5 + 1$$

of genus 2 over $\mathbb{F}_2$. It can be shown using the techniques established previously that the number of points on this curve over $\mathbb{F}_{2^k}$ is $2^k + 1$ unless $k = 4m$ in which case it is

$$2^{4m} + 1 + (-1)^{m+1} 2^{2m+2}$$

Continuing this example, over $\mathbb{F}_{2^4}$ the curve has 33 points. Let D be the sum of these points and $G = m \cdot \mathcal{O}$. For the code $\mathbb{C}(D, G)$ one obtains a sequence of codes $(32, k = 33 - m, d_m)$ over $\mathbb{F}_{16}$ with $d_m = m - 2$ for $3 \leq m \leq 31$, $m \neq 3, 5$, $d_3 = 2$, $d_5 = 4$ meeting the bound for such codes, that is, with defect $g = 2$ from the Singleton bound.

In general one can use either of the two standard constructions of codes from hyperelliptic curves, although it is difficult to improve on sharpening estimates of the code parameters from those given by the Riemann-Roch theorem. An approach to using hyperelliptic curves that uses only elementary properties of polynomials, rather than the theorems of algebraic geometry, is attempted in [**44**]. They obtain codes for low genus, that meet the general distance bound. The decoding of codes from hyperelliptic curves was considered in [**2**].

The use of hyperelliptic curves in cryptosystems is less advanced than the use of elliptic curves, but nonetheless has attracted attention [**21, 31, 25, 4**]. We outline the approach here, following the approach of [**21**] and [**25**]. As noted already, the idea is to determine an abelian group with an efficient presentation, and with a cyclic subgroup of large prime order. Of course, there is no point addition for genus $g > 1$ curves, as for elliptic curves. The group of choice for hyperelliptic curves is the jacobian already described, the factor group of divisors of degree zero $\boldsymbol{D}^0$ by the subgroup of principal divisors $\boldsymbol{P}$. To compute in such a group $\boldsymbol{J} = \boldsymbol{D}^0/\boldsymbol{P}$, it is convenient to determine a set of divisors that

represent equivalence classes of the group where $D_1 \sim D_2$ iff $D_1 - D_2 \in \boldsymbol{P}$. To do this, define a divisor $D = \sum_i m_i P_i - kP \in \boldsymbol{D}^0$ to be a reduced divisor [**21, 25**] if:

a. all of the m_i are non-negative and $m_i \leq 1$ if P_i is equal to its opposite point;

b. if P_i does not equal to its opposite point, then both do not occur in the divisor D;

c. $\sum_i m_i \leq g$.

Given a divisor, it is not difficult to determine its unique reduced divisor, or equivalence class representative. Any reduced divisor $D = \sum_i m_i P_i - k\mathcal{O}$ can be uniquely represented as the gcd of the divisor associated with the polynomial $a(x) = \prod_i (x - x_i)^{m_i}$ and the divisor associated with the polynomial $b(x) - y$ where $P_i = (x_i, y_i)$ and $b(x)$ is the unique polynomial of degree less than that of $a(x)$, such that $b(x_i) = y_i$ for all i and $b^2(x) + h(x)b(x) - k(x)$ is divisible by $a(x)$. For D represented in this manner, denote $D = \mathrm{div}(a, b)$.

Now given any two reduced divisors $D_1 = \mathrm{div}(a_1, b_1)$ and $D_2 = \mathrm{div}(a_2, b_2)$ there is an algorithm to compute the reduced divisor D_3 that is equivalent to $D_1 + D_2$. Although computing in the jacobian of a general curve is well defined, the algorithms for computing reduced divisors and the addition of reduced divisors to yield a sum reduced divisor, is particularly straight forward to describe for hyperelliptic curves, which is the reason the jacobian has been proposed for cryptographic use. The jacobians of more general curves are less attractive computationally and little seems to have been done in that direction.

5. Comments

The uses of both elliptic and hyperelliptic curves in algebraic geometries to coding and cryptography has been considered. The case of elliptic curves is perhaps the most interesting at this point. In terms of coding, starting from extended Reed-Solomon codes of length q and dimension k over $\mathbb{F}_q$ with minimum distance $q + 1 - k$, the codes from elliptic curves demonstrate that the length can be extended to at most $q + 1 + 2\sqrt{q}$ with a decrease of one in the minimum distance. The trade-off between alphabet size, code length, dimension, and minimum distance is a long standing coding problem and whether, for the parameters of the elliptic code curves, this is the best possible seems a very difficult question, as is the related question for Reed-Solomon codes. It does, however, represent an interesting contribution to the problem.

The use of elliptic curves in cryptography is also dramatic. The fact that the best known algorithm to break the elliptic curve discrete logarithm problem is the square root attack, implies that the same level of security can be achieved with a smaller group than, for example, for the multiplicative group of a finite field in which the discrete logarithm is usually considered. This has very direct and important implication for system efficiency. Thus the consideration of

curves in algebraic geometries has led in a very direct way to a crypto system that is very efficient and has already been included in industry standards [16].

The case of hyperelliptic curves represents an extension to the case of elliptic curves. In terms of both coding and cryptography less is known about such systems. For coding, there are many codes known from hyperelliptic curves that achieve the bounds in terms of minimum distance and dimension predicted by the theory. However, less is known in general. For cryptography, the jacobian proposed for use as the one way function seems interesting but it has received much less attention than the elliptic curve group and is of quite a different form than that one. It has not, to date, as far as the author is aware, been considered for standards as has the elliptic case. It has however, been considered for such an application, by several authors.

This consideration of the two classes of curves, elliptic and hyperelliptic, has shown the directions that investigations have taken and the results achieved. It is an interesting exercise in its own right to see how the elegant results of algebraic geometry have, and continue to be, important in both theory and practice.

References

[1] A.M. BARG, S.L. KATSMAN AND M.A. TSFASMAN, Algebraic geometric codes from curves of small genus, *Probl. Inform. Transmission*, vol. 23, pp. 34–38, 1987.

[2] D. LE BRIGAND, Decoding of codes on hyperelliptic curves, in *Lect. Notes Computer Science*, vol. 514, Berlin: Springer-Verlag, pp. 126–134, 1990.

[3] M.A. DE BOER, Almost MDS codes, *Designs, Codes and Cryptography*, vol. 9, pp. 143–155, 1996.

[4] D. CANTOR, Computing in the Jacobian of a hyperelliptic curve, *Math. Computation*, vol. 48, pp. 95–101, 1987.

[5] H. COHEN, *A Course in Computational Algebraic Number Theory*, Berlin: Springer 1993.

[6] Y. DRIENCOURT, Some properties of elliptic codes over a field of characteristic 2, in *Lect. Notes Computer Science*, vol. 229, Berlin: Springer-Verlag 1985.

[7] Y. DRIENCOURT AND J.F. MICHON, Elliptic codes over fields of characteristic 2, *J. Pure Appl. Algebra*, vol. 45, pp. 15–39, 1987.

[8] Y. DRIENCOURT AND J.F. MICHON, Remarques sur les codes géométriques, *Comptes Rendus*, vol. 301, pp. 15–17, 1985.

[9] A. FALDUM AND W. WILLEMS, Codes of small defect, *Designs, Codes, and Cryptography*, vol. 10, pp. 341–350, 1997.

[10] W. FULTON, *Algebraic Curves: An Introduction to Algebraic Geometry*, Reading, MA: Addison-Wesley 1969.

[11] A. GARCIA AND H. STICHTENOTH, A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlăduţ bound, *Inventiones Math.*, vol. 121, pp. 211–222, 1995.

[12] ———, On the asymptotic behavior of some towers of function fields over finite fields, *J. Number Theory*, vol. 61, pp. 248–273, 1996.

[13] V.D. GOPPA, Codes on algebraic curves, *Soviet Math. Dokl.*, vol. 24, pp. 170–172, 1981.

[14] ———, Codes associated with divisors, *Probl. Inform. Transmission*, vol. 13, pp. 22–27, 1977.

[15] ———, Algebraico-geometric codes, *Math. USSR Izvestiya*, vol. 21, pp. 75–91, 1983.

[16] IEEE P1363 Standard for RSA, Diffie-Hellman, and Related Public-Key Cryptography: Elliptic Curve Systems, February 6, 1997.
[17] G.L. KATSMAN AND M.A. TSFASMAN, Spectra of algebraic geometry codes, *Probl. Inform. Transmission*, vol. 23, pp. 262–275, 1987.
[18] N. KOBLITZ, *A Course in Number Theory and Cryptography*, Berlin: Springer 1994.
[19] ———, Elliptic curve cryptosystems, *Math. Computation*, vol. 48, pp. 203–209, 1987.
[20] ———, Constructing elliptic curve cryptosystems in characteristic 2, in *Lect. Notes Computer Science*, vol. 537, pp. 156–167, Berlin: Springer-Verlag 1991.
[21] ———, Elliptic and hyperelliptic cryptosystems, unpublished manuscript, 1997.
[22] G.-J. LAY AND H.G. ZIMMER, Constructing elliptic curves with given group order over large finite fields, in *Algorithmic Number Theory: Part I, Lect. Notes Computer Science*, vol. 877, pp. 250–263, Berlin: Springer-Verlag 1994.
[23] F.J. MACWILLIAMS AND N.J.A. SLOANE, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland, 1977.
[24] A.J. MENEZES, *Elliptic Curve Public Key Cryptosystems*, Dordrecht: Kluwer 1993.
[25] A.J. MENEZES, Y.-H. WU, AND R.J. ZUCCHERATO, An elementary introduction to hyperelliptic curves, unpublished manuscript, 1997.
[26] A.J. MENEZES, P.C. VAN OORSCHOT, AND S.A. VANSTONE, *Handbook of Applied Cryptography*, Boca Raton, FL: CRC Press 1996.
[27] V. MILLER, Uses of elliptic curves in cryptography, in *Advances in Cryptology: Crypto'85, Lect. Notes Computer Science*, vol. 218, pp. 417–426, Berlin: Springer-Verlag 1985.
[28] C. MORENO, *Algebraic Curves over Finite Fields*, Cambridge, UK: Cambridge University Press 1993.
[29] A.M. ODLYZKO, *The future of integer factorization*, in Cryptobytes, RSA Inc., 1993.
[30] R. PELLIKAAN, H. STICHTENOTH, AND F. TORRES, Weierstrass semigroups in an asymptotically good tower of function fields, unpublished manuscript.
[31] J. PILA, Frobenius maps of abelian varieties and finding roots of unity in finite fields, *Math. Computation*, vol. 55, pp. 745–763, 1990.
[32] R. SCHOOF, Elliptic curves over finite fields and the computation of square roots mod p, *Math. Computation*, vol. 44, pp. 483–494, 1988.
[33] ———, Counting points on elliptic curves over finite fields, *J. Théorie des Nombres*, vol. 7, pp. 219–254, 1995.
[34] J.-P. SERRE, Nombres des points des courbes algébriques sur $\mathbb{F}_q$, *Séminaire de Théorie des Nombres de Bordeaux*, vol. 22, 1982/83.
[35] ———, Sur le nombre des points rationnels d'une courbe algébrique sur un corps fini, *Comptes Rendus*, vol. 296, pp. 397–402, 1983.
[36] J. SILVERMAN, *The Arithmetic of Elliptic Curves*, Berlin: Springer-Verlag 1986.
[37] H. STICHTENOTH, *Algebraic Function Fields and Codes*, Berlin: Springer-Verlag 1991.
[38] M.A. TSFASMAN AND S.G. VLĂDUŢ, *Algebraic-Geometric Codes*, Dordrecht: Kluwer Academic 1991.
[39] M.A. TSFASMAN, S.G. VLĂDUTS, AND T. ZINK, Modular curves, Shimura curves, and Goppa codes better than the Varshamov-Gilbert bound, *Math. Nachrichten*, vol. 109, pp. 21–28, 1982.
[40] G. VAN DER GEER, Codes and elliptic curves, in *Effective Methods in Algebraic Geometry*, T. Mora and C. Traverso (Editors), pp. 160–168, Basel: Birkhäuser 1991.
[41] J.H. VAN LINT AND G. VAN DER GEER, *Introduction to Coding Theory and Algebraic Geometry*, Basel: Birkhäuser 1988.
[42] W.C. WATERHOUSE, Abelian varieties over finite fields, *Ann. Sci. E.N.S.*, vol. 2, pp. 521–560, 1969.
[43] A. WEIL, *Variétés Abéliennes et Courbes Algébriques*, Paris: Hermann 1948.
[44] T. YAGHOOBIAN AND I.F. BLAKE, Codes from hyperelliptic curves, in *Proc. 30-th Allerton Conf. Comm., Control, and Computing*, Monticello, IL., October 1992.

Part II

CODES AND SIGNALS

Transforms and Groups

G. David Forney, Jr.

MOTOROLA, INC., MANSFIELD, MA 02048, USA
AND DEPARTMENT OF ELECTRICAL ENGINEERING
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MA 02139, USA
LUSE27@email.mot.com

ABSTRACT. The fundamental properties of Fourier transforms derive from properties of locally compact abelian groups. Fourier transforms are group character transforms, and time-frequency duality is Pontryagin duality. Applications to block codes, MacWilliams identities and fast Fourier transforms are given.

1. Introduction

The Fourier transform appears to be inherently multiplicative. The thesis of this paper is that many of its fundamental properties are actually based on the properties of abelian groups. In particular:

- *The time domain $\mathcal{T}$ and frequency domain $\mathcal{F}$ are dual character groups.*
- *The Fourier transform is a character transform.*
- *The orthogonality relations of group characters underlie the inverse Fourier transform and the unitarity of the Fourier transform operator.*
- *The duality between subgroups and quotients of orthogonal subgroups leads to aliasing/decimation relations and fast Fourier transforms.*
- *The Poisson summation formula yields the MacWilliams identities for block codes and the Poisson-Jacobi identities for lattices.*

Nothing here is new, with the possible exception of the general statement of the picket-fence miracle in § 7. The basic theorems will be found in the early chapters of any book on abstract harmonic analysis, for example in [8] or [11]. For key early work on coding applications, see [5].

I am grateful to Thomas Ericson [6] for getting me started in this area, and to M.D. Trott, H.-A. Loeliger and T. Mittelholzer for further education.

2. Locally compact abelian groups

In this paper the time domain $\mathcal{T}$ and the frequency domain $\mathcal{F}$ will be locally compact abelian (LCA) groups. For example, the reals $\mathbb{R}$ form a locally compact abelian group under addition, with the usual Euclidean topology.

All other groups considered here are LCA groups that may be obtained from $\mathbb{R}$ by taking subgroups, quotients, or finite direct products. In particular:

- The integers $\mathbb{Z}$, an infinite discrete subgroup of $\mathbb{R}$;
- The torus $\mathbb{R}/\mathbb{Z}$ (the "reals mod 1") under addition, a compact group which is topologically isomorphic to the complex circle group $\mathbb{C}$, the group of unit-magnitude complex numbers under multiplication;
- Any finite cyclic group $\mathbb{Z}/m\mathbb{Z} \cong \mathbb{Z}_m$;
- Euclidean n-space $\mathbb{R}^n$, the n-fold direct product of $\mathbb{R}$ with itself;
- Any lattice $\Lambda \subseteq \mathbb{R}^n$, i.e., any discrete subgroup of $\mathbb{R}^n$;
- Any finite direct product $\mathcal{G}$ of finite cyclic groups; i.e., any finite abelian group.

Although most of the results discussed in this paper hold for LCA groups in general, we will often illustrate these results using the specific groups listed above.

3. Pontryagin duality

The *character group* $\mathcal{G}^\wedge$ of an LCA group $\mathcal{G}$ is the group under composition of all continuous homomorphisms $\varphi : \mathcal{G} \to \mathbb{C}$ from $\mathcal{G}$ into the complex circle group $\mathbb{C}$. Note that the homomorphic property implies $\varphi(0) = 1$ and $\varphi(g+h) = \varphi(g)\varphi(h)$.

In this paper we will regard an LCA group $\mathcal{G}$ as a *time domain* $\mathcal{T}$, and its character group $\mathcal{T}^\wedge$ as a *frequency domain* $\mathcal{F}$. The elements of $\mathcal{T}^\wedge$ are continuous homomorphisms $\varphi_f : \mathcal{T} \to \mathbb{C}$, where f ranges through some index set $\mathcal{I}$. We will often identify the character group $\mathcal{T}^\wedge$ with $\mathcal{I}$. We claim that the well-known dualities between time and frequency domains are essentially instances of Pontryagin duality [10], whose fundamental result is the following theorem.

Pontryagin duality theorem. *Let $\mathcal{T}$ be an LCA group. Then:*

a. *Its character group $\mathcal{F} = \mathcal{T}^\wedge$ is LCA, with an appropriate topology.*

b. *The pairing $\mathcal{F} \times \mathcal{T} \to \mathbb{C}$ defined by $\langle f, t \rangle = \varphi_f(t)$ is bihomomorphic. This pairing is often called the inner product.*

c. *The map $\varphi_t : \mathcal{F} \to \mathbb{C}$ defined by $\varphi_t(f) = \langle f, t \rangle$ is a continuous character of $\mathcal{F}$, and all continuous characters of $\mathcal{F}$ are of this form.*

d. *The character group $\mathcal{F}^\wedge = \mathcal{T}^{\wedge\wedge}$ is thus naturally isomorphic to $\mathcal{T}$; that is $\mathcal{T}^{\wedge\wedge} \cong \mathcal{T}$. One may even say that $\mathcal{T}^{\wedge\wedge}$ is $\mathcal{T}$.*

In view of this theorem, the character group $\mathcal{F} = \mathcal{T}^\wedge$ is sometimes called the dual group to $\mathcal{T}$.

Example 1. The character group of the real time axis $\mathcal{T} = \mathbb{R}$ is the group $\mathcal{T}^\wedge$ of continuous homomorphisms $\varphi_f : \mathbb{R} \to \mathbb{C}$ for $f \in \mathbb{R}$, defined by

$$\varphi_f(t) \;=\; \exp(-j2\pi ft)$$

where j is the square root of -1. The group $\mathcal{T}^\wedge$ under composition is topologically isomorphic to $\mathbb{R}$. Thus we may identify $\mathcal{F} = \mathcal{T}^\wedge$ with $\mathbb{R}$; i.e., the frequency domain in this case is also the real axis. The pairing $\langle f, t\rangle$ is

$$\langle f, t\rangle \;=\; \exp(-j2\pi ft)$$

which is evidently bihomomorphic in $\mathcal{T}$ and $\mathcal{F}$. The character group of $\mathcal{F} = \mathbb{R}$ is similarly $\mathcal{F}^\wedge \cong \mathbb{R}$, which may be identified with $\mathbb{R}$ using the same pairing. □

A second duality notion arises from a natural definition of orthogonality. If $\mathcal{T}$ and $\mathcal{F} = \mathcal{T}^\wedge$ are dual character groups, then $t \in \mathcal{T}$ and $f \in \mathcal{F}$ are said to be *orthogonal* if $\langle f, t\rangle = 1$. For a subset $\mathcal{S} \subseteq \mathcal{T}$, the *orthogonal subgroup* $\mathcal{S}^\perp$ is the set of all $f \in \mathcal{F}$ that are orthogonal to all $t \in \mathcal{S}$. Namely:

$$\mathcal{S}^\perp \;=\; \{f \in \mathcal{F} : \langle f, t\rangle = 1 \text{ for all } t \in \mathcal{S}\}$$

The orthogonal subgroup of $\mathcal{T}$ itself is the trivial subgroup $\{0\} \in \mathcal{F}$ consisting of the *zero character* 0, for which $\langle 0, t\rangle = 1$ for all $t \in \mathcal{T}$. It is straightforward to verify that $\mathcal{S}^\perp$ is a closed subgroup of $\mathcal{F}$. Indeed, orthogonal subgroups and closed subgroups are intimately linked by the following duality theorem.

Orthogonal subgroup duality theorem. *Let $\mathcal{T}$ be an LCA group with character group $\mathcal{F} = \mathcal{T}^\wedge$ and let $\mathcal{S}$ be a subset of $\mathcal{T}$. Then:*

a. *The orthogonal subgroup $\mathcal{S}^\perp$ is a closed subgroup of $\mathcal{F}$.*
b. *The orthogonal subgroup $\mathcal{S}^{\perp\perp}$ is the closure $\mathcal{S}^c$ of $\mathcal{S}$ in $\mathcal{T}$.*
c. *The orthogonal subgroup $\mathcal{S}^{\perp\perp\perp}$ is equal to $\mathcal{S}^\perp$.*
d. *$\mathcal{S}$ is a closed subgroup of $\mathcal{T}$ if and only if $\mathcal{S}^{\perp\perp} = \mathcal{S}$.*

Thus two orthogonal closed subgroups $\mathcal{S} = \mathcal{S}^{\perp\perp} \subseteq \mathcal{T}$ and $\mathcal{S}^\perp \subseteq \mathcal{F}$ are dual to each other in the second sense. For instance, when we speak of dual codes or lattices, it is in the orthogonal subgroup sense. For clarity, we therefore avoid the use of the term "dual group" in favor of "character group" or "orthogonal subgroup." The following are instances of orthogonal subgroup duality:

- If $\mathcal{T} = \mathbb{R}^n$ and $\mathcal{S}$ is a *subspace* of $\mathcal{T}$ as a vector space over $\mathbb{R}$, then $\mathcal{S}^\perp$ is the *orthogonal subspace* to $\mathcal{S}$ in $\mathcal{F} = \mathbb{R}^n$.
- If $\mathcal{T} = \mathbb{R}^n$ and $\mathcal{S}$ is a *lattice* in $\mathbb{R}^n$, then $\mathcal{S}^\perp$ is the *dual lattice* in $\mathcal{F} = \mathbb{R}^n$, namely: $\mathcal{S}^\perp = \{\boldsymbol{f} \in \mathbb{R}^n : \boldsymbol{f} \cdot \boldsymbol{t} \equiv 0 \mod \mathbb{Z}, \text{ for all } \boldsymbol{t} \in \mathcal{S}\}$.
- If $\mathcal{T} = (\mathbb{Z}_m)^n$ and $\mathcal{S}$ is a subgroup (a *linear block code* of length n over $\mathbb{Z}_m$), then $\mathcal{S}^\perp$ is the *dual linear block code* in $\mathcal{F} = (\mathbb{Z}_m)^n$.

We now have another important duality theorem, which shows that orthogonal subgroups and quotient groups are dual character groups. The theorem basi-

cally arises from the observation that for $t \in \mathcal{S}$ the pairing $\langle f, t \rangle$ is constant on the cosets $\mathcal{S}^\perp + f$ of $\mathcal{S}^\perp$ in $\mathcal{F} = \mathcal{T}^\wedge$, so that the quotient group $\mathcal{T}^\wedge/\mathcal{S}^\perp$ acts as the character group $\mathcal{S}^\wedge$ of $\mathcal{S}$ with the pairing thus induced.

Subgroup/quotient group duality theorem. *If $\mathcal{S}$ is a closed subgroup of an LCA group $\mathcal{T}$ and $\mathcal{S}^\perp$ is its orthogonal subgroup in $\mathcal{F} = \mathcal{T}^\wedge$, then $\mathcal{S}$ and $\mathcal{T}^\wedge/\mathcal{S}^\perp$ act as dual character groups; so do $\mathcal{T}/\mathcal{S}$ and $\mathcal{S}^\perp$.*

As an important example, $\mathbb{Z}$ and $\mathbb{R}/\mathbb{Z}$ are dual character groups. Thus if the time domain $\mathcal{T}$ is the set of integers $\mathbb{Z}$, then the frequency domain $\mathcal{F}$ may be taken as a "Nyquist interval" $[0, 1)$ representing the cosets of $\mathbb{Z}$ in $\mathbb{R}$. Conversely, if the time domain represents the cosets of $\mathbb{Z}$ in $\mathbb{R}$ — for example, if we are considering periodic functions with period 1, or functions defined on the interval $[0, 1)$ — then the frequency domain is the set of integers $\mathbb{Z}$. In either case, the time and frequency domains may be scaled by factors of τ and $1/\tau$, respectively.

This example shows that dual character groups need not be isomorphic. Furthermore, it illustrates an important general result.

Discreteness/compactness duality theorem. *An LCA group $\mathcal{T}$ is discrete if and only if its character group $\mathcal{T}^\wedge$ is compact.*

The subgroup/quotient group duality theorem may be straightforwardly generalized to a chain of closed subgroups, say $\mathcal{R} \subseteq \mathcal{S} \subseteq \mathcal{T}$, and their corresponding quotient groups. Note that the chain of orthogonal subgroups runs in the reverse order:

$$\{0\} = \mathcal{T}^\perp \subseteq \mathcal{S}^\perp \subseteq \mathcal{R}^\perp$$

Quotient group duality theorem. *If $\mathcal{R} \subseteq \mathcal{S} \subseteq \mathcal{T}$, then $\mathcal{S}/\mathcal{R}$ and $\mathcal{R}^\perp/\mathcal{S}^\perp$ act as dual character groups.*

We call $\mathcal{S}/\mathcal{R}$ and $\mathcal{R}^\perp/\mathcal{S}^\perp$ dual quotient groups, where "dual" is used in the character group sense.

The subgroup/quotient group duality theorem is a special case of this theorem with $\mathcal{R} = \{0\}$ and $\mathcal{R}^\perp = \mathcal{T}^\wedge$. That is, $\mathcal{S} = \mathcal{S}/\{0\}$ and $\mathcal{T}^\wedge/\mathcal{S}^\perp$ are dual quotient groups. For example, consider the chain

$$\{0\} \subseteq m\mathbb{Z} \subseteq \mathbb{Z} \subseteq \mathbb{R}$$

whose quotients are $m\mathbb{Z} \cong \mathbb{Z}$, $\mathbb{Z}/m\mathbb{Z} \cong \mathbb{Z}_m$, and $\mathbb{R}/\mathbb{Z}$. The orthogonal chain is

$$\mathcal{R}^\perp = \{0\} \subseteq \mathbb{Z}^\perp = \mathbb{Z} \subseteq (m\mathbb{Z})^\perp = m^{-1}\mathbb{Z} \subseteq \{0\}^\perp = \mathbb{R}$$

It follows that $\mathbb{R}/\mathbb{Z}$ and $\mathbb{Z}$, $m\mathbb{Z} \cong \mathbb{Z}$ and $\mathbb{R}/m^{-1}\mathbb{Z} \cong \mathbb{R}/\mathbb{Z}$, $\mathbb{Z}/m\mathbb{Z} \cong \mathbb{Z}_m$ and $m^{-1}\mathbb{Z}/\mathbb{Z} \cong \mathbb{Z}_m$ are dual quotient groups.

In particular, this shows that the character group of a finite cyclic group $\mathbb{Z}/m\mathbb{Z} \cong \mathbb{Z}_m$ is isomorphic to $\mathbb{Z}_m$. In this case the pairing $\langle f, t \rangle = \exp(-j2\pi ft)$ lies in the group $\langle \mathcal{F}, \mathcal{T} \rangle = \{\omega^k : 0 \leq k \leq m-1,\ \omega = \exp(-j2\pi/m)\}$ of m-th

complex roots of unity, a subgroup of $\mathbb{C}$ that is isomorphic to $\mathbb{Z}_m$. In fact, by identifying $\mathbb{Z}/m\mathbb{Z}$, $m^{-1}\mathbb{Z}/\mathbb{Z}$, and $\langle \mathcal{F}, \mathcal{T} \rangle$ with $\mathbb{Z}_m$, the pairing $\langle f, t \rangle$ may be computed as the usual product ft in $\mathbb{Z}_m$.

Finally, suppose that $\mathcal{T}$ is a finite direct product of LCA groups. Then we have the following theorem.

Direct product theorem. *If $\mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_m$ is a finite direct product of LCA groups, then:*

a. *$\mathcal{T}$ is LCA.*

b. *the character group of $\mathcal{T}$ is the direct product*
$$\mathcal{F} = \mathcal{T}^\wedge = \mathcal{T}_1^\wedge \times \cdots \times \mathcal{T}_m^\wedge$$

c. *if $t = (t_1, \cdots, t_m)$ and $f = (f_1, \cdots, f_m)$, then the pairing $\langle f, t \rangle$ is the product*
$$\langle f, t \rangle \;=\; \langle f_1, t_1 \rangle \cdots \langle f_m, t_m \rangle$$

It follows from the direct product theorem that if $\mathcal{T} = \mathbb{R}^n$, then $\mathcal{F} = \mathcal{T}^\wedge = \mathbb{R}^n$, and the pairing $\langle f, t \rangle$ is given by

$$\langle f, t \rangle \;=\; \exp\{-2\pi j(f_1 t_1 + \cdots + f_m t_m)\} \;=\; \exp\{-2\pi j(f \cdot t)\}$$

where $f \cdot t$ is the usual "dot product." Furthermore, since any finite abelian group can be expressed as a direct product of finite cyclic groups, and since for a finite cyclic group we have $(\mathbb{Z}_m)^\wedge \cong \mathbb{Z}_m$, it also follows that the character group $\mathcal{F} = \mathcal{T}^\wedge$ of any finite abelian group $\mathcal{T}$ is isomorphic to $\mathcal{T}$.

While in these two cases $\mathcal{T}$ and $\mathcal{F} = \mathcal{T}^\wedge$ are isomorphic, the isomorphisms are not "natural." Moreover, as we have seen with the dual character groups $\mathbb{Z}$ and $\mathbb{R}/\mathbb{Z}$, the groups $\mathcal{T}$ and $\mathcal{F}$ need not even be isomorphic. However, in some sense it is fair to think of $\mathcal{T}$ and $\mathcal{F}$ as having the "same size."

4. Fourier transforms

For any LCA group $\mathcal{T}$, there exists an essentially unique (up to scaling) complex-valued translation-invariant measure μ on $\mathcal{T}$, called *Haar measure*. The translation-invariance property means that if $\mathcal{S} \subseteq \mathcal{T}$ has measure $\mu(\mathcal{S}) = \int_{\mathcal{S}} d\mu(t)$, then $\mu(\mathcal{S} + t) = \mu(\mathcal{S})$ for all $t \in \mathcal{T}$. We can think of Haar measure as a "uniform distribution over $\mathcal{T}$." For example:

- If $\mathcal{T} = \mathbb{R}$, then Haar measure is the usual Lebesgue measure on $\mathbb{R}$, and $d\mu(t)$ is denoted by dt. The measure of $\mathcal{S} \subseteq \mathbb{R}$ is its volume $V(\mathcal{S}) = \int_{\mathcal{S}} dt$.
- If $\mathcal{T}$ is discrete, then each $t \in \mathcal{T}$ has equal Haar measure $\mu(t)$, conventionally scaled so that $\mu(t) = 1$. An integral $\int_{\mathcal{S}} x(t) d\mu(t)$ becomes a sum $\sum_{\mathcal{S}} x(t)$. The measure of $\mathcal{S} \subseteq \mathcal{T}$ becomes its size $\mu(\mathcal{S}) = \int_{\mathcal{S}} d\mu(t) = |\mathcal{S}|$.

- If $\mathcal{T}$ is compact, then $\mu(\mathcal{T}) = \int_{\mathcal{T}} d\mu(t)$ is finite, and Haar measure is conventionally scaled so that $\mu(\mathcal{T}) = 1$. For example, Haar measure on $\mathbb{R}/\mathbb{Z}$ may be represented by the usual Lebesgue measure dt on $[0, 1)$, in which case $\mu(\mathbb{R}/\mathbb{Z}) = \int_{[0,1)} dt = 1$.
- If $\mathcal{T}$ is finite, then it is both discrete and compact, so these two conventions conflict. In this case we take $\mu(t) = 1$ on $\mathcal{T}$ so that $\mu(\mathcal{T}) = |\mathcal{T}|$, and on the character group $\mathcal{F} = \mathcal{T}^\wedge$, whose size is also $|\mathcal{F}| = |\mathcal{T}|$, we take $\mu(f) = 1/|\mathcal{T}|$ so that $\mu(\mathcal{F}) = 1$.

For any continuous complex-valued Borel function $x : \mathcal{T} \to \mathbb{C}$, called a *waveform*, the integral

$$x(\mathcal{T}) = \int_{\mathcal{T}} x(t)\, d\mu(t)$$

is well defined. The L_p-norm of a waveform x is $\|x\|_p = \left[\int_{\mathcal{T}} |x(t)|^p\, d\mu(t)\right]^{1/p}$. We let $L_p(\mathcal{T})$ denote the set of waveforms with finite L_p-norm.

For a waveform x in $L_1(\mathcal{T})$, the *Fourier transform* of x is a function $X \in L_1(\mathcal{F})$ defined for all $f \in \mathcal{F}$ by

$$X(f) = \int_{\mathcal{T}} x(t)\langle f, t\rangle d\mu(t)$$

In this context, the pairing $\langle f, t\rangle$ is sometimes called the *Fourier kernel*. If $\mathcal{T}$ is discrete and $\mu(t) = 1$ for all t, this becomes

$$X(f) = \sum_{\mathcal{T}} x(t)\langle f, t\rangle$$

We see that when $\mathcal{T} = \mathbb{R}$, $\mathcal{T} = \mathbb{Z}$, or $\mathcal{T} = \mathbb{Z}_m$, this reduces to the standard definition of the Fourier transform of a complex-valued waveform, infinite sequence, or finite sequence, respectively. If $\mathcal{T} = \mathbb{R}/\mathbb{Z}$, then this definition yields the Fourier coefficients of a periodic waveform with period 1, or of a finite waveform of length 1. We say that $x(t)$ and $X(f)$ are a *Fourier transform pair*, written $x(t) \leftrightarrow X(f)$.

The following Fourier transform properties follow directly from its definition, the bihomomorphism of $\langle f, t\rangle$, and the translation-invariance of Haar measure.

a. Linearity: $ax(t) + by(t) \leftrightarrow aX(f) + bY(f)$;

b. Time reversal $\leftrightarrow$ frequency reversal: $x(-t) \leftrightarrow X(-f)$
(since $\langle -f, -t\rangle = \langle f, t\rangle$);

c. Translation $\leftrightarrow$ modulation: $x(t - \tau) \leftrightarrow \langle f, \tau\rangle X(f)$
(since $\langle f, t + \tau\rangle = \langle f, t\rangle\langle f, \tau\rangle$);

d. Modulation $\leftrightarrow$ translation: $\langle \phi, t\rangle x(t) \leftrightarrow X(f + \phi)$
(since $\langle f + \phi, \tau\rangle = \langle f, t\rangle\langle \phi, t\rangle$);

e. Convolution $\leftrightarrow$ multiplication: $x(t) * y(t) \leftrightarrow X(f)Y(f)$
(since $\langle f, t\rangle = \langle f, \tau\rangle\langle f, t-\tau\rangle$);

f. Integral = value at 0: $\int_{\mathcal{T}} x(t)d\mu(t) = X(0)$
(since $\langle 0, t\rangle = 1$).

As engineers know, another very useful transform pair is:

g. delta function $\leftrightarrow$ constant: $\delta(t) \leftrightarrow 1$,

where the delta function $\delta(t)$ stands for the identity waveform under convolution: $x * \delta = x$ for all $x \in L_1(\mathcal{T})$. As mathematicians know, there exists such an identity function in $L_1(\mathcal{T})$ if $\mathcal{T}$ is discrete, namely the *Kronecker delta function* ($\delta(t) = 1$ if $t = 0$, else 0), but not if $\mathcal{T}$ is indiscrete. In this case, one can define $\delta(t)$ as the limit of a series of approximate identity waveforms $\epsilon(t) \in L_1(\mathcal{T})$ concentrated on neighborhoods of $0 \in \mathcal{T}$ such that the difference between x and $x * \epsilon$ becomes arbitrarily small. For example, for $\mathcal{T} = \mathbb{R}$, the series $\epsilon(t)$ might be a series of Gaussian probability densities with variances $\sigma^2 \to 0$. The *Dirac delta function* $\delta(t)$ is a symbolic expression for this limit in $\mathcal{T} = \mathbb{R}$. We may define a Dirac delta function $\delta(t)$ analogously for every indiscrete group $\mathcal{T}$. Then property (g) follows from the continuity of $f(t) = \langle f, t\rangle$ and the fact that $\langle f, 0\rangle = 1$. Of course, the constant function 1 is not in $L_1(\mathcal{F})$ either, but is similarly the limit of $L_1(\mathcal{F})$ functions. For example in $\mathcal{F} = \mathbb{R}$, this might be the limit of Gaussian functions with value 1 at $f = 0$ and variances $\sigma^2 \to \infty$.

Another useful Fourier transform relation holds for product functions. Let $\mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_m$ be a direct product group, and let $x(t)$ a product function. That is, if $t = (t_1, \cdots, t_m)$, then

$$x(t) \;=\; x_1(t_1)\cdots x_m(t_m)$$

for some set of functions $\{x_i(t_i), t_i \in \mathcal{T}_i\}$ defined on the groups $\{\mathcal{T}_i\}$. Then it follows immediately from the factorization $\langle f, t\rangle = \langle f_1, t_1\rangle \cdots \langle f_m, t_m\rangle$ that the transform of $x(t)$ is also a product function,

$$X(f) \;=\; X_1(f_1)\cdots X_m(f_m)$$

where the groups $\{\mathcal{F}_i\}$ are the character groups of the groups $\{\mathcal{T}_i\}$. In summary, we have the following relation:

h. Product function $\leftrightarrow$ product function:

$$x_1(t_1)\cdots x_m(t_m) \leftrightarrow X_1(f_1)\cdots X_m(f_m).$$

5. Orthogonality relations

The most important properties of group characters are their orthogonality relations. These relations lead to the inverse Fourier transform and the unitarity properties reflected in Parseval's theorem, as discussed in this section.

Orthogonality lemma. *Let f, $f' \in \mathcal{F}$ be two distinct characters of $\mathcal{T}$. Then*

$$\int_{\mathcal{T}} \langle f,t\rangle\langle -f',t\rangle \, d\mu(t) \;=\; 0$$

Proof. Since $\langle f,t\rangle\langle -f',t\rangle = \langle f-f',t\rangle$, it suffices to evaluate $\int_{\mathcal{T}}\langle f,t\rangle \, d\mu(t)$ for $f \neq 0$. If $f \neq 0$, then there is some $\tau \in \mathcal{T}$ such that $\langle f,\tau\rangle \neq 1$ (else $f = 0$). Thus we have

$$\int_{\mathcal{T}} \langle f,t\rangle \, d\mu(t) \;=\; \int_{\mathcal{T}} \langle f,t+\tau\rangle \, d\mu(t) \;=\; \langle f,\tau\rangle \int_{\mathcal{T}} \langle f,t\rangle \, d\mu(t)$$

by translation-invariance and $\langle f,t+\tau\rangle = \langle f,t\rangle\langle f,\tau\rangle$. Since $\langle f,\tau\rangle \neq 1$, we conclude that $\int_{\mathcal{T}}\langle f,t\rangle \, d\mu(t) = 0$ if $f \neq 0$. ∎

Similarly, if t, $t' \in \mathcal{T}$ are two distinct characters of $\mathcal{F}$, then we have the orthogonality relation

$$\int_{\mathcal{F}} \langle f,t\rangle\langle f,-t'\rangle \, d\mu(f) \;=\; 0$$

If $\mathcal{T}$ is compact, then $\mu(\mathcal{T}) = \int_{\mathcal{T}} d\mu(t)$ is finite, $\mathcal{F}$ is discrete, and thus we have

$$\int_{\mathcal{T}} \langle f,t\rangle\langle -f',t\rangle \, d\mu(t) \;=\; \mu(\mathcal{T})\,\delta(f-f')$$

where $\delta(f-f')$ is the Kronecker delta function. Usually μ is scaled so that $\mu(\mathcal{T}) = 1$, unless $\mathcal{T}$ is finite in which case $\mu(\mathcal{T}) = |\mathcal{T}|$. If on the other hand $\int_{\mathcal{T}} d\mu(t)$ is infinite, then this integral becomes a Dirac-type delta function,

$$\int_{\mathcal{T}} \langle f,t\rangle\langle -f',t\rangle \, d\mu(t) \;=\; \delta(f-f')$$

where we assume that μ is scaled so that no multiplicative constant is needed. In summary, we have the Fourier transform pair

i. constant $\leftrightarrow$ delta function: $1 \leftrightarrow \mu(\mathcal{T})\delta(f)$, $\mathcal{T}$ compact; else $1 \leftrightarrow \delta(f)$.

From property (**i**) we obtain the *inverse Fourier transform*, which in these two cases is given by:

$$x(t) \;=\; \begin{cases} \dfrac{1}{\mu(\mathcal{T})} \displaystyle\sum_{\mathcal{F}} X(f)\langle -f,t\rangle, & \mathcal{T} \text{ compact} \\[2ex] \displaystyle\int_{\mathcal{F}} X(f)\langle -f,t\rangle \, d\mu(f), & \text{otherwise} \end{cases}$$

For example, the latter expression can be verified as follows:

$$\begin{aligned} \int_{\mathcal{T}} x(t)\langle f,t\rangle \, d\mu(t) \;&=\; \int_{\mathcal{F}} X(f') \, d\mu(f) \int_{\mathcal{T}} \langle -f',t\rangle\langle f,t\rangle \;\, d\mu(t) \\ &=\; \int_{\mathcal{F}} \delta(f-f')X(f') \, d\mu(f) \\ &=\; X(f) \end{aligned}$$

The inverse Fourier transform yields duals of the Fourier transform relations above. That is, up to possible factors of $1/\mu(\mathcal{T})$, we have:

j. Multiplication $\leftrightarrow$ convolution: $x(t)y(t) \leftrightarrow X(f) * Y(f)$;

k. Value at 0 = integral: $x(0) = \int_{\mathcal{F}} X(f)d\mu(f)$.

Because the Fourier kernel $\langle f,t\rangle$ is on the complex unit circle $\mathbb{C}$, we have the relation $\langle -f,t\rangle = \langle f,t\rangle^{-1} = \langle f,t\rangle^*$, where $\langle f,t\rangle^*$ is the complex conjugate of $\langle f,t\rangle$. This leads to the additional transform pairs:

l. Conjugate $\leftrightarrow$ reversed conjugate: $x^*(t) \leftrightarrow X^*(-f)$;

m. Cross-correlation $\leftrightarrow$ cross-power spectrum: $x(t) * y^*(-t) \leftrightarrow X(f)Y^*(f)$;

n. Real part $\leftrightarrow$ even part: $\frac{1}{2}(x(t) + x^*(t)) \leftrightarrow \frac{1}{2}(X(f) + X^*(-f))$;

o. Imaginary part $\leftrightarrow$ odd part: $\frac{1}{2}(x(t) - x^*(t)) \leftrightarrow \frac{1}{2}(X(f) - X^*(-f))$.

and their duals. Furthermore, the combination of relations (**k**) and (**m**) yields the relation:

p. Inner product = inner product: $\int_{\mathcal{T}} x(t)y^*(t)\,d\mu(t) = \int_{\mathcal{F}} X(f)Y^*(f)\,d\mu(f)$,

which in the particular case $x = y$ yields the well-known *Parseval's theorem*:

q. L_2-norm = L_2-norm: $\int_{\mathcal{T}} |x(t)|^2\,d\mu(t) = \int_{\mathcal{F}} |X(f)|^2\,d\mu(f)$.

The fact that the L_2-norm of x is equal to the L_2-norm of X (up to a factor of $1/\mu(\mathcal{T})$) shows that the Fourier transform operator and its inverse are unitary.

If $\mathcal{T}$ is finite, then $\langle f,t\rangle$ and $\langle -f,t\rangle$ are $|\mathcal{T}| \times |\mathcal{T}|$ complex-valued matrices M and M^*, respectively, where M^* is the conjugate transpose of M. These are called the *character tables* of $\mathcal{T}$. The orthogonality theorem shows that

$$MM^* = M^*M = |\mathcal{T}|\, I_{|\mathcal{T}|}$$

where $I_{|\mathcal{T}|}$ is the $|\mathcal{T}| \times |\mathcal{T}|$ identity matrix. This implies that M is a scaled unitary matrix; that is, the rows or columns of M (the character vectors) form an orthogonal basis of complex $|\mathcal{T}|$-space. Furthermore this implies Parseval's theorem because, regarding $x(t)$ and $X(f)$ as column $|\mathcal{T}|$-tuples $\boldsymbol{x}$ and $\boldsymbol{X}$, we have $\boldsymbol{X} = M\boldsymbol{x}$ and $\|\boldsymbol{X}\|^2 = \boldsymbol{X}^*\boldsymbol{X} = \boldsymbol{x}^*M^*M\boldsymbol{x} = |\mathcal{T}|\,\boldsymbol{x}^*\boldsymbol{x} = |\mathcal{T}|\,\|\boldsymbol{x}\|^2$.

Example 2. The character table of $\mathbb{Z}_2$ is a 2×2 matrix M_2 defined by the relation $\langle f,t \rangle = \omega^{ft}$, where $\omega = -1$ is a square root of 1; i.e.,

$$M_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$$

Note that the rows (or columns) of M_2 are orthogonal, and thus $M_2^* M_2 = 2I_2$.

The character table of $(\mathbb{Z}_2)^n$ is a $2^n \times 2^n$ matrix M_{2^n}, which can be obtained as follows. If $t = (t_1, \cdots, t_n)$ and $f = (f_1, \cdots, f_n)$, then

$$\langle f,t \rangle = \langle f_1,t_1 \rangle \cdots \langle f_n,t_n \rangle = \omega^{f_1 t_1 + \cdots + f_n t_n} = \omega^{f \cdot t}$$

where again $\omega = -1$ and $f \cdot t$ is the dot product $f_1 t_1 + \cdots + f_n t_n$. This implies that M_{2^n} is the n-fold tensor product of M_2 with itself:

$$M_{2^n} = (M_2)^{\otimes n}$$

The matrix M_{2^n} thus consists of 2^n orthogonal rows (or columns) whose values are all in $\{\pm 1\}$, and is therefore an orthogonal basis of either real or complex 2^n-space. It is called a *Hadamard matrix*. The Fourier transform $\boldsymbol{X} = M_{2^n} \boldsymbol{x}$ is sometimes called a *Walsh transform*. Under the homomorphism from the multiplicative group $\{\pm 1\}$ to $\mathbb{Z}_2$, M_{2^n} becomes a binary $2^n \times 2^n$ matrix whose rows may be used to generate the binary *Reed-Muller codes* of length 2^n. □

6. Fourier transforms over other fields

As Blahut has emphasized [1], the Fourier transform may be generalized to fields $\mathbb{F}$ other than the complex field $\mathbb{C}$, particularly finite fields. In this section we discuss which properties of the Fourier transform carry over to more general fields.

Given dual character groups $\mathcal{T}$ and $\mathcal{F}$ and their pairing image group

$$\langle \mathcal{F}, \mathcal{T} \rangle = \{\langle f,t \rangle : f \in \mathcal{F}, t \in \mathcal{T}\} \subseteq \mathbb{C}$$

one may replace the pairing $\mathcal{F} \times \mathcal{T} \to \mathbb{C}$ by $\mathcal{F} \times \mathcal{T} \to M \subseteq \mathbb{F}$ if and only if $\mathbb{F}$ has a multiplicative subgroup M that is topologically isomorphic to $\langle \mathcal{F}, \mathcal{T} \rangle$.

In the real field $\mathbb{R}$, the only nontrivial multiplicative subgroup of $\mathbb{R}^* = \mathbb{R} \backslash \{0\}$ is the binary group $\{\pm 1\}$. Thus Fourier transforms over $\mathbb{R}$ exist if and only if $\mathcal{T}$ is a group of elements of order 2, such as the groups $(\mathbb{Z}_2)^n$ discussed in Example 2.

In a finite field $\mathbb{F}$ of order q, the multiplicative group $\mathbb{F}^* = \mathbb{F} \backslash \{0\}$ is a cyclic abelian group of order $q-1$; i.e., it is isomorphic to $\mathbb{Z}_{q-1}$. Therefore $\mathbb{F}^*$ has subgroups $M = \{\omega^k : 0 \leq k \leq m-1\}$ isomorphic to $\mathbb{Z}_m$ for all m such that m divides $q-1$, where ω is an element of $\mathbb{F}$ of multiplicative order m. Because a discrete Fourier transform (DFT) on m points is based on a time axis $\mathbb{Z}_m$ and the pairing $\langle f,t \rangle = \omega^{ft}$, any m-point DFT such that m divides $q-1$ may be computed over $\mathbb{F}$ as

$$X(f) = \sum_{\mathcal{T}} x(t) \omega^{ft}$$

This observation leads to cyclic block codes of length m over finite fields of order q; for example, the *Reed-Solomon codes*.

If we review the development of the properties of the Fourier transform in § 5, we find that properties (**a**)–(**k**) and the orthogonality relations continue to hold for discrete Fourier transforms over finite fields. On the other hand, properties (**l**)–(**q**), including Parseval's theorem, depend on being able to define a conjugation operation (involution) on $\mathbb{F}$ that maps to an inverse on M, and such an operation does not generally exist in finite fields.

7. Sampling and aliasing

From the subgroup/quotient group duality theorem, one expects that if $\mathcal{T}$ and $\mathcal{F}$ are dual character groups, then one may take a subgroup $\mathcal{S}$ (respectively, a quotient group $\mathcal{T}/\mathcal{S}$) as a time domain, its character group $\mathcal{F}/\mathcal{S}^\perp$ (respectively, $\mathcal{S}^\perp$) as a frequency domain, and that there will be a Fourier transform relationship between them. One further expects a relationship between this Fourier transform and the original transform defined on $\mathcal{T}$ and $\mathcal{F}$. For example, if the time domain is $\mathbb{Z}$, then the frequency domain is $\mathbb{R}/\mathbb{Z}$, and one gets the "discrete-time" Fourier transform along with its well-known relations to the "continuous-time" Fourier transform. We briefly sketch these relationships in general terms.

Our main group-theoretic tools are coset decompositions of $\mathcal{T}$ and $\mathcal{F}$, as follows. Let $[\mathcal{T}/\mathcal{S}]$ be a (compact) set of coset representatives for the cosets of $\mathcal{S}$ in $\mathcal{T}$. Then every $t \in \mathcal{T}$ can be uniquely written as $t = s + \tau$ where $s \in \mathcal{S}$ and $\tau \in [\mathcal{T}/\mathcal{S}]$. Correspondingly $\langle f, t\rangle = \langle f, s\rangle\langle f, \tau\rangle$. Thus we write $\mathcal{T} = \mathcal{S} + [\mathcal{T}/\mathcal{S}]$. Similarly, $\mathcal{F} = \mathcal{S}^\perp + [\mathcal{F}/\mathcal{S}^\perp]$. We will assume that both $\mathcal{S}$ and $\mathcal{S}^\perp$ are discrete.

Let $x(t)$ be *$\mathcal{S}$-periodic*, meaning that $x(t)$ is constant on cosets of $\mathcal{S}$ in $\mathcal{T}$. Then

$$\begin{aligned} X(f) &= \int_{\mathcal{T}} x(t)\langle f, t\rangle \, d\mu(t) \\ &= \int_{[\mathcal{T}/\mathcal{S}]} \int_{\mathcal{S}} x(s+\tau)\langle f, s+\tau\rangle \, d\mu(s+\tau) \\ &= \left(\int_{[\mathcal{T}/\mathcal{S}]} x(\tau)\langle f, \tau\rangle \, d\mu(\tau)\right) \left(\int_{\mathcal{S}} \langle f, s\rangle \, d\mu(s)\right) \end{aligned}$$

By the orthogonality lemma, the function $\Pi_{\mathcal{S}}(f) = \int_{\mathcal{S}} \langle f, s\rangle \, d\mu(s)$ is equal to 0 unless $\langle f, s\rangle = 1$ for all $s \in \mathcal{S}$; i.e., unless $f \in \mathcal{S}^\perp$. It is thus concentrated on the elements of $\mathcal{S}^\perp$ and comprises a set of Dirac or Kronecker delta functions:

$$\Pi_{\mathcal{S}}(f) = \begin{cases} \dfrac{1}{\mu(\mathcal{S})} \displaystyle\sum_{m \in \mathcal{S}^\perp} \delta(f-m) & \text{if } \mathcal{S} \text{ is compact} \\ \displaystyle\sum_{m \in \mathcal{S}^\perp} \delta(f-m) & \text{otherwise} \end{cases}$$

Such an $\mathcal{S}^\perp$-periodic impulsive function $\Pi_{\mathcal{S}}(f)$ is called a *picket-fence function*.

Now we have

$$X(f) = X'(f)\Pi_{\mathcal{S}}(f)$$

where:

$$X'(f) = \int_{[\mathcal{T}/\mathcal{S}]} x(\tau)\langle f,\tau\rangle d\mu(\tau), \qquad \text{for } f \in \mathcal{S}^\perp$$

an integral over one "period" $[\mathcal{T}/\mathcal{S}]$, is the Fourier transform of $x(\tau)$ over $[\mathcal{T}/\mathcal{S}]$. Thus the Fourier transform $X(f)$ is an impulsive function whose support is the character group $\mathcal{S}^\perp$ of $\mathcal{T}/\mathcal{S}$, with the "Fourier coefficients" $\{X'(f), f \in \mathcal{S}^\perp\}$.

The inverse Fourier transform, up to factors of $\mu(\mathcal{S})$, is then given by:

$$x(t) = \int_{\mathcal{F}} X'(f)\Pi_{\mathcal{S}}(f)\langle -f,t\rangle\, d\mu(f) = \sum_{m\in\mathcal{S}^\perp} X'(f)\langle -m,t\rangle$$

which is $\mathcal{S}$-periodic since $\langle -m, t+s\rangle = \langle -m,t\rangle$ for all $m \in \mathcal{S}^\perp$ and $s \in \mathcal{S}$. We conclude that the transform of an $\mathcal{S}$-periodic function is $\mathcal{S}^\perp$-impulsive. Similarly, the transform of an $\mathcal{S}$-impulsive function is $\mathcal{S}^\perp$-periodic.

Since the picket-fence function $\Pi_{\mathcal{S}}(f)$ is both $\mathcal{S}^\perp$-impulsive and $\mathcal{S}^\perp$-periodic, it follows that its transform $\pi_{\mathcal{S}}(t)$ is both $\mathcal{S}$-impulsive and $\mathcal{S}$-periodic; i.e., it is a picket-fence function in the time domain. This determines $\pi_{\mathcal{S}}(t)$ up to a multiplicative constant, which turns out to be equal to $1/\mu(\mathcal{T}/\mathcal{S})$. (If $\mathcal{S}^\perp$ is finite, then $\mu(\mathcal{T}/\mathcal{S}) = |\mathcal{S}^\perp|$.) The fact that the transform of a picket-fence function is a picket-fence function is called the *picket-fence miracle*.

Picket-fence miracle. *Let $\mathcal{S}$ and $\mathcal{S}^\perp$ be discrete orthogonal subgroups of $\mathcal{T}$ and $\mathcal{F} = \mathcal{T}^\wedge$, respectively. Then the following two picket-fence functions are transform pairs:*

$$\pi_{\mathcal{S}}(t) = \sum_{k\in\mathcal{S}} \delta(t-k)$$

$$\Pi_{\mathcal{S}}(f) = \frac{1}{\mu(\mathcal{T}/\mathcal{S})} \sum_{m\in\mathcal{S}^\perp} \delta(f-m)$$

where the delta functions are Kronecker or Dirac according to the respective topologies of $\mathcal{T}$ and $\mathcal{F}$.

For example, let Λ be a lattice in $\mathbb{R}^n$, and let $\Lambda^\perp$ be its dual lattice in $\mathbb{R}^n$. Then the transform $X'(f)$ of *one period* of a periodic function $x(t)$ is an integral over a fundamental region $[\mathbb{R}^n/\Lambda] = \mathbb{R}(\Lambda)$ of Λ, whose measure $\mu(\mathbb{R}^n/\Lambda)$ is the volume $V(\Lambda)$ of Λ. The transform $X(f)$ of $x(t)$ is the $\Lambda^\perp$-impulsive function:

$$X(f) = \sum_{m\in\Lambda^\perp} X'(m)\,\delta(f-m)$$

where:

$$X'(m) = \int_{\mathbb{R}(\Lambda)} x(t)\exp\{-2\pi j(m\cdot t)\}\, dt, \qquad \text{for } m \in \Lambda^\perp$$

with the inverse transform given by:

$$x(t) = \frac{1}{V(\Lambda)} \sum_{m\in\Lambda^\perp} X'(m)\exp\{2\pi j(m\cdot t)\}$$

This generalizes the usual discrete-time Fourier transform relations (in which case $\mathcal{T} = \tau\mathbb{Z}$ and $\mathcal{F} = \mathbb{R}/\tau^{-1}\mathbb{Z}$) to n-dimensional time axes and Λ-periodic or Λ-impulsive functions.

From the picket-fence theorem we may easily derive the aliasing theorem, as it is called in signal theory (or the decimation theorem, as it is sometimes called in the discrete case). Let $x(t)$ be a function with transform $X(f)$. Sampling of $x(t)$ on $\mathcal{S}$ can be accomplished by multiplying $x(t)$ by the picket-fence function $\pi_{\mathcal{S}}(t)$ to get an $\mathcal{S}$-impulsive function $s_{\mathcal{S}}(t)$, as follows:

$$s_{\mathcal{S}}(t) \;=\; x(t)\pi_{\mathcal{S}}(t) \;=\; \sum_{k\in\mathcal{S}} x(k)\delta(t-k)$$

Since $\pi_{\mathcal{S}}(t)$ and $\Pi_{\mathcal{S}}(f)$ are a transform pair, it follows that the Fourier transform $\mathcal{S}_{\mathcal{S}}(f)$ is given by:

$$\mathcal{S}_{\mathcal{S}}(f) \;=\; X(f) * \Pi_{\mathcal{S}}(f) \;=\; \frac{1}{\mu(\mathcal{T}/\mathcal{S})} \sum_{m\in\mathcal{S}^\perp} X(f-m)$$

which is $\mathcal{S}^\perp$-periodic. In summary, we have the following theorem:

Sampling/aliasing theorem. *The Fourier transform of an $\mathcal{S}$-sampled function $s_{\mathcal{S}}(t) = x(t)\pi_{\mathcal{S}}(t)$, which is $\mathcal{S}$-impulsive, is the $\mathcal{S}^\perp$-aliased function*

$$\mathcal{S}_{\mathcal{S}}(f) \;=\; \frac{1}{\mu(\mathcal{T}/\mathcal{S})} \sum_{m\in\mathcal{S}^\perp} X(f-m)$$

which is $\mathcal{S}^\perp$-periodic.

For example, taking $\mathcal{S}$ as the trivial subgroup $\{0\}$, we find that the transform of $x(0)\delta(t)$ is the constant ($\mathcal{F}$-periodic) function $(1/\mu(\mathcal{T}))\sum_{m\in\mathcal{F}} X(f)$, which of course also follows from properties (**g**) and (**k**).

These results may be straightforwardly extended to cosets $\mathcal{S}+\tau$ or $\mathcal{S}^\perp+\phi$ and dualized, as follows:

r. Picket-fence miracle:

$$\sum_{k\in\mathcal{S}+\tau} \delta(t-k) \quad\leftrightarrow\quad \frac{\langle f,\tau\rangle}{\mu(\mathcal{T}/\mathcal{S})} \sum_{m\in\mathcal{S}^\perp} \delta(f-m);$$

s. Sampling $\leftrightarrow$ aliasing:

$$\sum_{k\in\mathcal{S}+\tau} x(k)\delta(t-k) \quad\leftrightarrow\quad \frac{\langle f,\tau\rangle}{\mu(\mathcal{T}/\mathcal{S})} \sum_{m\in\mathcal{S}^\perp} X(f-m);$$

t. Modulated $\mathcal{S}$-periodic $\leftrightarrow$ $(\mathcal{S}^\perp+\phi)$-impulsive:

$$\langle -f,t\rangle \sum_{k\in\mathcal{S}} x(t-k) \quad\leftrightarrow\quad \frac{1}{\mu(\mathcal{T}/\mathcal{S})} \sum_{m\in\mathcal{S}^\perp+\phi} X(m)\delta(f-m).$$

8. The Poisson summation formula and the MacWilliams identities

As an immediate corollary of the sampling/aliasing theorem, we obtain the following important relation.

Poisson summation formula. *Let $x(t)$ and $X(f)$ be a transform pair defined on $\mathcal{T}$ and $\mathcal{F}$, and let $\mathcal{S}$ and $\mathcal{S}^\perp$ be discrete orthogonal subgroups of $\mathcal{T}$ and $\mathcal{F}$, respectively. Then*

$$\sum_{k\in\mathcal{S}} x(k) = \frac{1}{\mu(\mathcal{T}/\mathcal{S})} \sum_{m\in\mathcal{S}^\perp} X(m)$$

Proof. The right side is the value of the $\mathcal{S}^\perp$-aliased function $\mathcal{S}_\mathcal{S}(f)$ at $f = 0$, which is:

$$\mathcal{S}_\mathcal{S}(0) = \int_\mathcal{T} s_\mathcal{S}(t) d\mu(t) = \sum_{k\in\mathcal{S}} x(k)$$

■

From the Poisson summation formula follow the MacWilliams identities of coding theory [9], as well as the Poisson-Jacobi identities of lattice theory [4]. The following argument is essentially due to Ericson [6], who observed that it shows that "MacWilliams identity is actually a property of abelian groups rather than a property of linear spaces." We may also quote Conway and Sloane [4]:

> One may regard [the Poisson-Jacobi identities], as well as the MacWilliams identity for weight enumerators of codes, as consequences of the general version of the Poisson summation formula, which states that the sum of a function over a linear space is equal to the sum of the Fourier transform of the function over the dual space ...

Let $\mathcal{T}$ and $\mathcal{F} = \mathcal{T}^\wedge$ be dual character groups, and let $\{x(t), t \in \mathcal{T}\}$ be a set of indeterminates indexed by $\mathcal{T}$. The transform of $x(t)$ is defined by

$$X(f) = \int_\mathcal{T} x(t)\langle f, t\rangle \, d\mu(t)$$

which yields a set of indeterminates $\{X(f), f \in \mathcal{F}\}$ that is dual to the primal set of indeterminates $\{x(t), t \in \mathcal{T}\}$.

For example, if $\mathcal{T} = \mathbb{Z}_2$, then $X(0) = x(0) + x(1)$ and $X(1) = x(0) - x(1)$. As another example, take $\mathcal{T} = \mathbb{Z}_3$, in which case $X(0) = x(0) + x(1) + x(2)$, $X(1) = x(0) + \omega x(1) + \overline{\omega} x(2)$ and $X(2) = x(0) + \overline{\omega} x(1) + \omega x(2)$, where ω is a cube root of unity and $\overline{\omega} = \omega^*$.

Consider now the set $\mathcal{T}^n$ of all n-tuples over $\mathcal{T}$. The *complete weight enumerator* of an n-tuple $\boldsymbol{t} \in \mathcal{T}^n$ is defined as the product

$$\boldsymbol{x}(\boldsymbol{t}) = \prod_k x(t_k)$$

Thus $\boldsymbol{x}(\boldsymbol{t})$ is a monomial in the set of indeterminates $\{x(t), t \in \mathcal{T}\}$ in which each $x(t)$ appears with an exponent equal to the number of times that t occurs

in t. Moreover, $x(t)$ is a product function; hence by property (h) its transform is also a product function, given by

$$X(f) = \prod_k X(f_k)$$

where $X(f)$ is the enumerator of an n-tuple $f \in \mathcal{F}^n$ corresponding to the dual indeterminate set $\{X(f), f \in \mathcal{F}\}$.

Let $\mathcal{S}$ and $\mathcal{S}^\perp$ be discrete orthogonal subgroups of $\mathcal{T}^n$ and $\mathcal{F}^n$, respectively. Their complete weight enumerators are defined as

$$x(\mathcal{S}) = \sum_{\mathcal{S}} x(t)$$

$$X(\mathcal{S}^\perp) = \sum_{\mathcal{S}^\perp} X(f)$$

A generalized MacWilliams identity for complete weight enumerators then follows directly from the Poisson summation formula.

Generalized MacWilliams identities (complete weight enumerators). *If $\mathcal{S}$ and $\mathcal{S}^\perp$ are discrete orthogonal subgroups of $\mathcal{T}^n$ and $\mathcal{F}^n$, respectively, then their complete weight enumerators are related by*

$$x(\mathcal{S}) = \frac{1}{\mu(\mathcal{T}^n/\mathcal{S})} X(\mathcal{S}^\perp)$$

For the case in which $\mathcal{T}$ and $\mathcal{F}$ are finite, we have $\mu(\mathcal{T}^n/\mathcal{S}) = |\mathcal{S}^\perp|$, and we get the usual MacWilliams identity for complete weight enumerators of dual codes.

For the case in which $\mathcal{T} = \mathcal{F} = \mathbb{R}$ and $\mathcal{S}$ and $\mathcal{S}^\perp$ are dual n-dimensional lattices, we have $\mu(\mathcal{T}^n/\mathcal{S}) = V(\mathcal{S})$, and we get a corresponding identity for lattices.

A general method of obtaining other enumerators from complete weight enumerators has been given by Ericson and Zinoviev [7]. Partition $\mathcal{T}$ into disjoint subsets $\{\mathcal{T}_j\}$ and $\mathcal{F}$ into disjoint subsets $\{\mathcal{F}_i\}$. Such a partition pair is said to be *Fourier-invariant* if for all i, j the transform of the indicator function $\varphi(\mathcal{T}_j)$ of $\mathcal{T}_j$, given by

$$\vartheta_j(f) = \int_{\mathcal{T}} \varphi(\mathcal{T}_j)\langle f, t\rangle \, d\mu(t)$$

depends only on the subset $\mathcal{F}_i$ that contains f, and similarly for the inverse transform. The constants

$$\{K_{ij}\} = \{\vartheta_j(f), \quad f \in \mathcal{F}_i\}$$

$$\{L_{ji}\} = \left\{\int_{\mathcal{F}} \varphi(\mathcal{F}_i)\langle f, -t\rangle \, d\mu(f), \quad t \in \mathcal{T}_i\right\}$$

are called *generalized Krawtchouk coefficients*. For example, if $\mathcal{T}$ and $\mathcal{F}$ are both finite, then the "Hamming partitions" with $\mathcal{T}_0 = \{0\}$, $\mathcal{T}_1 = \mathcal{T}\backslash\{0\}$, $\mathcal{F}_0 = \{0\}$,

$\mathcal{F}_1 = \mathcal{F}\backslash\{0\}$ are Fourier-invariant. The generalized Krawtchouk coefficients K_{ij} thus form the matrix

$$K = \begin{bmatrix} 1 & |\mathcal{T}| - 1 \\ 1 & -1 \end{bmatrix}$$

The matrix L of inverse coefficients L_{ji} is also equal to K, and $KL = |\mathcal{T}|I_2$.

The generalized Krawtchouk coefficients may be thought of as defining a pairing $\langle \mathcal{F}_i, \mathcal{T}_j \rangle$ between the partition subsets $\{\mathcal{F}_i\}$ and the partition subsets $\{\mathcal{T}_j\}$. For any function $x(t)$ that is constant on the partition subsets $\{\mathcal{T}_j\}$, one derives from the Fourier transform a *generalized Krawtchouk transform*

$$X(\mathcal{F}_i) = \sum_j x(\mathcal{T}_j)\langle \mathcal{F}_i, \mathcal{T}_j \rangle$$

which is constant on the subsets $\{\mathcal{F}_i\}$, and a corresponding inverse transform. For example, in the finite case, we have

$$x(\mathcal{T}_j) = \frac{1}{|\mathcal{T}|} \sum_i X(\mathcal{F}_i)\langle \mathcal{F}_i, -\mathcal{T}_j \rangle$$

Finally, if one regards $\{x(\mathcal{T}_j)\}$ and $\{X(\mathcal{F}_i)\}$ as dual sets of indeterminates defined on the partition subsets and related by the generalized Krawtchouk transform, then one gets an identity for the enumerators of such Fourier-invariant partition subsets.

Generalized MacWilliams identities (Fourier-invariant enumerators). *If $\mathcal{S}$ and $\mathcal{S}^\perp$ are discrete orthogonal subgroups of $\mathcal{T}^n$ and $\mathcal{F}^n$, respectively, and $\{\mathcal{T}_j\}$ and $\{\mathcal{F}_i\}$ are a Fourier-invariant pair of partitions of $\mathcal{T}$ and $\mathcal{F}$, respectively, then their enumerators with respect to these partitions are related by*

$$\boldsymbol{x}(\mathcal{S}) = \frac{1}{\mu(\mathcal{T}^n/\mathcal{S})} \boldsymbol{X}(\mathcal{S}^\perp)$$

where $\{x(\mathcal{T}_j)\}$ and $\{X(\mathcal{F}_i)\}$ are sets of indeterminates defined on the partition subsets that are related by the generalized Krawtchouk transform defined above.

For example, in the case of Hamming partitions with $\mathcal{T}$ finite, we have

$$\begin{aligned} X(0) &= x(0) + (|\mathcal{T}| - 1)x(1) \\ X(1) &= x(0) - x(1) \\ x(0) &= (X(0) + (|\mathcal{T}| - 1)X(1))/|\mathcal{T}| \\ x(1) &= (X(0) - X(1))/|\mathcal{T}| \end{aligned}$$

The generalized Krawtchouk transform becomes the usual MacWilliams transform, and the identity above becomes the usual MacWilliams identity for Hamming weight enumerators.

9. Subgroup chains and fast Fourier transforms

Fast Fourier transforms [2] exist when there exists a subgroup chain

$$\{0\} = \mathcal{S}_0 \subseteq \mathcal{S}_1 \subseteq \cdots \subseteq \mathcal{S}_{r-1} \subseteq \mathcal{S}_r = \mathcal{T}$$

If such a chain exists, then there also exists a dual subgroup chain

$$\{0\} = \mathcal{S}_r^\perp \subseteq \mathcal{S}_{r-1}^\perp \subseteq \cdots \subseteq \mathcal{S}_1^\perp \subseteq \mathcal{S}_0^\perp = \mathcal{F}$$

in which the quotients $\mathcal{S}_{j-1}^\perp/\mathcal{S}_j^\perp$ are isomorphic to $\mathcal{S}_j/\mathcal{S}_{j-1}$ for $j = 1, 2, \ldots, r$.

For all $j = 1, 2, \ldots, r$, let us fix sets of coset representatives $[\mathcal{S}_j/\mathcal{S}_{j-1}]$ for the cosets of $\mathcal{S}_{j-1}$ in $\mathcal{S}_j$. Then every $t \in \mathcal{T}$ has a unique chain coset decomposition

$$t = t_1 + t_2 + \cdots t_r, \qquad t_j \in [\mathcal{S}_j/\mathcal{S}_{j-1}] \text{ for } j = 1, 2, \ldots, r$$

and we may therefore establish a one-to-one correspondence between $\mathcal{T}$ and the Cartesian product $[\mathcal{S}_1/\mathcal{S}_0] \times \cdots \times [\mathcal{S}_r/\mathcal{S}_{r-1}]$. Similarly a one-to-one correspondence may be established between $\mathcal{F}$ and $[\mathcal{S}_{r-1}^\perp/\mathcal{S}_r^\perp] \times \cdots \times [\mathcal{S}_0^\perp/\mathcal{S}_1^\perp]$, where $[\mathcal{S}_{j-1}^\perp/\mathcal{S}_j^\perp]$ are fixed sets of coset representatives for the cosets of $\mathcal{S}_j^\perp$ in $\mathcal{S}_{j-1}^\perp$.

From the orthogonality properties of dual chains, we have $\langle f_i, t_j \rangle = 1$ if $i > j$. Therefore

$$\left\langle \sum_i f_i, \sum_j t_j \right\rangle = \prod_j \prod_{i \le j} \langle f_i, t_j \rangle$$

Furthermore, if (and only if) $\mathcal{T}$ is the direct product of the quotients $\mathcal{S}_j/\mathcal{S}_{j-1}$, then

$$\langle f_i, t_j \rangle = 1 \qquad \text{if } i \neq j$$

and

$$\left\langle \sum_i f_i, \sum_j t_j \right\rangle = \prod_i \langle f_i, t_i \rangle$$

The fast Fourier transform recursion is then as follows.

Initialization: $x_0(t_1, t_2, \cdots, t_r) = x(t_1 + t_2 + \cdots + t_r)$

Iteration: for $j = 1, 2, \ldots, r$, compute

$$x_j(f_1, \cdots, f_j, t_{j+1}, \ldots, t_r) = \langle f_1 + \cdots + f_j, t_{j+1} + \cdots + t_r \rangle \sum_{t_j} \langle f_j, t_j \rangle x_{j-1}(f_1, \ldots, f_{j-1}, t_j, \cdots, t_r)$$

where the *twiddle factor* $\langle f_1 + \cdots + f_j, t_{j+1} + \cdots + t_r \rangle$ is taken as 1 for $j = r$. Note that the j-th stage of the recursion involves $|\mathcal{T}|/|\mathcal{S}_j/\mathcal{S}_{j-1}|$ Fourier transforms

($|\mathcal{S}_j/\mathcal{S}_{j-1}|$ points each) and multiplications by a twiddle factor. It is straightforward to verify that the last stage of the recursion produces the Fourier transform terms:

$$x_r(f_1, \ldots, f_r) = \sum_t \langle f_1 + \cdots + f_r, t \rangle x(t) = X(f_1 + \cdots + f_r)$$

If (and only if) $\mathcal{T}$ is the direct product of the quotients $\mathcal{S}_j/\mathcal{S}_{j-1}$, $1 \le j \le r$, then $\langle f_i, t_j \rangle = 1$ if $i \neq j$, and

$$\left\langle \sum_i f_i, \sum_j t_j \right\rangle = \prod_i \langle f_i, t_i \rangle$$

In this case (and only in this case) the twiddle factors are all equal to 1, and the fast Fourier transform is of the Good-Thomas type [**2**].

Finally, note that this development does not depend on $\mathcal{T}$ being finite. For example, for real $\tau > 0$ and integer $P > 1$, consider the chain $\{0\} \subseteq P\tau\mathbb{Z} \subseteq \tau\mathbb{Z}$ and the corresponding coset decomposition $\tau\mathbb{Z} = P\tau\mathbb{Z} + [\tau\mathbb{Z}/P\tau\mathbb{Z}]$, where $[\tau\mathbb{Z}/P\tau\mathbb{Z}]$ is a set of coset representatives for $\tau\mathbb{Z}/P\tau\mathbb{Z} \cong \mathbb{Z}_P$. A τ-spaced discrete-time sequence $\{x(t), t \in \tau\mathbb{Z}\}$ can be correspondingly regarded as P parallel $P\tau$-spaced sequences $\{x_0(t_1, t_2), t_1 \in P\tau\mathbb{Z}, t_2 \in [\tau\mathbb{Z}/P\tau\mathbb{Z}]\}$, and a Fourier transform of $x(t)$ can be performed by first transforming these parallel $P\tau$-spaced sequences to obtain $x_1(f_1, f_2)$, and then performing a P-point discrete Fourier transform at each frequency $f_1 \in [0, 1/P\tau)$. Such decompositions are the basis of serial/parallel modulation techniques, such as multicarrier modulation.

Envoi. Happy birthday, Dick! How about a book on transforms and groups?

References

[1] R.E. BLAHUT, *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley 1983.

[2] ———, *Fast Algorithms for Digital Signal Processing*, Reading, MA: Addison-Wesley 1985.

[3] P. CAMION, Codes and association schemes, in *Handbook of Coding Theory*, V.S. Pless and W.C. Huffman (Editors), Amsterdam: Elsevier 1998.

[4] J.H. CONWAY AND N.J.A. SLOANE, *Sphere Packings, Lattices and Groups*, New York: Springer-Verlag 1988.

[5] PH. DELSARTE, An algebraic approach to the association schemes of coding theory, *Philips Res. Rept. Suppl.*, vol. 10, 1973.

[6] T. ERICSON, private communication, *c.* 1992.
[7] T. ERICSON AND V. ZINOVIEV, On Fourier-invariant partitions of finite abelian groups and the MacWilliams identity for group codes, *Problemy Peredachi Informatsii*, vol. 32, pp. 137–143, 1996.
[8] E. HEWITT AND K.A. ROSS, *Abstract Harmonic Analysis*, New York: Springer 1979.
[9] F.J. MACWILLIAMS AND N.J.A. SLOANE, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland 1977.
[10] L. PONTRYAGIN, *Topological Groups*, Princeton, NJ: Princeton University Press 1946.
[11] W. RUDIN, *Fourier Analysis on Groups*, New York: Wiley 1990.

Spherical Codes Generated from Self-Dual Association Schemes

Thomas Ericson

DEPARTMENT OF ELECTRICAL ENGINEERING
LINKÖPING UNIVERSITY
S-581 83 LINKÖPING, SWEDEN
thomas@isy.liu.se

ABSTRACT. The relation between spherical codes and self-dual association schemes is outlined. In particular, we show that certain spherical codes can be expressed in a natural way as a family of functions characterized by the property that their Fourier transforms have support in a fixed set of partition classes.

1. Introduction

Fourier transforms play a central role in all of communication theory. In classical circuit theory, the Fourier transform is the basic tool for characterizing transfer properties of linear time-invariant filters. The fast Fourier transform has become a standard tool for solving a large class of problems in digital signal processing. In coding theory, the MacWilliams identity and the linear programming bound are fundamental concepts; both concepts are based on Fourier transforms. It has been shown by many people — and in particular by Richard E. Blahut [1] — that many cyclic codes can naturally be defined and analyzed in terms of Fourier transforms. The list of examples could easily be extended.

In this paper, we develop a partly new aspect of Fourier transforms in connection with codes. Largely in the spirit of Blahut, we show that spherical codes generated from self-dual association schemes have a natural and simple characterization in terms of Fourier transforms. We give several explicit examples.

2. Association schemes

Let $\{\Gamma_i : 1 \leq i \leq d\}$ be a set of graphs over a common vertex set V of size v. Suppose the graphs form a *partition* of the complete graph K_v. Let Γ_0 be the empty graph (containing all the vertices, but no edges). Denote by A_i the

adjacency matrix of the graph Γ_i, for $i = 0, 1, \ldots, d$. We will assume throughout that the matrices A_i satisfy the following properties:

$$\textbf{(i)}\ \ A_0 = I, \qquad \textbf{(ii)}\ \ A_i^T = A_i, \qquad \textbf{(iii)}\ \ \sum_{i=0}^{d} A_i = J$$

where I denotes the $v \times v$ identity matrix and J denotes the $v \times v$ matrix with ones in all positions.

Consider the linear space $\mathcal{L}(A)$ formed by all linear combinations of the matrices in the family $A = \{A_0, A_1, \ldots, A_d\}$. Suppose this space is closed under matrix multiplication. This means that $\mathcal{L}(A)$ is an *algebra* and that any product $A_i A_j$ of matrices in $\mathcal{L}(A)$ has an expansion in the form

$$\textbf{(iv)}\quad A_i A_j = \sum_{k=0}^{d} p_{ijk} A_k$$

where p_{ijk} are integer constants. As the matrices A_i are symmetric, the following property must also hold:

$$\textbf{(v)}\quad A_i A_j = A_j A_i.$$

The set $A = \{A_0, A_1, \ldots, A_d\}$ of adjacency matrices satisfying these properties is said to be an *association scheme*. See [**2, 3, 4**] for more on this. The space $\mathcal{L}(A)$ is referred to as the *Bose-Mesner algebra*. We will be concerned with a very special case, but before going into this we need to develop some background.

A *spherical code* is a set of unit vectors in Euclidean space $\mathbb{R}^n$. The smallest n such that all points of the code are contained in $\mathbb{R}^n$ is called the *dimension*. The number of points M is called the *size*, and the smallest Euclidean distance $\|x - y\|$ between any pair x, y of points from the code is called the *minimum distance*. Alternatively, we may characterize the code by its *minimum squared distance* or its *maximal inner product*. These latter concepts are usually more convenient to work with. We denote the minimum squared distance by ρ and the maximal inner product by s. The relation $\rho = 2 - 2s$ always holds.

Any association scheme generates a family of spherical codes [**5, 9**]. More precisely, let A be an arbitrary element in the Bose-Mesner algebra and let U be a matrix such that its columns are unit vectors spanning an eigenspace of A. Let θ be the corresponding eigenvalue. Then we have $AU = \theta U$. Moreover, as all matrices in the Bose-Mesner algebra are *symmetric* they have only real eigenvalues, and as the algebra is *commutative* all matrices in the algebra have the same eigenspaces. Denoting by θ_i the eigenvalue of the matrix A_i corresponding to a given eigenspace, we have $A_i U = \theta_i U$, for $i = 0, 1, \ldots, d$. The matrix $E = UU^T$ is the corresponding projection matrix.

By the definition, all columns in U have equal (unit) length. It turns out [**5, 9**] that also the *rows* of U have a fixed length. Thus, upon proper normalization, the rows in the matrix U form a spherical code. Many spherical codes of practical interest can be generated in this way [**5**]. In the present context, we will focus on codes generated by self-dual Abelian schemes. A definition follows.

3. Self-dual schemes

Suppose we endow the vertex set V with the structure of an Abelian group. Let us change the notation from V to $\mathcal{A}$ in order to indicate this assumption. Let $\mathcal{P} = \{P_0, P_1, \ldots, P_d\}$ be a partition of $\mathcal{A}$ and let χ_i be the corresponding indicator functions

$$\chi_i(x) = \begin{cases} 1 & x \in P_i \\ 0 & x \neq P_i \end{cases} \qquad \text{for } i = 0, 1, \ldots, d$$

Let $\mathcal{P} = \{P_0, P_1, \ldots, P_d\}$ be symmetric, meaning that $-P_i = P_i$ for all $P_i \in P$. Then we can define graphs Γ_i with adjacency matrices A_i according to the rule

$$A_i(x, y) = \chi_i(x - y) \qquad x, y \in \mathcal{A}, \quad i = 0, 1, \ldots, d$$

The conditions **(i)**–**(iii)** will be satisfied. If conditions **(iv)** and **(v)** also hold, we have an association scheme generated by the partition $\mathcal{P} = \{P_0, P_1, \ldots, P_d\}$. An association scheme of this kind is referred to as a *symmetric Abelian scheme.*

Let $\mathcal{A}^*$ denote the group of irreducible characters over $\mathcal{A}$, with group addition defined in the natural way. It is well known that the groups $\mathcal{A}$ and $\mathcal{A}^*$ are isomorphic. One consequence of this fact is that there is a double isomorphism Ψ from the Cartesian product $\mathcal{A} \times \mathcal{A}$ into the set of units in the complex plane $\mathbb{C}$. This function can be used to define a *Fourier transform.*

Let $f : \mathcal{A} \to \mathbb{R}$ be an arbitrary function assigning a real number $f(u)$ to every element u in the group $\mathcal{A}$. Then the Fourier transform f^* is a complex-valued function defined as follows

$$f^*(u) = \sum_{v \in \mathcal{A}} \Psi(u, v) f(v)$$

Define $\tilde{f}$ by the relation $\tilde{f}(u) = f(-u)$ for all $u \in \mathcal{A}$. We then have $f^{**} = |\mathcal{A}|\tilde{f}$, which shows that the Fourier transform is invertible.

We define a multiplication in $\mathcal{L}(A)$ by the formula $f * g(u) = \sum_{v \in \mathcal{A}} f(v) g(u - v)$. This multiplication is referred to as a *convolution.* As usual, we have the relation

$$(f * g)^* = f^* g^*$$

Let $\mathcal{L}(\mathcal{P})$ denote the linear space spanned by the indicator functions χ_i corresponding to the partition $\mathcal{P} = \{P_0, P_1, \ldots, P_d\}$ and let $\mathcal{L}^*(\mathcal{P})$ be the space consisting of the Fourier transforms of the functions in $\mathcal{L}(P)$. It can be shown [**3, 4, 7**] that $\mathcal{P}$ defines an association scheme on $\mathcal{A}$ if and only if $\mathcal{L}^*(\mathcal{P}) = \mathcal{L}(\mathcal{Q})$ for some partition $\mathcal{Q}$ of $\mathcal{A}$. The partition $\mathcal{Q}$ is said to be *dual* to $\mathcal{P}$.

Note that matrix multiplication in the space $\mathcal{L}(A)$ corresponds to convolution in the space $\mathcal{L}(\mathcal{P})$. The space $\mathcal{L}^*(\mathcal{P})$ is isomorphic to $\mathcal{L}(\mathcal{P})$. In $\mathcal{L}^*(P)$ the Bose-Mesner algebra is expressed in terms of pointwise multiplication (this is sometimes also referred to as the *Schur product*).

It can be shown that the dual partition $\mathcal{Q}$ also defines an Abelian scheme. Moreover, the dual of $\mathcal{Q}$ is $\mathcal{P}$, so the relation of duality is symmetric. A partition $\mathcal{P}$ such that $\mathcal{P} = \mathcal{Q}$ is called an *$\mathcal{F}$-partition* (cf. [**8**]). The corresponding Abelian scheme is said to be *self-dual.* The present discussion will be confined to this particular case under the additional assumption that $\mathcal{P}$ is symmetric.

4. Spherical codes

Let L_i be one of the eigenspaces of $\mathcal{L}(\mathcal{P})$, let $f \in \mathcal{L}(\mathcal{P})$, and let θ be the eigenvalue corresponding to L_i. For $x \in L_i$, we have $f * x = \theta x$. Alternatively, taking the Fourier transform of both sides of this equation we obtain $f^* x^* = \theta x^*$, where now pointwise multiplication is employed. By assumption $\mathcal{L}^*(\mathcal{P}) = \mathcal{L}(\mathcal{P})$. Thus f^* is an element in $\mathcal{L}(\mathcal{P})$, and so has an expansion in the basis $\{\chi_i : 0 \leq i \leq d\}$. Let this expansion be

$$f^* = \sum_{i=0}^{d} f_i^* \chi_i$$

Inserting in the characteristic equation, we obtain $f_i^* x^*(u) = \theta x^*(u)$ for $u \in P_i$. Suppose all the coefficients f_i^* are different. It follows that x^* has support in just one of the cells P_i, and that the eigenvalue is given by $\theta = f_i^*$. Thus we have a bijective correspondence between the eigenspaces of $\mathcal{L}(\mathcal{P})$ and the cells P_i in the partition $\mathcal{P} = \{P_0, P_1, \ldots, P_d\}$. Let L_i be the eigenspace corresponding to the cell P_i. It consists simply of all real functions x for which the Fourier transform x^* has support in P_i. Thus we have $\dim L_i = |P_i|$.

In order to generate the function $x \in L_i$, we are free to specify x^* arbitrarily over P_i subject only to the restrictions that x^* must be identically zero outside P_i and that x must be real. The last requirement is satisfied if and only if $\tilde{x}^* = \bar{x}^*$, where the overbar indicates complex conjugation. As $-P_i = P_i$ by assumption, this requirement can be always met.

Consider the eigenspace L_i. Let $n = \dim L_i$ and let $x_1, x_2, \ldots, x_n$ be a set of orthogonal functions spanning L_i. By the remarks in § 2, we may choose the functions $x_i \in L_i$ such that each vector $x(u) = [x_1(u), x_2(u), \ldots, x_n(u)]$, for all $u \in \mathcal{A}$, has unit length. Some of the vectors $x(u)$ might coincide, but the set of distinct vectors forms a spherical code of dimension n.

5. Examples

Example 1. Let $\mathcal{A} = (\mathbb{Z}_q, +)$ be the additive group in the ring of integers modulo q. Let θ be a q-th root of unity. Let $\delta(\cdot)$ be the Kronecker delta function, meaning that $\delta(u) = 1$ for $u = 0$ and $\delta(u) = 0$ for $u \neq 0$.

It can be easily checked that the partition $\mathcal{P} = \{P_0, P_1\}$ of $\mathcal{A}$, with $P_0 = \{0\}$ and $P_1 = \mathcal{A} \setminus P_0$, is an $\mathcal{F}$-partition. The functions

$$x_k^*(v) = q\Big(\theta^{kv} - \delta(v)\Big), \qquad k \in \mathbb{Z}_q$$

have support in P_1. We have $\sum_{k \in \mathbb{Z}_q} x_k^*(v) = 0$, so the dimension of the space spanned by the functions $x_k^*(v)$ is $n = q - 1$. The inverse Fourier transforms are

$$x_k(u) = q\delta(u-k) - 1$$

We recognize the resulting spherical code as the simplex code with parameters $(n, \rho, M) = (q-1, 2q/(q-1), q)$. □

Example 2. Take $\mathcal{A} = (\mathbb{Z}_q \times \mathbb{Z}_2, +)$, and consider the following $\mathcal{F}$-partition $\mathcal{P} = \{P_0, P_1, P_2, P_3\}$ given by $P_0 = \{(u,v) = (0,0)\}$, $P_1 = \{(u,v) : u \neq 0; v = 1\}$, $P_2 = \{(u,v) : u \neq 0; v = 0\}$, and $P_3 = \{(u,v) = (0,1)\}$. Define

$$z_k^*(u,v) \;=\; \begin{cases} 2x_k^*(u), & v = 1 \\ 0, & v = 0 \end{cases} \qquad k \in \mathbb{Z}_q$$

where the functions x_k^* are as in Example 1. The functions z_k^* have support in P_1. Computing the inverse Fourier transform we get

$$z_k(u,v) \;=\; (-1)^v x_k(u), \qquad \text{for } k = 1, 2, \ldots, n$$

The resulting spherical code is a union of two antipodal simplex codes. The parameters of this code are $(n, \rho, M) = (q-1, 2(q-2)/(q-1), 2q)$. □

Example 3. Let $\mathcal{A} = (\mathbb{F}_2^n, +)$ be the additive group in the linear space of all n-length sequences of symbols from the binary field $\mathbb{F}_2$. Define

$$P_i \;=\; \Big\{u \in \mathbb{F}_2^n : w_H(u) = i\Big\}, \qquad \text{for } i = 0, 1, \ldots, n$$

It is well known that this partition generates a self-dual association scheme. Let $x_i^*(v) = 2^n/\sqrt{n}$ if $v = (v_1, v_2, \ldots, v_n)$ is zero in all but the i-th position, and let $x_i^*(v) = 0$ otherwise. Clearly, the functions x_i^* have support in P_1. Thus the inverse Fourier transforms are $x_i(u) = (-1)^{u_i}/\sqrt{n}$ for $i = 1, 2, \ldots, n$, where $u = (u_1, u_2, \ldots, u_n)$. The corresponding spherical code is a hypercube with parameters $(n, \rho, M) = (n, 4/n, 2^n)$. □

References

[1] R.E. BLAHUT, *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley 1983.

[2] E. BANNAI AND T. ITO, *Algebraic Combinatorics I: Association Schemes*, New York: Benjamin 1984.

[3] P. CAMION, Codes and association schemes, in *Handbook of Coding Theory*, V.S. Pless and W.C. Huffman (Editors), Amsterdam: Elsevier 1998.

[4] PH. DELSARTE, An algebraic approach to the association schemes of coding theory, *Philips Res. Rept. Suppl.*, vol. 10, 1973.

[5] T. ERICSON, Distance regular spherical codes, in *Proc. 35-th Allerton Conf. Comm., Control, and Computing*, Monticello, IL., pp. 413–421, October 1997.

[6] T. ERICSON, J. SIMONIS, H. TARNANEN, AND V. ZINOVIEV, $\mathcal{F}$-partitions of cyclic groups, *Appl. Algebra Engrg. Comm. Comput.* vol. 8, pp. 387–393, 1997.

[7] ———, On abelian schemes, unpublished manuscript, 1998.

[8] T. ERICSON AND V. ZINOVIEV, On Fourier-invariant partitions of finite abelian groups and the MacWilliams identity for group codes, *Problemy Peredachi Informatsii*, vol. 32, pp. 137–143, 1996.

[9] C.D. GODSIL, *Algebraic Combinatorics*, New York: Chapman and Hall 1993.

Codes and Ciphers: Fourier and Blahut

James L. Massey

SIGNAL AND INFORMATION PROCESSING LABORATORY
SWISS FEDERAL INSTITUTE OF TECHNOLOGY, ETH-ZENTRUM
CH-8092 ZÜRICH, SWITZERLAND
101767.233@compuserve.com

ABSTRACT. A self-contained treatment of the discrete Fourier transform (DFT) and its generalization over an arbitrary field is given, culminating in Blahut's theorem, which relates the DFT to linear complexity. Some applications of the DFT in coding and in cryptography are described. The DFT over general commutative rings is introduced and the condition for its existence given. Blahut's Theorem is shown to hold unchanged in general commutative rings.

1. Introduction

Our intention in this paper is to provide a self-contained treatment of the discrete Fourier transform (DFT) as applied in coding and cryptography. For this purpose, we give in § 2 a review of the standard DFT over an arbitrary field. § 3 surveys the most important properties of linear complexity, culminating in Blahut's Theorem, which serves as the crucial link between the DFT and its applications in coding and cryptography, and relates the DFT to linear complexity.

§ 4 gives some of the principal applications of the DFT and Blahut's Theorem in coding. In § 5, a generalization of the DFT is given for the case of a field of prime characteristic p in which no DFT of the desired length N can be formulated. This culminates in the Guñther-Blahut Theorem, which is the natural generalization of Blahut's Theorem to this case. § 6 gives some of the principal applications of the DFT and its generalization in cryptography.

In § 7, a novel extension of the DFT to commutative rings is given. The corresponding generalization of Blahut's Theorem is given in § 8.

2. The usual DFT

The theory of the discrete Fourier transform is closely tied to the notion of roots of unity. An element ξ in a field $\mathbb{F}$ is said to be a *primitive N-th root of unity*, where N is a positive integer, if ξ has multiplicative order N; that is, if

$\xi^N = 1$ but $\xi^i \neq 1$ for $1 \leq i < N$. If ξ is a primitive N-th root of unity and j is any integer, then $\xi^j = \xi^{j \bmod N}$, where here and hereafter $j \bmod N$ denotes the remainder when j is divided by N. In particular, $\xi^j = 1$ if and only if j is divisible by N. If ξ is a primitive N-th root of unity, then so is ξ^{-1}, and conversely, as ξ and ξ^{-1} must have the same multiplicative order.

The following symmetry property of roots of unity is the primary reason for their usefulness in the DFT.

Lemma 1. *If ξ is a primitive N-th root of unity in a field $\mathbb{F}$ and j is any integer, then*

$$\sum_{i=0}^{N-1} \xi^{ij} = \begin{cases} N, & \text{if } j \bmod N = 0 \\ 0, & \text{otherwise} \end{cases}$$

where the N on the right of the equality denotes the sum of N ones in $\mathbb{F}$.

Proof. If $j \bmod N = 0$, then $ij \bmod N = 0$ as well. Thus $\xi^{ij} = 1$ and hence the sum on the left is indeed $1 + 1 + \cdots + 1$ (N times). Otherwise, we note that

$$(1 - \xi^j) \sum_{i=0}^{N-1} \xi^{ij} = 1 - \xi^{Nj} = 0$$

But $\xi^j \neq 1$ because N does not divide j. Hence $1 - \xi^j \neq 0$ so that the sum on the left must vanish. ∎

If ξ is a primitive N-th root of unity in a field $\mathbb{F}$, then ξ^i is a zero of the polynomial $1 - D^N$. Thus $\{1, \xi, \xi^2, \ldots, \xi^{N-1}\}$ is the full set of zeroes of $1 - D^N$, as is also $\{1, \xi^{-1}, \xi^{-2}, \ldots, \xi^{-(N-1)}\}$, which is the same set. It follows that

$$1 - D^N = \prod_{j=0}^{N-1} (1 - \xi^j D)$$

Noting that $1 - D^N = 1 - (\xi^i D)^N$, one sees upon dividing by $1 - \xi^i D$ that

$$1 + \xi^i D + (\xi^i D)^2 + \cdots + (\xi^i D)^{N-1} = \prod_{\substack{j=0 \\ j \neq i}}^{N-1} (1 - \xi^j D)$$

Now setting $D = \xi^{-i}$ gives

$$N = \prod_{\substack{j=0 \\ j \neq i}}^{N-1} (1 - \xi^{j-i}) \neq 0$$

which establishes the following result.

Lemma 2. *If ξ is a primitive N-th root of unity in a field $\mathbb{F}$, then the reciprocal of N, the sum of N ones in $\mathbb{F}$, is well defined, i.e., this sum must be nonzero.*

The above proof of this lemma, which avoids the use of the formal derivative, is due to my frequent collaborator Shirlei Serconek.

Until further notice, ξ will denote a primitive N-th root of unity in a field $\mathbb{F}$. Let $\mathbb{F}^N$ denote the vector space of N-tuples (or "row vectors" of length N) with components in $\mathbb{F}$. The "usual" *discrete Fourier transform (DFT)* of length N generated by ξ is the mapping $\mathrm{DFT}_\xi(\cdot)$ from $\mathbb{F}^N$ to $\mathbb{F}^N$ defined by $\mathbf{B} = \mathrm{DFT}_\xi(\mathbf{b})$, where $\mathbf{b} = (b_0, b_1, \ldots, b_{N-1})$ and $\mathbf{B} = (B_0, B_1, \ldots, B_{N-1})$, in the manner

$$B_i = \sum_{n=0}^{N-1} b_n \xi^{in}, \qquad \text{for } i = 0, 1, \ldots, N \tag{1}$$

In communication engineering studies, the N-tuple $\mathbf{b}$ is called the *time-domain* sequence while the N-tuple $\mathbf{B}$ is called the *frequency-domain* sequence.

The DFT is always a true "transform," which means that it is always invertible. The inverse transform is given by

$$b_n = \frac{1}{N} \sum_{i=0}^{N-1} B_i \xi^{-in}, \qquad \text{for } n = 0, 1, \ldots, N \tag{2}$$

where $\frac{1}{N}$ denotes the reciprocal, whose existence is guaranteed by Lemma 2, of the sum of N ones in the field $\mathbb{F}$. This *inverse DFT formula* follows simply by noting that, for $0 \leq n < N-1$, we have

$$\sum_{i=0}^{N-1} B_i \xi^{-in} = \sum_{i=0}^{N-1} \sum_{m=0}^{N-1} b_m \xi^{im} = \sum_{m=0}^{N-1} b_m \sum_{i=0}^{N-1} \xi^{i(m-n)} = \begin{cases} Nb_n & \text{if } m = n \\ 0 & \text{otherwise} \end{cases}$$

as follows immediately from Lemma 1. By comparing the inverse DFT formula to the direct DFT formula, one sees that

$$\mathbf{b} = \frac{1}{N} \mathrm{DFT}_{\xi^{-1}}(\mathbf{B})$$

Thus the inverse DFT, except for the trivial multiplication by $\frac{1}{N}$, is just the DFT defined using ξ^{-1} as the underlying primitive N-th root of unity. There is no essential difference between the DFT and the inverse DFT.

The DFT equation can be written in matrix form as

$$\mathbf{B} = \mathbf{b}\, \mathrm{M}_\xi \tag{3}$$

where:

$$\mathrm{M}_\xi = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \xi & \xi^2 & \cdots & \xi^{N-1} \\ 1 & \xi^2 & \xi^{2\cdot 2} & \cdots & \xi^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \xi^{N-1} & \xi^{(N-1)2} & \cdots & \xi^{(N-1)(N-1)} \end{bmatrix} \tag{4}$$

It is often useful to identify the vectors $\mathbf{b}$ and $\mathbf{B}$ with the polynomials

$$b(X) = b_0 + b_1 X + \cdots + b_{N-1} X^{N-1} \tag{5}$$

and

$$B(D) = B_0 + B_1 D + \cdots + B_{N-1} D^{N-1} \tag{6}$$

respectively. In this notation, the DFT and inverse DFT formulas have their simplest forms, namely:

$$B_i = b(\xi^i), \qquad \text{for } i = 0, 1, \ldots, N \tag{7}$$

and

$$b_n = \frac{1}{N} B(\xi^{-n}), \qquad \text{for } n = 0, 1, \ldots, N \tag{8}$$

respectively.

3. Linear complexity and Blahut's theorem

What makes the DFT useful in coding and cryptography is its relation to the linear complexity of sequences. The *linear complexity* $\mathcal{L}(\tilde{\mathbf{s}})$ of the finite or semi-infinite $\mathbb{F}$-ary sequence $\tilde{\mathbf{s}} = s_0, s_1, s_2, \ldots$, where $s_i \in \mathbb{F}$, is the smallest integer $L \geq 0$ for which there exist coefficients $c_1, c_2, \cdots, c_L$ in the field $\mathbb{F}$ such that

$$s_j + c_1 s_{j-1} + \cdots + c_L s_{j-L} = 0 \qquad \text{for all} \quad j \geq L$$

Equivalently, $\mathcal{L}(\tilde{\mathbf{s}})$ is the smallest nonnegative integer L such that there exists a polynomial

$$C(D) = 1 + c_1 D + c_2 D^2 + \cdots + c_L D^L$$

with coefficients in $\mathbb{F}$ such that

$$S(D)C(D) = \mathcal{P}(D) \tag{9}$$

where $S(D)$ is the *formal power series* $S(D) = s_0 + s_1 D + s_2 D^2 + \cdots$ (which is a polynomial if the sequence is finite) and where $\mathcal{P}(D)$ is a polynomial of degree strictly less than L.

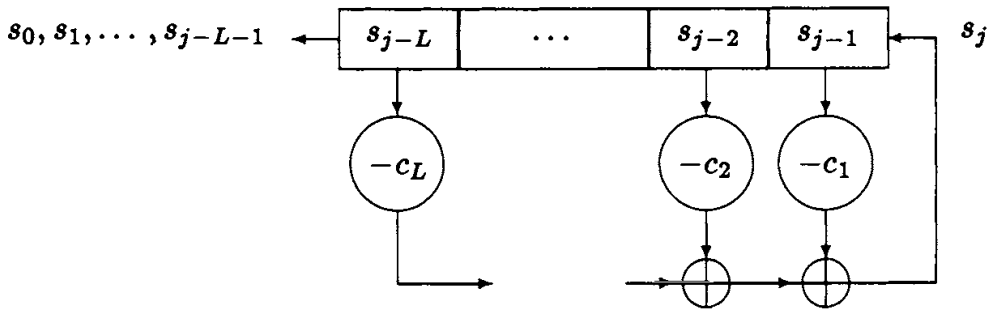


Figure 1. *A linear feedback shift register (LFSR)*

Again equivalently, $\mathcal{L}(\tilde{\mathbf{s}})$ is the length L of the shortest linear feedback shift-register (LFSR) as shown in Figure 1 that can generate $\tilde{\mathbf{s}}$ when the first L digits of $\tilde{\mathbf{s}}$ are initially loaded in the register. The polynomial $C(D)$ is called the *connection polynomial* of the LFSR. This shift register viewpoint of linear complexity is perhaps the one most often adopted in engineering studies (cf. [9]).

The semi-infinite sequence $\tilde{\mathbf{s}}$ will be called *N-periodic* if $s_i = s_{i+N}$ for all $i \geq 0$. Note that the smallest N for which $\tilde{\mathbf{s}}$ is N-periodic is the true *period* of the sequence. If $\tilde{\mathbf{s}}$ is N-periodic, then its formal power series $S(D)$ can be written

$$S(D) = s_N(D)(1 + D^N + D^{2N} + \cdots)$$

where $s_N(D)$ is the polynomial $s_N(D) = s_0 + s_1 D + s_2 D^2 + \cdots + s_{N-1} D^{N-1}$ of degree less than N determined by the first N digits of $\tilde{\mathbf{s}}$. Hence

$$S(D)(1 - D^N) = s_N(D)$$

Multiplying by $C(D)$ and using (9) gives

$$\mathcal{P}(D)(1 - D^N) = s_N(D)C(D)$$

which ensures that $\deg \mathcal{P}(D) < \deg C(D)$. If $\mathcal{P}(D)$ and $C(D)$ are relatively prime, that is if $\gcd(\mathcal{P}(D), C(D)) = 1$, then their degrees must be as small as possible so that L is the linear complexity of $\tilde{\mathbf{s}}$. We have proved the following theorem, which was given in [**12**].

Theorem 3. *A polynomial $C(D) = 1 + c_1 D + c_2 D^2 + \cdots + c_L D^L$ of degree L with coefficients in the field $\mathbb{F}$ is the connection polynomial of the shortest LFSR that generates the N-periodic $\mathbb{F}$-ary sequence $\tilde{\mathbf{s}} = s_0, s_1, s_2, \ldots$ if and only if*

$$s_N(D)C(D) = \mathcal{P}(D)(1 - D^N) \tag{10}$$

where $s_N(D) = s_0 + s_1 D + s_2 D^2 + \cdots + s_{N-1} D^{N-1}$, and $\mathcal{P}(D)$ is a polynomial satisfying $\gcd(\mathcal{P}(D), C(D)) = 1$. Moreover, L is the linear complexity of $\tilde{\mathbf{s}}$.

Recalling that the zeroes of $1 - D^N$ are just the elements ξ^{-n} for $n = 0, 1, \ldots, N$, we see from (10) and from $\gcd(\mathcal{P}(D), C(D)) = 1$ that ξ^{-n} is a zero of either $C(D)$ or $s_N(D)$, *but not both* for $n = 0, 1, \ldots, N$, and that $C(D)$ has no other zeroes. It follows that the degree of $C(D)$ is the number of ξ^{-n} for $0 \leq n < N$ that are **not** zeroes of $s_N(D)$. But $s_N(\xi^{-n})$ according to (8) is just a component of the inverse DFT $\mathbf{b}$ of the vector $\mathbf{B} = (s_0, s_1, \ldots, s_{N-1})$, up to scaling by the non-zero constant $\frac{1}{N}$. This proves the following result of exceptional importance for coding and cryptographic applications of the DFT.

Theorem 4 (Blahut's Theorem). *If $\mathbf{B} = \mathrm{DFT}_\xi(\mathbf{b})$, where the DFT is in any field $\mathbb{F}$, then the linear complexity of the semi-infinite sequence*

$$\tilde{\mathbf{s}} = \mathbf{B}, \mathbf{B}, \mathbf{B}, \ldots$$

obtained by periodic repetition of the frequency-domain vector $\mathbf{B}$ is equal to the Hamming weight $w_H(\mathbf{b})$ of the time-domain vector $\mathbf{b}$. Similarly, the linear complexity of the sequence obtained by periodic repetition of the time-domain vector is equal to the Hamming weight of the frequency-domain vector.

As we stated in our review [**10**] of [**3**] some thirteen years ago, we have since the appearance in 1979 of [**2**] consistently referred to the above result as "Blahut's Theorem" although it was not to be found explicitly in Blahut's work. However, its content so pervasively underlies Blahut's pioneering paper [**2**] applying

the DFT to cyclic codes that we believe the attribution to Blahut to be fully deserved. The above proof of Blahut's Theorem is taken from [12].

4. Coding applications of the DFT

Coding applications of the DFT rely on the polynomial formulation (7)-(8) of the DFT. Suppose now that ξ is a primitive N-th root of unity in a finite field $\mathbb{F}$. A cyclic code of length N over $\mathbb{F}$, or over a subfield of $\mathbb{F}$, with generator polynomial $g(X)$, where $g(X)$ is a monic polynomial that divides $X^N - 1$, is the set of all N-tuples $\mathbf{b}$ with components in $\mathbb{F}$ (or in the specified subfield of $\mathbb{F}$) such that $g(X)$ divides $b(X)$. But the zeroes of $X^N - 1$ are just the elements ξ^i for $0 \leq i < N$. Thus $g(X)$ is completely characterized by those i for which ξ^i is a zero, say $i \in \mathcal{J}$. Moreover, $g(X)$ divides $b(X)$ if and only if ξ^i is also a zero of $b(X)$ for $i \in \mathcal{J}$. But $b(\xi^i) = B_i$ according to (7), and hence $\mathbf{b}$ is a codeword if and only if the components B_i of its DFT vanish for $i \in \mathcal{J}$.

The first to use the DFT explicitly by name to describe cyclic codes appears to have been Gore [5]. But Blahut [2] was certainly the first to appreciate fully the power of this transform approach to the study of cyclic codes–it placed the many techniques developed by signal processing researchers for the DFT directly at the service of the coding theorist. In particular, fast transform techniques can be used to simplify the encoding and decoding of cyclic codes.

When a sequence is generated by an LFSR of length L as shown in Figure 1, then every L consecutive digits of the sequence form the "state" of the LFSR that is used to compute the next digit in the sequence by linear combination of its components. If the state is all-zero, then all future digits in the sequence must also be all zeroes. Thus, if a sequence contains a run of $d-1$ consecutive zeroes followed by a non-zero digit, then it must have linear complexity at least d. It follows that if $g(X)$ has $d-1$ consecutive powers of ξ as zeroes, then every non-zero codeword in the corresponding cyclic code has Hamming weight at least d, and hence the minimum distance of the code satisfies $d_{\min} \geq d$. This is the well known *BCH bound* (cf. [3, p. 216]). In fact, *all* of the known lower bounds on the minimum distance of cyclic codes can be derived in this manner, namely by deriving lower bounds on the linear complexity of a sequence obtained by periodically repeating a non-zero vector, based on the pattern of known zeros in the vector [16]. See also [11] for some simple examples.

5. The generalized DFT and the Günther-Blahut theorem

Let $\mathbb{F}$ be a field of prime characteristic p so that $N = np^\nu$ where $\gcd(n, p) = 1$. Then

$$1 - D^N = (1 - D^n)^{p^\nu} \tag{11}$$

Thus the zeroes of $1 - D^N$ cannot be distinct when $\nu \neq 0$, and hence there can be no primitive N-th root of unity in $\mathbb{F}$. But there does exist a primitive n-th

root of unity ξ in $\mathbb{F}$, or in some extension field $\mathbb{E}$ of $\mathbb{F}$, so that ξ^i is a zero of $1-D^N$ with multiplicity p^ν for $0 \le i < n$. Nonetheless, one can set up a "DFT" of length N with the help of the venerable Hasse derivative [7], sometimes also called the *hyperderivative*, which we now briefly review. The j-th *Hasse derivative* of the polynomial

$$a(D) = \sum_i a_i D^i$$

is defined as

$$a^{[j]}(D) = \sum_i \binom{i}{j} a_i D^{i-j}$$

The more familiar j-th *formal derivative* of $a(D)$ is defined as:

$$a^{(j)}(D) = \sum_i i(i-1)\cdots(i-j+1)\, a_i D^{i-j} = j! \sum_i \binom{i}{j} a_i D^{i-j}$$

It follows that

$$a^{(j)}(D) = (j!)a^{[j]}(D)$$

which shows why the formal derivative is of limited usefulness in fields of prime characteristic p, since $j!$ and hence also the j-th formal derivative must vanish for all $j \ge p$. Note, however, that it is always true that $a^{(1)}(D) = a^{[1]}(D)$. That the Hasse derivative is the right derivative to use in fields of prime characteristic p is evident from the following theorem due to Hasse [7].

Theorem 5. *If $h(D)$ is irreducible in $\mathbb{F}[D]$ with $h^{(1)}(D) \neq 0$ and if m is any positive integer, then $[h(D)]^m$ divides $a(D)$ if and only if $h(D)$ divides $a(D)$ as well as its first $m-1$ Hasse derivatives.*

We remark that if $h(D)$ is irreducible in the ring of polynomials $\mathbb{F}[D]$ with coefficients in $\mathbb{F}$, where $\mathbb{F}$ is a finite field or a field of characteristic 0, then it is always true that $h^{(1)}(D) \neq 0$. But this property does not hold in general in infinite fields of prime characteristic p. This is illustrated by the degree-two polynomial $h(D) = 1 + \frac{1}{X}D^2$ with coefficients in the field $\mathbb{F}_2(X)$ of rational functions with coefficients in the binary field $\mathbb{F}_2$. There is no rational function in $\mathbb{F}_2(X)$ whose square is X so that $h(D) = 1 + \frac{1}{X}D^2$ has no zero in $\mathbb{F}_2(X)$ and hence no polynomial divisor of degree one. It follows that $h(D)$ is irreducible, yet its first formal derivative vanishes.

Several authors [**14, 1, 6**] have proposed generalizations of the DFT that permit its application to N-tuples with $\gcd(N,p) \neq 1$. We use here the generalization that was proposed by Massey and Serconek [**12**], which is based closely on Gu̇ther's generalization [**6**] but is somewhat simpler. The definition is a natural generalization of the polynomial formulation (7)-(8) of the usual DFT.

In the remainder of this section, ξ denotes a primitive n-th root of unity in a finite field $\mathbb{F}$ of prime characteristic p. The *generalized discrete Fourier trans-*

form (GDFT) of the $\mathbb{F}$-ary vector $\mathbf{s}_N = (s_0, s_1, \cdots, s_{N-1})$, where $N = np^\nu$ and $\gcd(n,p) = 1$, is the $p^\nu \times n$ $\mathbb{F}$-ary matrix

$$\mathbf{S}_{p^\nu \times n} = \begin{bmatrix} s_N(1) & s_N(\xi) & \cdots & s_N(\xi^{n-1}) \\ s_N^{[1]}(1) & s_N^{[1]}(\xi) & \cdots & s_N^{[1]}(\xi^{n-1}) \\ \vdots & \vdots & & \vdots \\ s_N^{[p^\nu-1]}(1) & s_N^{[p^\nu-1]}(\xi) & \cdots & s_N^{[p^\nu-1]}(\xi^{n-1}) \end{bmatrix}$$

When $\nu = 0$, then the GDFT reduces to the usual DFT. We refer the reader to [**12**] for the proof that the GDFT is invertible, as is always demanded of a transform, and for other basic properties of the GDFT.

Example. Consider the binary 12-periodic sequence $\tilde{\mathbf{s}}$, that is a sequence over $\mathbb{F} = \mathrm{GF}(2)$, with $\mathbf{s}_{12} = (0,0,1,0,1,0,1,0,0,0,0,0)$. Then $N = 12$ and $p = 2$, which gives $n = 3$ and $\nu = 2$. Taking ξ as a primitive third root of unity in an extension of GF(2) requires that $\xi^2 = \xi + 1$. The sequence $\mathbf{s}_{12}$ corresponds to the polynomial $s_{12}(D) = D^2 + D^4 + D^6$. Taking Hasse derivatives gives

$$s_{12}^{[1]}(D) = 0, \qquad s_{12}^{[2]}(D) = 1 + D^4, \qquad s_{12}^{[3]}(D) = 0$$

Direct substitution in these polynomials gives $s_{12}(1) = 1$, $s_{12}(\xi) = s_{12}(\xi^2) = 0$, $s_{12}^{[1]}(1) = s_{12}^{[1]}(\xi) = s_{12}^{[1]}(\xi^2) = 0$, while

$$s_{12}^{[2]}(1) = 0 \qquad s_{12}^{[2]}(\xi) = 1 + \xi \qquad s_{12}^{[2]}(\xi^2) = \xi$$

and $s_{12}^{[3]}(1) = s_{12}^{[3]}(\xi) = s_{12}^{[3]}(\xi^2) = 0$. It follows that the GDFT of the binary vector $\mathbf{s}_{12} = (0,0,1,0,1,0,1,0,0,0,0,0)$ is the matrix:

$$\mathbf{S}_{4\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1+\xi & \xi \\ 0 & 0 & 0 \end{bmatrix}$$

□

The *Günther weight*, defined in [**12**], of a matrix is the number of its entries that are non-zero or that lie below a non-zero entry. For example, the matrix $\mathbf{S}_{4\times 3}$ above has Günther weight 8. Note that Guüther weight reduces to Hamming weight when the matrix has a single row.

We can now state the main result relating the GDFT to linear complexity. This theorem appears in [**12**], where its essential content is credited to Guüther [**6**].

Theorem 6 (Günther-Blahut Theorem). *The linear complexity of the $\mathbb{F}$-ary N-periodic sequence $\tilde{\mathbf{s}}$, where $\mathbb{F}$ is a finite field of characteristic p and where $N = np^\nu$ with $\gcd(n,p) = 1$, is equal to the Günther weight of the GDFT matrix $\mathbf{S}_{p^\nu \times n}$ of $\mathbf{s}_N = (s_0, s_1, \cdots, s_{N-1})$.*

Proof. Combining (10) and (11) gives $s_N(D)C(D) = \mathcal{P}(D)(1 - D^n)^{p^\nu}$. But ξ^i is a zero of $(1 - D^n)^{p^\nu}$ with multiplicity p^ν for $i = 0, 1, \ldots, n$. It thus follows from Theorem 3 that, for every $i = 0, 1, \ldots, n$ the sum of the multiplicities of ξ^i as a zero of $s_N(D)$ and as a zero of $C(D)$ must be exactly p^ν, unless ξ^i is a zero

of $s_N(D)$ with multiplicity p^ν or greater, in which case it cannot also be a zero of $C(D)$. But the multiplicity of ξ^i as a zero of $s_N(D)$, according to Theorem 5, is p^ν or greater if $s_N^{[j]}(\xi^i) = 0$ for $j = 0, 1, \ldots, n$. Otherwise, this multiplicity is given by $\min\{j : 0 \le j < n \text{ and } s_N^{[j]}(\xi^i) \neq 0\}$. Hence the multiplicity μ_i of ξ^i as a zero of $C(D)$ is equal to 0 provided $s_N^{[j]}(\xi^i) = 0$ for $j = 0, 1, \ldots, n$ and is given by $p^\nu - \min\{j : 0 \le j < n \text{ and } s_N^{[j]}(\xi^i) \neq 0\}$ otherwise. But this is just the Gunther weight of the column of the GDFT $\mathbf{S}_{p^\nu \times n}$ whose uppermost entry is $s_N(\xi^i)$. It follows further from Theorem 3 that $C(D)$ has no other zeroes and hence that the linear complexity of the N-periodic sequence $\tilde{\mathbf{s}}$ is $\mu_0 + \mu_1 + \cdots + \mu_{n-1}$, which is just the Gunther weight of the GDFT $\mathbf{S}_{p^\nu \times n}$. ∎

6. Cryptographic applications of the DFT

Linear complexity plays a very important role in the theory of stream ciphers, so it is not surprising that the DFT has been applied to such problems [**13**], as also has been the GDFT [**12**]. More surprisingly perhaps has been the application of the multidimensional DFT over the real numbers with $\xi = -1$, and thus length $N = 2$ in each dimension (the so-called *Walsh-Hadamard transform*), to the analysis of a sequence produced by a nonlinear combination of sequences, which we now review.

Siegenthaler [**17**] defined a boolean function f of ν binary variables to be m-th *order correlation immune* if the following condition holds: when the inputs are independent balanced binary random variables, the output is independent of every set of m or fewer of the inputs. The idea is that this function would be used to "combine" ν pseudorandom sequences. If the function is m-th order correlation immune, then an attacker who observes some portion of the resulting sequence would have to consider the input pseudorandom sequences at least $m+1$ at a time to acquire useful information and thus would be prevented from making a "divide and conquer" attack on the individual input pseudorandom sequences or on a small number of these sequences.

The Walsh-Hadamard transform can be applied to this problem by treating both the value of the function $f(n_1, n_2, \ldots, n_\nu)$ and the values of its arguments as the real numbers 0 or 1. The ν arguments are the *time* indices for ν different time axes. With this understanding, the Walsh-Hadamard transform $\mathcal{F}$ of f is the function of ν *frequency* indices given by

$$\mathcal{F}(i_1, i_2, \ldots, i_\nu) = \sum_{n_1=0}^{1} \cdots \sum_{n_\nu=0}^{1} f(n_1, n_2, \ldots, n_\nu)(-1)^{i_1 n_1 + i_2 n_2 + \cdots + i_\nu n_\nu}$$

for all $(i_1, i_2, \ldots, i_\nu) \in \{0,1\}^\nu$. With the notation $\mathbf{n} = (n_1, n_2, \ldots, n_\nu)$ and $\mathbf{i} = (i_1, i_2, \ldots, i_\nu)$, this expression can be written more simply as

$$\mathcal{F}(\mathbf{i}) = \sum_{\mathbf{i} \in \{0,1\}^\nu} f(\mathbf{n})(-1)^{\mathbf{i} \cdot \mathbf{n}}$$

where $\cdot$ denotes the usual dot product (or scalar product) of real-valued vectors.

The following connection of the Walsh-Hadamard transform to independence of random variables was given in [18]. To simplify notation, we will hereafter use $\langle \mathbf{x}, \mathbf{i} \rangle$ to denote $(\mathbf{x} \cdot \mathbf{i}) \bmod 2$, the dot product of $\mathbf{x}$ and $\mathbf{i}$ taken modulo 2.

Lemma 7. *Let $\mathbf{X} = (X_1, X_2, \ldots, X_\nu)$ be a random vector whose components are independent binary coin-tossing random variables, with values treated as the real numbers 0 or 1, and let $\mathbf{i} \in \{0,1\}^\nu$ with $\mathbf{i} \neq \mathbf{0}$. Then $f(\mathbf{X})$ is independent of $\langle \mathbf{X}, \mathbf{i} \rangle$ if and only if $\mathcal{F}(\mathbf{i}) = 0$.*

Proof. Since $\mathbf{i} \neq \mathbf{0}$, it is clear that $\langle \mathbf{x}, \mathbf{i} \rangle = 0$ holds for exactly half of the 2^ν values of $\mathbf{x} \in \{0,1\}^\nu$. Thus one has

$$\Pr\Big\{f(\mathbf{X}) = 1 \mid \langle \mathbf{X}, \mathbf{i} \rangle = 0\Big\} - \Pr\Big\{f(\mathbf{X}) = 1 \mid \langle \mathbf{X}, \mathbf{i} \rangle = 1\Big\} =$$

$$\frac{1}{2^{\nu-1}} \left(\sum_{\mathbf{x}:\langle \mathbf{x},\mathbf{i}\rangle = 0} f(\mathbf{x}) - \sum_{\mathbf{x}:\langle \mathbf{x},\mathbf{i}\rangle = 1} f(\mathbf{x}) \right) = \frac{1}{2^{\nu-1}} \sum_{\mathbf{x} \in \{0,1\}^\nu} f(\mathbf{x})(-1)^{\mathbf{x}\cdot\mathbf{i}} = \frac{1}{2^{\nu-1}} \mathcal{F}(\mathbf{i})$$

It follows that the two conditional probabilities above are equal if and only if $\mathcal{F}(\mathbf{i}) = 0$, which proves the lemma. ∎

The above lemma was used in [18] to obtain the following spectral characterization of correlation immunity.

Theorem 8. *The boolean function $f(n_1, n_2, \ldots, n_\nu)$ is m-th order correlation immune if and only if its Walsh-Hadamard transform $\mathcal{F}(i_1, i_2, \ldots, i_\nu)$ vanishes at all ν-dimensional frequencies $\mathbf{i} = (i_1, i_2, \ldots, i_\nu)$ in $\{0,1\}^\nu$ having Hamming weight between 1 and m, inclusive.*

An elegant proof of this theorem, which relies essentially on the following fact, has been given by Brynielsson [4].

Lemma 9. *Suppose that $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$ is a random vector whose components are independent binary coin-tossing random variables and that $\mathcal{Z}$ is any discrete random variable. Then $\mathbf{Y}$ and $\mathcal{Z}$ are independent if and only if $\langle \mathbf{Y}, \mathbf{i} \rangle$ is independent of $\mathcal{Z}$ for all non-zero $\mathbf{i}$ in $\{0,1\}^m$.*

Proof. The necessity of the condition is obvious, so it suffices to prove sufficiency. Suppose, then, that $\langle \mathbf{Y}, \mathbf{i} \rangle$ is indeed independent of $\mathcal{Z}$ for all non-zero $\mathbf{i}$ in $\{0,1\}^m$. (Note that when $\mathbf{i} = \mathbf{0}$ then $\langle \mathbf{Y}, \mathbf{i} \rangle = 0$, which is trivially independent of $\mathcal{Z}$.) Thus the conditional expectation $\mathsf{E}\big[(-1)^{\langle \mathbf{Y},\mathbf{i}\rangle} \mid \mathcal{Z}{=}z\big]$ has the same value for every value z of $\mathcal{Z}$ with $\Pr\{\mathcal{Z}{=}z\} \neq 0$. Hence

$$\mathsf{E}\Big[(-1)^{\langle \mathbf{Y},\mathbf{i}\rangle} \;\Big|\; \mathcal{Z}{=}z\Big] = \mathsf{E}\Big[(-1)^{\mathbf{Y}\cdot\mathbf{i}} \;\Big|\; \mathcal{Z}{=}z\Big] = \sum_{\mathbf{y} \in \{0,1\}^m} (-1)^{\mathbf{y}\cdot\mathbf{i}} \Pr\{\mathbf{Y}{=}\mathbf{y} \mid \mathcal{Z}{=}z\}$$

which is just the Walsh-Hadamard transform of the conditional probability distribution $P\{\mathbf{Y}{=}\cdot \mid \mathcal{Z}{=}z\}$, also has the same value for such every value z of $\mathcal{Z}$.

Thus $P\{\mathbf{Y}=\cdot \mid \mathcal{Z}=z\}$ itself must be the same distribution for every such value z of $\mathcal{Z}$ so that $\mathbf{Y}$ and $\mathcal{Z}$ are indeed independent. ■

Theorem 8 follows directly from Lemmas 7 and 9, by choosing $Y_1, Y_2, \ldots, Y_m$ to be an arbitrary selection of m of the random variables $X_1, X_2, \ldots, X_\nu$ that are used as arguments of the function f in Theorem 8.

7. The DFT over commutative rings

We now investigate how some of the previous theory must be modified when the sequences have components in a commutative ring $\mathcal{R}$ instead of a field $\mathbb{F}$. In particular, we are interested in when the DFT is well defined in such a ring and whether Blahut's Theorem then holds. We remark that although earlier in algebra one sometimes considered rings without the multiplicative identity 1, the modern practice is to include the existence of the multiplicative identity 1 in the definition of a ring [**8**, p.86], and we will adhere to this later convention.

A *primitive N-th root of unity*, where N is a positive integer, is defined the same way in an arbitrary ring $\mathcal{R}$ as it is defined in a field $\mathbb{F}$. Thus ξ is a primitive N-th *root of unity* in $\mathcal{R}$ if ξ has multiplicative order N. This implies that ξ is a *unit* of $\mathcal{R}$, i.e., an element with a multiplicative inverse, and indeed $\xi^{-1} = \xi^{N-1}$.

Hereafter, $\mathcal{R}$ denotes an arbitrary commutative ring and ξ denotes a primitive N-th *root of unity* in $\mathcal{R}$. The *ring of integers modulo m*, which we denote by $\mathbb{Z}_m$, whose elements are $0, 1, \ldots, m-1$ and whose rules are addition and multiplication modulo m, is a finite ring of this type. We remind the reader that the element i of $\mathbb{Z}_m$ is a unit if and only if $\gcd(m, i) = 1$, where i is treated as an ordinary integer for computation of the greatest common divisor.

Among the peculiarities of the ring case are the facts that neither Lemma 1 nor Lemma 2 hold in general. For instance in $\mathbb{Z}_8$, the element $\xi = 3$ is a primitive second root of unity, but taking $j = 1$ in Lemma 1 gives $1+\xi = 4 \neq 0$; moreover, $N = 2$ is not a unit of $\mathbb{Z}_8$ because $\gcd(8, 2) \neq 1$. One cannot set up a "usual" DFT of length $N = 2$ over $\mathbb{Z}_8$, even though one can find a primitive second root of unity.

Hereafter, ξ denotes a primitive N-th root of unity in a commutative ring $\mathcal{R}$. We now derive the necessary and sufficient condition for ξ to determine a "usual" DFT of length N, i.e., one that satisfies (1) and (2), or equivalently, that satisfies (3) and (4), or again equivalently, that satisfies (7) and (8). The matrix-form DFT equation (3) is uniquely solvable for $\mathbf{b}$, when $\mathbf{B}$ is given, just when the determinant $\Delta(\mathrm{M}_\xi)$ of the matrix M_ξ in (4) is a **unit** of the ring $\mathcal{R}$. But M_ξ is a Vandermonde matrix [**3**, pp. 169–170], and hence its determinant is simply

$$\Delta(\mathrm{M}_\xi) = \prod_{j=1}^{N-1} \prod_{i=0}^{j-1} (\xi^j - \xi^i) = \prod_{j=1}^{N-1} \prod_{i=0}^{j-1} \xi^i (\xi^{j-i} - 1)$$

Because a product of ring elements is a unit if and only if each element is a unit and because ξ itself is a unit, it follows that $\Delta(\mathrm{M}_\xi)$ is a unit if and only if $\xi^k - 1$ is a unit for all $k = 1, 2, \ldots, N-1$.

Theorem 10 (DFT over rings). *If ξ is a primitive N-th root of unity in a commutative ring $\mathcal{R}$, then the usual DFT equation defines an invertible mapping from $\mathcal{R}^N$ to $\mathcal{R}^N$ whose inverse is given by the usual inverse DFT equation if and only if $\xi^k - 1$ is a unit of $\mathcal{R}$ for $k = 1, 2, \ldots, N-1$.*

Example. $\xi = 2$ is a primitive fourth root of unity in $\mathbb{Z}_{15}$ but does **not** generate a DFT of length $N = 4$ in this ring because, although $\xi - 1 = 1$ is a unit, $\xi^2 - 1 = 3$ is not a unit. □

Example. $\xi = 8$ is a primitive fourth root of unity in $\mathbb{Z}_{65}$ and **does** generate a DFT of length $N = 4$ because $\xi - 1 = 7$, $\xi^2 - 1 = 63$, and $\xi^3 - 1 = 57$ are all units in $\mathbb{Z}_{65}$. □

It is an easy matter with the help of Theorem 10 to prove the following theorem, which is well known in signal processing [**15**] (cf. *number theory transform*). However, it does not seem to have been realized previously that the simple necessary and sufficient condition given in Theorem 10 is all that must be checked.

Theorem 11. *Let $m = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$, where $p_1, p_2, \ldots, p_k$ are distinct primes and $e_1, e_2, \ldots, e_k$ are positive integers. Then there exists a DFT of length N over the ring $\mathbb{Z}_m$ if and only if N is a divisor of* $\gcd(p_1 - 1, p_2 - 1, \ldots, p_k - 1)$.

8. Blahut's theorem in commutative rings

Our last task is to determine whether Theorem 4 (Blahut's Theorem) applies for the DFT over commutative rings whenever such a DFT exists. We begin with a useful lemma.

Lemma 12. *Suppose that ξ generates a DFT of length N over a commutative ring $\mathcal{R}$; that is, suppose that ξ is a primitive N-th root of unity in $\mathcal{R}$ and that $\xi^k - 1$ is a unit of $\mathcal{R}$ for $k = 1, 2, \ldots N-1$. Then*

$$1 - D^N = \prod_{i=0}^{N-1} (1 - \xi^i D) \tag{12}$$

and, moreover,

$$N = \prod_{i=1}^{N-1} (1 - \xi^i) \tag{13}$$

Proof. Because $1 = \xi^0$ is a zero of $1 - D^N$, we have $1 - D^N = Q_0(D)(1 - D)$, for some polynomial $Q_0(D)$ with coefficients in $\mathcal{R}$. But ξ^{-1} is also a zero of $1 - D^N$ and hence

$$0 = Q_0(\xi^{-1})(1 - \xi^{-1})$$

By hypothesis, $1 - \xi^{-1}$ is a unit of $\mathcal{R}$ and hence ξ^{-1} must be a zero of $Q_0(D)$. Thus we have

$$1 - D^N = Q_1(D)(1 - \xi D)(1 - D)$$

for some polynomial $Q_1(D)$ with coefficients in $\mathcal{R}$. Iterating this argument gives (12). Dividing by $1 - D$ in (12) gives

$$\frac{1 - D^N}{1 - D} = 1 + D + \ldots + D^{N-1} = \prod_{i=1}^{N-1} (1 - \xi^i D)$$

Setting $D = 1$ in this equation yields (13). ■

Remark. The factorization (12) does not hold in general without the hypothesis that $\xi^k - 1$ is a unit of $\mathcal{R}$ for $k = 1, 2, \ldots N - 1$. For instance, taking $\xi = 3$ in $\mathbb{Z}_8$ so that $N = 2$, one finds

$$\prod_{i=0}^{N-1} (1 - \xi^i D) = (1 - D)(1 - 3D) = 1 - 4D + 3D^2 \neq 1 - D^2$$

◊

Suppose now that ξ defines a DFT of length N over $\mathcal{R}$ in accordance with Theorem 10. This guarantees that N in (13) is a unit of $\mathcal{R}$. We will be interested in the linear complexity of an arbitrary periodically repeated N-tuple $\mathbf{B} = (B_0, B_1, \ldots, B_{N-1})$ in $\mathcal{R}^N$. The power series of this n-periodic sequence is

$$\frac{B(D)}{1 - D^N}$$

where $B(D) = B_0 + B_1 D + \cdots + B_{N-1} D^{N-1}$. The factorization (12) suggests the partial-fraction expansion

$$\frac{B(D)}{1 - D^N} = \sum_{i=0}^{N-1} b_i \frac{1}{1 - \xi^i D} \tag{14}$$

That this is indeed valid follows from the fact that we can multiply on both sides by $1 - \xi^n D$, then set $D = \xi^{-n}$ and solve for b_n, precisely because the resulting denominator

$$\prod_{\substack{i=0 \\ i \neq n}}^{N-1} (1 - \xi^{i-n})$$

on the left side is guaranteed to be a unit. Carrying out this process, we find with the help of (13) that

$$b_n = \frac{1}{N} B(\xi^{-n})$$

which is just the usual inverse DFT formula (8). Thus we have shown that the power series of the N-periodic sequence can be written in terms of the N-tuple $\mathbf{b} = (b_0, b_1, \ldots, b_{N-1})$ as

$$\sum_{\substack{n=0 \\ b_n \neq 0}}^{N-1} b_n \frac{1}{1 - \xi^n D} = \frac{\mathcal{P}(D)}{\mathcal{C}(D)}$$

where

$$\mathcal{C}(D) = \prod_{\substack{n=0 \\ b_n \neq 0}}^{N-1} (1 - \xi^n D)$$

is a polynomial of degree $w_H(\mathbf{b})$ with coefficients in $\mathcal{R}$ and where the polynomial $\mathcal{P}(D)$ has degree strictly less than that of $\mathcal{C}(D)$. Moreover, the degree of $\mathcal{C}(D)$ cannot be reduced by some sort of cancellation with $\mathcal{P}(D)$ as else there would have been fewer than $w_H(\mathbf{b})$ terms in the partial fraction expansion (14). Equivalently, we have

$$\mathcal{P}(D)(1 - D^N) = B(D)\mathcal{C}(D)$$

It follows that $\mathcal{C}(D)$ is the connection polynomial of the shortest LFSR with feedback coefficients in $\mathcal{R}$ and that this shortest LFSR has length $w_H(\mathbf{b})$, which by definition is the linear complexity of the sequence.

We have thus proved that Blahut's Theorem does indeed hold in commutative rings in which the usual DFT can be defined.

Theorem 13 (Blahut's Theorem over rings). *If ξ generates a DFT of length N in a commutative ring $\mathcal{R}$ and $\mathbf{B}$ is the DFT of $\mathbf{b}$, then the linear complexity of the N-periodic sequence $\mathbf{B}, \mathbf{B}, \mathbf{B}, \ldots$ is equal to the Hamming weight of $\mathbf{b}$. Similarly, the linear complexity of the sequence obtained by periodic repetition of the time-domain vector $\mathbf{b}$ is equal to the Hamming weight of the frequency-domain vector $\mathbf{B}$.*

9. Concluding remarks

It has been our intention to illustrate the impact that the work of Richard E. Blahut has had on the fields of coding and cryptography. His role in bridging the fields of coding and signal processing has led to important advances in both fields.

Acknowledgment. We are pleased here to express our gratitude to Dr. Shirlei Serconek of the Universidade Federal de Goias, Brazil, to Prof. Ben Smeets of Lund University, Sweden, for helpful suggestions, and to Prof. Alexander Vardy of the University of Illinois for his careful editing of the manuscript.

References

[1] S.R. BLACKBURN, A generalisation of the discrete Fourier transform: Determining the minimal polynomial of a periodic sequence, *IEEE Trans. Inform. Theory*, vol. 40, pp. 1702–1704, November 1994.

[2] R.E. BLAHUT, Transform techniques for error control codes, *IBM J. Res. Dev.*, vol. 23, pp. 299–315, May 1979.

[3] ———, *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley 1983.

[4] L. BRYNIELSSON, A short proof of the Xiao-Massey lemma, *IEEE Trans. Inform. Theory*, vol. 35, p. 1344, November 1989.

[5] W.C. GORE, Transmitting binary symbols with Reed-Solomon codes, in *Proc. Princeton Conf. Information Sciences & Systems*, Princeton, NJ, pp. 495-497, March 1973.

[6] C.G. GÜNTHER, A finite field Fourier transform for vectors of arbitrary length, in *Communications and Cryptography: Two Sides of One Tapestry*, R.E. Blahut, D.J. Costello, Jr., U. Maurer, and T. Mittelholzer (Editors), pp. 141–153, Dordrecht: Kluwer 1994.

[7] H. HASSE, Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenen Konstantenkörper bei beliebiger Charakteristik, *J. Reine & Angewandte Math.*, vol. 175, pp. 50–54, 1936.

[8] N. JACOBSON, *Basic Algebra: Part I*, (Second Edition), New York: Freeman, 1985.

[9] J.L. MASSEY, Shift-register synthesis and BCH decoding, *IEEE Trans. Inform. Theory*, vol. 15, pp. 122–127, January 1969.

[10] ———, Review of [3], *IEEE Trans. Inform. Theory*, vol. 31, pp. 553–554, July 1985. Reprinted in *Proc. IEEE*, vol. 74, pp. 1293–1294, 1986.

[11] J.L. MASSEY AND T. SCHAUB, Linear complexity in coding theory, in *Coding Theory and Applications*, G.D. Cohen and Ph. Godlewski (Editors), *Lect. Notes Computer Science*, vol. 311, pp. 19–32, Berlin: Springer-Verlag 1988.

[12] J.L. MASSEY AND S. SERCONEK, Linear complexity of periodic sequences: A general theory, in *Advances in Cryptology*, N. Koblitz (Editor), *Lect. Notes Computer Science*, vol. 1109, pp. 358–371, Berlin: Springer-Verlag 1996.

[13] ———, A Fourier transform approach to the linear complexity of nonlinearly filtered sequences, in *Advances in Cryptology*, Y.G. Desmedt (Editor), *Lect. Notes Computer Science*, vol. 839, pp. 322–340, Berlin: Springer-Verlag 1994.

[14] P. MATHYS, A generalization of the discrete Fourier transform in finite fields, in *Proc. IEEE Symp. Inform. Theory*, San Diego, CA., pp. 14–19, January 1990.

[15] H.J. NUSSBAUMER, *Fast Fourier Transform and Convolution Algorithms*. Berlin: Springer-Verlag 1982.

[16] T. SCHAUB, *A Linear Complexity Approach to Cyclic Codes*, Ph.D. Dissertation, Swiss Federal Institute of Technology, Tech. Report ETH No. 8730, Zürich, Switzerland, 1988.

[17] T. SIEGENTHALER, Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Trans. Inform. Theory*, vol. 30, pp. 776–780, October 1984.

[18] G.Z. XIAO AND J.L. MASSEY, A spectral characterization of correlation-immune combining functions, *IEEE Trans. Inform. Theory*, vol. 34, pp. 569–571, May 1988.

Bounds on Constrained Codes in One and Two Dimensions

Jørn Justesen

DEPARTMENT OF TELECOMMUNICATION
TECHNICAL UNIVERSITY OF DENMARK
DK–2800 LYNGBY, DENMARK
jju@tele.dtu.dk

ABSTRACT. We consider a generalization of balanced ternary codes from one to two dimensions. The construction uses linear filtering of random fields, and we discuss upper bounds on the entropy based on linear prediction.

1. Introduction

The physical properties of communication links or storage media may require sequences of transmitted/stored symbols to satisfy certain constraints. Other constraints may be imposed in order to facilitate synchronization or other receiver functions. Typical constraints in one dimension include limits on run-lengths and balancing to suppress DC content. Some problems of maximizing the entropy of a code subject to the constraints are easily solved in one dimension by using properties of Markov sources, others are more subtle. In two dimensions, the theory of Markov random fields is not of much help, and we shall discuss a different approach. First we mention a one dimensional construction based on linear filtering and an upper bound based on linear least mean square are error prediction. Then we consider generalizations to two dimensions.

2. Constrained codes in one dimension

In most cases, the flexibility of finite state Markov sources allows us to construct codes that satisfy the given constraints, and probability distributions and other properties of interest may be calculated. If the structure of the source is derived from the constraint, as in the case of run-length limitations, the entropy may be maximized by varying the transition probabilities. However, if only the power spectrum is specified, it is not possible to completely specify the structure of the code, and there is no direct way of finding the maximal entropy. As an alternative to Markov sources we may consider linear filtering as a means of controlling the power spectrum. Signals generated by passing discrete sequences through filters with integer values of the impulse response are known as partial response signals in communications. We shall consider three examples in some detail.

Example 1. Assume that the alphabet is $\{0, 1, -1\}$ and the power spectrum is $\frac{1}{2}(1 - \cos\omega)$. The AMI code encodes binary information into a sequence satisfying these constraints by mapping logical '0' on 0 and alternating between 1 and -1 for representing '1'. An alternative encoder is a linear filter with ± 1 input and system function $G(\omega) = \frac{1}{2}(1 - e^{-i\omega})$. It is not easy to find a ternary code with the same power spectrum and entropy greater than 1 bit/symbol. □

In [1] we obtained upper bounds on the entropy of codes with given power spectra in the following way. From the given power spectrum we derive the linear minimum mean square error predictor, its variance, and the variance of the prediction error. The entropy is upper bounded as follows:

$$H(X) = H(X_t \mid X_{t-1}, X_{t-2}, X_{t-3}, \dots) \leq H(X_t X_t^*) \leq H(X_t) - R(e)$$

where X^* is the lmmse predictor and $R(e)$ is the rate-distortion function for X_t with the distortion e equal to the prediction error. In some cases, it is possible to restrict the set of values assumed by the predictor to a finite set.

Example 1. (continued) For the sequence discussed above, the predictor is

$$X_t^* = -(X_{t-1} + X_{t-2} + X_{t-3}, \dots) \qquad \text{or} \qquad X_t^* = X_{t-1}^* - X_{t-1}$$

The variance of the predictor and the prediction error are both $\frac{1}{4}$. If the predictor is properly initialized, the values are $\pm\frac{1}{2}$, and the prediction error of $\frac{1}{4}$ can be obtained only by letting the conditional probabilities for 0 and 1 be $\frac{1}{2}$ when the prediction equals $\frac{1}{2}$. Thus the conditional entropy is 1 bit. This is not the upper bound, however, since we only know that the set of values assumed by the predictor is $\{a, a \pm 1, a \pm 2, \dots\}$, but we have no information about a. If we take $a = 0$, there can be only three active symbols in the set, and we get the conditional entropy of 0.939 bit. In order to find the actual upper bound, we have to calculate the rate-distortion function for an asymmetric set of symbols and vary a to get the largest value. Blahut's algorithm [2] is an important tool in this process. For simplicity, we shall present an approximation which comes quite close to the true value. Let $a = \frac{1}{3}$, and take the active symbols to be $\frac{4}{3}$, $\frac{1}{3}$, and $-\frac{2}{3}$ (actually $-\frac{5}{3}$ might also have a nonzero probability). The probability distribution in this case is $(\frac{1}{72}, \frac{46}{72}, \frac{25}{72})$ and the conditional probabilities are:

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{17}{46} & \frac{28}{46} & \frac{1}{46} \\ 0 & \frac{8}{25} & \frac{17}{25} \end{bmatrix}$$

The conditional entropy computed from this distribution is 1.0083 bit. This value is very close to the true upper bound. □

3. Two dimensional codes

Digital codes in two dimensions are interesting for certain storage media. Although the discrete Markov random fields share some of the properties of finite state Markov sources, it is usually very difficult to calculate probability distri-

butions and entropies. In order to arrive at models that allow practical encoding and which can be analyzed with a reasonable effort, restricted classes of Markov fields have to be considered, or different approaches have to be found.

Example 2. We shall follow the approach based on linear filtering, which was used in Example 1. As the simplest possible case we shall use a separable filter which has the same function in the x and y direction

$$G(\omega) = {}^1\!/_2(1 - e^{-i\omega_x})(1 - e^{-i\omega_y})$$

The encoding consists of passing an array of ± 1 data through this filter. Thus the power spectrum becomes $(1-\cos\omega_x)(1-\cos\omega_y)$ and the correlation function is given by:

$$\begin{bmatrix} {}^1\!/_4 & -{}^1\!/_2 & {}^1\!/_4 \\ -{}^1\!/_2 & 1 & -{}^1\!/_2 \\ {}^1\!/_4 & -{}^1\!/_2 & {}^1\!/_4 \end{bmatrix}$$

The alphabet of the code is $\{0, \pm 1, \pm 2\}$ with probability distribution $({}^3\!/_8, {}^1\!/_4, {}^1\!/_4, {}^1\!/_{16}, {}^1\!/_{16})$. In each row or column, the sum of n consecutive symbols is between 2 and -2, and in any finite rectangle the sum of the symbols is also in this range.

By following the same approach as in the previous section, we note that the predictor variance becomes ${}^3\!/_4$ and the prediction error is ${}^1\!/_4$. For the code under consideration, the prediction values are $\pm{}^1\!/_2$, $\pm{}^3\!/_2$ with probability ${}^3\!/_8$ and ${}^1\!/_8$. The prediction error of ${}^1\!/_4$ is possible only if the source symbol has one of the two values closest to the prediction. Thus the conditional entropy is 1 bit. However if we let the reproducing alphabet consist of the same 5 integers as the source alphabet, we can get the right variances with probability distribution $({}^1\!/_{48}, {}^7\!/_{24}, {}^3\!/_8, {}^7\!/_{24}, {}^1\!/_{48})$ on the reproducing alphabet and transition probabilities:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ {}^1\!/_7 & {}^5\!/_7 & {}^1\!/_7 & 0 & 0 \\ 0 & {}^1\!/_9 & {}^7\!/_9 & {}^1\!/_9 & 0 \\ 0 & 0 & {}^1\!/_7 & {}^5\!/_7 & {}^1\!/_7 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In this case the conditional entropy becomes $H = 1.04$ bits. Thus the bound indicates the possibility of a symmetric code with the same power spectrum and slightly greater entropy. As in the previous section, the iterative algorithm may be used to find the actual upper bound. □

In the one dimensional case it is often possible to construct a model of the code as a Markov source based on the properties of the predictor. From this model, we may obtain an actual encoder by matching the input to the transition probabilities. However, in two dimensions, it is not obvious how the structure of the code can be derived from the predictor.

The linear filter has the disadvantage of increasing the size of the output alphabet. Thus it may be of interest to combine filtering and other methods in order to limit the alphabet or control the symbol distribution.

Example 3. In order to avoid the symbols ± 2 in Example 2, we could restrict the input data to avoid the particular configurations:

$$\begin{matrix} 1 & -1 \\ -1 & 1 \end{matrix} \qquad \begin{matrix} -1 & 1 \\ 1 & -1 \end{matrix}$$

This may be done with a Pickard field [**3**], and we shall briefly indicate the construction. We assume that the symbols are equally probable, and let:

$$\Pr[1\,1] = \tfrac{1}{2}(1-q), \qquad \Pr[-1\,1] = q/2$$

for some real value q. The particular property of a Pickard field is that the next symbol in the first column is independent of the symbol in the other column. We let $*$ stand for any symbol, and write $\bar{1}$ for -1. With this notation, we have:

$$\begin{aligned} \Pr[1\,1 \mid 1\,*] &= \tfrac{1}{2}(1-q)^2 \\ \Pr[1\,1 \mid \bar{1}\,*] &= \tfrac{1}{2}(1-q) \\ \Pr[1\,\bar{1} \mid \bar{1}\,*] &= \tfrac{1}{2}q^2 \end{aligned}$$

Assuming symmetry between the vertical and horizontal direction and setting the probabilities of the two unwanted configurations to 0, we can find the probabilities of the remaining 2×2 configurations as follows:

$$\begin{aligned} \Pr[1\,\bar{1} \mid \bar{1}\,\bar{1}] &= \tfrac{1}{2}q^2 \\ \Pr[1\,1 \mid 1\,1] &= \tfrac{1}{2}(1-2q) \\ \Pr[1\,1 \mid \bar{1}\,\bar{1}] &= \tfrac{1}{2}q - q^2 \end{aligned}$$

The entropy can be calculated from the probability distribution of the lower right symbol conditioned on the three other symbols:

$$H = (1-q)^2\,\mathcal{H}\left(\frac{q^2}{(1-q)^2}\right) + 2\frac{q}{1-q}\,\mathcal{H}\left(\frac{q}{1-q}\right)$$

Here $\mathcal{H}(\cdot)$ indicates the binary entropy function. Because of the independence assumption, all sequences of symbols where either the row or column index increases in each step are first order Markov chains. Thus the correlation function may be expressed as $R(j,k) = r^{|j|+|k|}$ where $r = 1-2q$. From this fact we can find the optimal linear predictor as follows:

$$y^*_{i,j} = r y_{i-1,j} + r y_{i,j-1} - r^2 y_{i-1,j-1}$$

The variance of the predictor is $2r^2 - r^4$ and the error variance is $1 - 2r^2 - r^4$. Obviously, the predictor assumes only eight distinct values, but we shall use a simpler version of the upper bound on the entropy letting the reproducing alphabet be $\pm\sqrt{2r^2 - r^4}$. In this way, we get the upper bound

$$E < \mathcal{H}\left(\frac{1}{2}\left(1 - \sqrt{2r^2 - r^4}\right)\right)$$

From the Pickard field we can get a balanced ternary field by applying the same filtering operation as in Example 2. Thus the correlation function can be found

as the convolution of the Pickard field correlation and the correlation found in Example 2. Similarly the power spectrum is obtained as the product

$$\mathcal{S}(\omega) \;=\; (1-r^2)^2 \;\frac{(1-\cos\omega_x)(1-\cos\omega_y)}{(1+r^2-2r\cos\omega_x)(1+r^2-2r\cos\omega_y)}$$

Clearly, the entropy is not changed since the mapping is one-to-one. For the same reason, the prediction error is just scaled by a factor $1/4$. The probability distribution of the symbols may be found from the probabilities of the binary 2×2 configurations, specifically $\Pr[1] = \Pr[-1] = 2q^2$. In this case we may find an upper bound by using the reproducing alphabet $\{0, \pm a\}$. If we make the assumption that there is no transition from a to -1 or from $-a$ to 1 and select the value of a, we can find $\Pr[a]$ to give the right variance of the prediction. Since the conditional probability of the source symbols must satisfy $\Pr[1|a] = a$ in order to give the right variance of the error, the remaining transition probabilities may be determined. The upper bound becomes

$$E \;<\; 2\Pr[a]\mathcal{H}(a) \;+\; \Pr[0]\left(\mathcal{H}\left(\frac{4q^2-2a\Pr[a]}{\Pr[0]}\right)+\frac{4q^2-2a\Pr[a]}{\Pr[0]}\right)$$

Letting $\Pr[1] = 1/4$, we get $q = 1/\sqrt{8}$ and $r = 0.2929$. Thus the entropy becomes 0.822. The upper bound for the Pickard field becomes 0.878. For the ternary field, the variance of the prediction error becomes $9/16 - \sqrt{2}/4$ and the prediction variance $\sqrt{2}/4 - 1/16$. Taking $a = 3/4$, we get

$$\Pr[a] \;=\; 1/18\left(4\sqrt{2}-1\right)$$

and the upper bound may be calculated as 0.909. The numbers may indicate that the upper bound is not as tight as in one dimension. However, some of the difference is certainly due to the fact that the Pickard field does not have the largest possible entropy. The difference between the two bounds may also indicate that there is some loss in using linear filtering for spectrum shaping.

4. Conclusions

Linear filtering may be a useful tool in constructing two dimensional codes. By this approach, it is possible to modify the power spectrum, while probability distributions and entropy can still be calculated. The rate of the code may be evaluated by comparing it to an upper bound on the entropy derived from linear prediction.

References

[1] R.E. BLAHUT, Computation of channel capacity and rate-distortion functions, *IEEE Trans. Inform. Theory*, vol. 18, pp. 460–473, 1972.

[2] J. JUSTESEN, Information rates and power spectra of digital codes, *IEEE Trans. Inform. Theory*, vol. 28, pp. 457–472, 1982.

[3] D.K. PICKARD, A curious binary lattice process, *J. Appl. Prob.*, vol. 14, pp. 717–731, 1977.

Classification of Certain Tail-Biting Generators for the Binary Golay Code

A. Robert Calderbank

AT&T Laboratories
Florham Park, NJ 07932, USA
rc@research.att.com

G. David Forney, Jr.

Motorola, Inc.
Mansfield, MA 02048, USA
LUSE27@email.mot.com

Alexander Vardy

Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801, USA
vardy@golay.csl.uiuc.edu

Abstract. We have recently found tail-biting trellis representations of the binary Golay code $\mathbb{C}_{24}$ that have considerably fewer states than the best conventional trellis. In particular, we have exhibited a 16-state, twelve-section tail-biting trellis for $\mathbb{C}_{24}$, whereas a conventional trellis must have at least 256 states at its midpoint. This trellis and the corresponding set of generators for $\mathbb{C}_{24}$ have an exceptionally regular structure. In this paper, we classify all the possible sets of generators of this type, up to automorphisms in the Mathieu group M_{24}. Our analysis is based on the Miracle Octad Generator representation of $\mathbb{C}_{24}$ and M_{24}.

1. Introduction

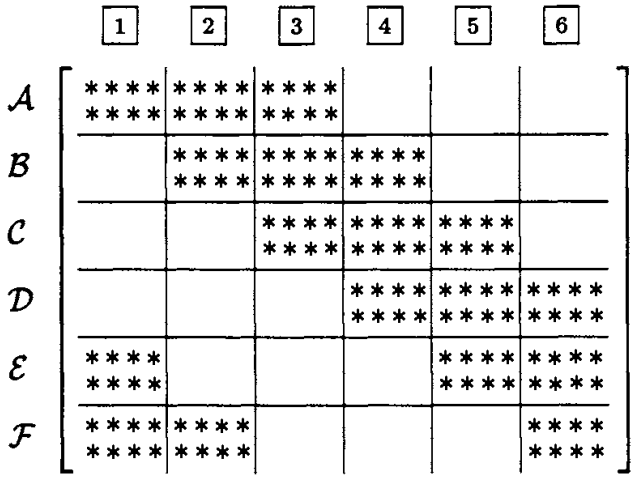
The $(24, 12, 8)$ Golay code $\mathbb{C}_{24}$ is the most extraordinary binary code. The codewords of any given weight form beautiful geometric configurations that continue to fascinate combinatorial mathematicians. The symmetry group of this code plays a central role in finite group theory, for it is the Mathieu group M_{24}, which is perhaps the most important of all the twenty-six sporadic simple groups.

We shall define $\mathbb{C}_{24}$ using the Miracle Octad Generator (MOG) and the hexacode $\mathcal{H}_6$, which is a $(6,3,4)$ code over the finite field with 4 elements. The MOG and the hexacode are computational tools that make calculation within the Golay code and its symmetry group M_{24} very easy.

A simple argument of Muder [**10**] shows that any trellis for the Golay code must have at least 256 states at its midpoint. Forney [**3**] shows this bound is met by the minimal trellis for $\mathbb{C}_{24}$, in the standard MOG coordinate ordering. This paper considers unconventional *tail-biting* representations of the Golay code, where the time axis has the topology of a circle rather than an interval.

Tail-biting trellises were studied earlier in [**9, 11**] in a different context. In particular, it has long been known that certain block codes have quasi-cyclic, or double-circulant, realizations leading to trellis representations with remarkably few states. For example, Solomon and van Tilborg [**11**] showed that a certain $(22,11,7)$ subcode of $\mathbb{C}_{24}$ could be realized with an 8-state quasi-cyclic encoder, and that this encoder could be augmented to generate the Golay code itself. Using this encoder, Solomon and van Tilborg [**11**] found a 64-state tail-biting trellis for $\mathbb{C}_{24}$, as did Kudryashov and Zakharova [**7**] somewhat later.

The starting point for our investigation was a result of Wiberg, Loeliger and Kötter [**12**], who showed that the number of states in a tail-biting trellis could be as low as the *square root* of the number of states in a conventional trellis at its midpoint. This led one of us [**4**] to conjecture that there might be a tail-biting trellis for the binary Golay code with as few as 16 states. More specifically, Forney [**4**] asked if there was a generator matrix for $\mathbb{C}_{24}$ of the following form:



(1)

where blanks denote zeros, and $*$ stands for either 1 or 0. It is shown in our companion paper [**1**] that the existence of such a generator matrix would imply the existence of a six-section tail-biting trellis for $\mathbb{C}_{24}$ with only 16 states. In this work, we will not only exhibit a generator matrix for $\mathbb{C}_{24}$ which has the structure of (1), but also classify all the generator matrices of this type, under the action of M_{24} in conjunction with several trivial symmetries described below.

The twelve coordinates labeled by $\boxed{i}$, $\boxed{i+1}$ and $\boxed{i+2}$ in (1), with additions modulo 6, support a two-dimensional subcode of $\mathbb{C}_{24}$, which we denote by the i-th

letter of the alphabet. We observe that every generator in (1) must be an *octad*, that is a codeword of $\mathbb{C}_{24}$ of weight 8. The twelve coordinates that support a particular subcode can be partitioned into three disjoint *tetrads* (4-element sets), such that the union of any two tetrads is an octad in the subcode. Each collection of three disjoint tetrads can be extended uniquely to a *sextet*, which is a collection of 6 disjoint tetrads such that the union of any two is an octad. We observe that the sextet determined by octads in $\mathcal{A}$ must coincide with that determined by the octads in $\mathcal{D}$, and similarly for the subodes $\mathcal{B}, \mathcal{E}$ and $\mathcal{C}, \mathcal{F}$. The existence of the generator matrix in (1) thus reduces to the existence of three particular sextets, and to a partition of each sextet into two sets of 3 tetrads.

Given a generator matrix $\mathbf{G}_1$ as in (1), we may construct a second matrix $\mathbf{G}_2$ by replacing generators $a, a' \in \mathcal{A}$, say, with $a, a + a'$. Similarly, we may replace any pair of generators that belong to one of the subcodes $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}$ by another pair of nonzero vectors taken from the same subcode. We shall not regard generator matrices related in this way as essentially different. Furthermore, given generators a, a'; b, b'; c, c'; d, d'; e, e'; f, f' it is sometimes possible that a different pairing, such as a', b; b', c; c', d; d', e; e', f; f', a, also produces a generator matrix of the form (1). Once again, we shall not regard generator matrices related in this way as essentially different. Thus existence and uniqueness are questions about the subcodes $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}$ and $\mathcal{F}$ rather than questions about twelve individual generators. The main result of this paper is a complete classification of such generator matrices up to automorphisms in M_{24}.

The structure of the generator matrix in (1) can be further refined to produce an even more interesting tail-biting trellis for $\mathbb{C}_{24}$. Specifically, we ask whether there exists a generator matrix for the Golay code of the following form:

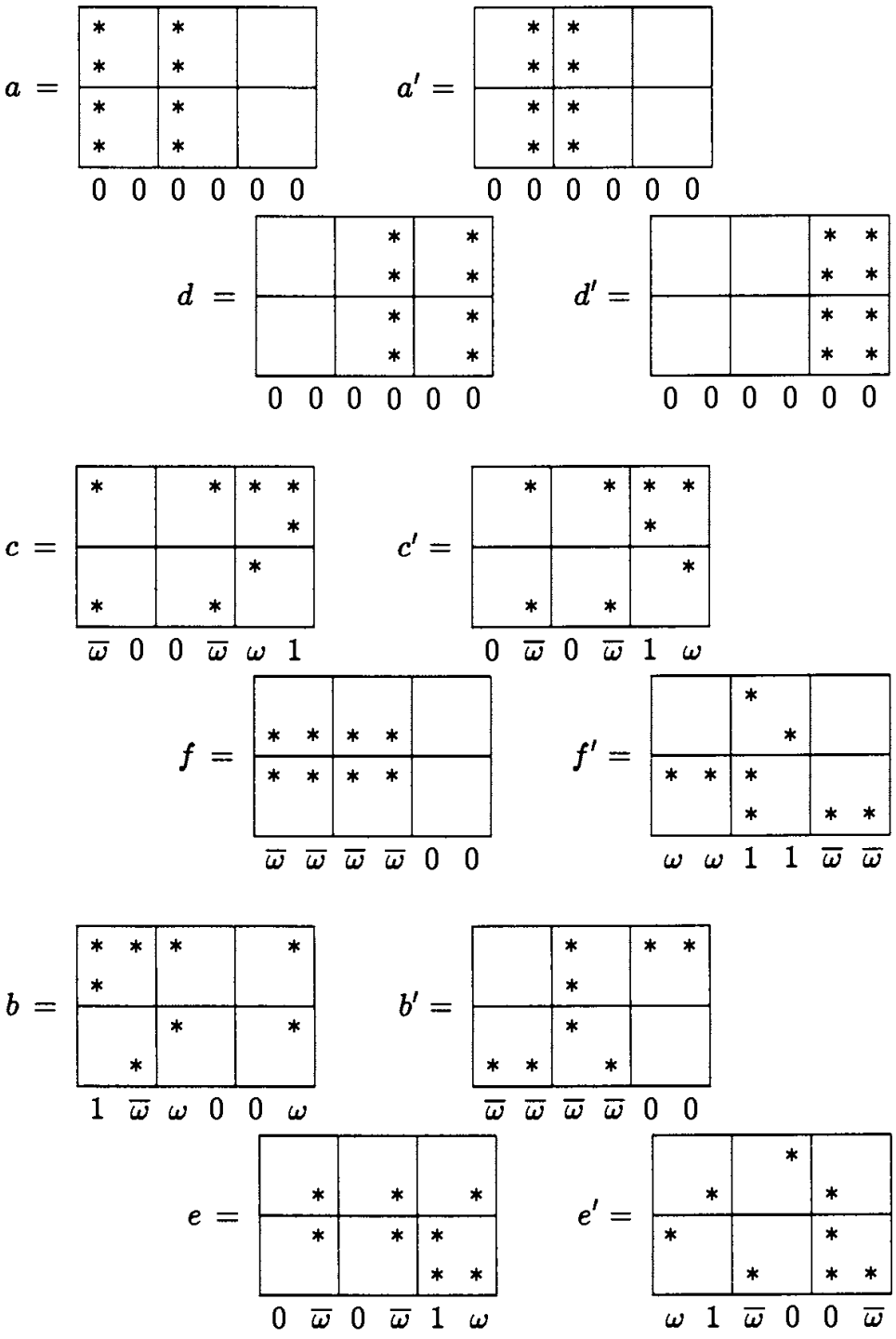
$$
\begin{array}{c|cccccc}
 & \boxed{1} & \boxed{2} & \boxed{3} & \boxed{4} & \boxed{5} & \boxed{6} \\
\mathcal{A} & \begin{array}{cc} **** & **** \\ ** & **** \end{array} & \begin{array}{cc} **** & ** \\ **** & **** \end{array} & \begin{array}{cc} ** & \\ **** & **** \end{array} & & & \\
\mathcal{B} & & \begin{array}{cc} **** & ** \\ ** & **** \end{array} & \begin{array}{cc} **** & ** \\ **** & **** \end{array} & \begin{array}{cc} ** & \\ **** & **** \end{array} & & \\
\mathcal{C} & & & \begin{array}{cc} **** & ** \\ ** & **** \end{array} & \begin{array}{cc} **** & ** \\ **** & **** \end{array} & \begin{array}{cc} ** & \\ **** & **** \end{array} & \\
\mathcal{D} & & & & \begin{array}{cc} **** & ** \\ ** & **** \end{array} & \begin{array}{cc} **** & ** \\ **** & **** \end{array} & \begin{array}{cc} ** & \\ **** & **** \end{array} \\
\mathcal{E} & \begin{array}{cc} ** & \\ **** & \end{array} & & & & \begin{array}{cc} **** & **** \\ ** & **** \end{array} & \\
\mathcal{F} & \begin{array}{cc} **** & ** \\ **** & **** \end{array} & & & & & \begin{array}{cc} **** & \\ ** & \end{array}
\end{array}
\qquad (2)
$$

We have shown in [1] that such a matrix produces a 12-section tail-biting trellis for $\mathbb{C}_{24}$ with 16 states, wheres the matrix in (1) produces a 6-section trellis. The 12-section trellis is more regular, and requires less decoding operations. Furthermore, by "unwrapping" the tail-biting generators in (2), we obtain generators for a time-varying, 16-state, binary convolutional code, with rate $1/2$ and minimum distance $d_{\text{free}} = 8$. This improves on the best possible [8] time-invariant, 16-state, convolutional code of rate $1/2$, which has minimum distance $d_{\text{free}} = 7$.

the MOG, to obtain the *projection map* π from $\mathbb{F}_2^{24}$ to $\mathbb{F}_4^6$ that respects binary addition. With this notation, we can now define the binary Golay code $\mathbb{C}_{24}$ as the set of all balanced vectors $x \in \mathbb{F}_2^{24}$ such that $\pi(x)$ is a hexacodeword.

3. Tail-biting generators and their symmetries

We start by exhibiting a set of generators for $\mathbb{C}_{24}$, which has the structure required in (1). This set of generators is given in the MOG notation below:



We do not elaborate upon how this set of generators was found. A careful reader will find clues to this effect scattered throughout our classification argument.

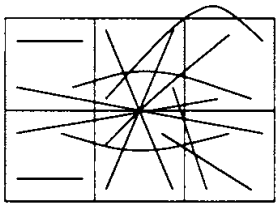
To see that these twelve vectors indeed generate the Golay code, observe that each one of them is a codeword of $\mathbb{C}_{24}$ according to the definition of $\mathbb{C}_{24}$ given in § 2. Further observe that the hexacodewords associated with these vectors span the hexacode, and that $\langle a, a', d, d', c+c'+f, f+b' \rangle$ is the 6-dimensional binary space associated with the zero hexacodeword. This implies that the twelve vectors are linearly independent, and thus generate the entire Golay code.

The corresponding coordinate labeling of the MOG is obtained by intersecting supports of the different two-dimensional subspaces. For example, the set of points labeled $\boxed{1}$ in (1) and (2) corresponds to $\mathsf{supp}(\mathcal{A}) \cap \mathsf{supp}(\mathcal{E})$. Similarly, the set of points labeled $\boxed{2}$ corresponds to $\mathsf{supp}(\mathcal{B}) \cap \mathsf{supp}(\mathcal{F})$, and so forth. The complete labeling of the MOG is as follows:

3	3	2	5	4	4
2	1	2	6	5	5
1	1	2	6	5	4
3	3	1	4	6	6

(3)

Now let ϕ be a symmetry of M_{24} that fixes the set $\{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}\}$. Then ϕ necessarily respects the grouping $(\mathcal{A}, \mathcal{D})$, $(\mathcal{B}, \mathcal{E})$, $(\mathcal{C}, \mathcal{F})$, permutes the three sextets associated with the generator matrices in (1),(2), and respects the coordinate labeling in (3). It is straightforward to verify that the involution

$\psi =$  (4)

satisfies these properties. It acts on the generators $a, a', d, d', \ldots, e, e'$ listed on the previous page as

$$(a, c')(a', c)(a+a', c+c')(d, f+f')(d', f)(d+d', f')(b')(b, b+b')(e)(e', e+e)$$

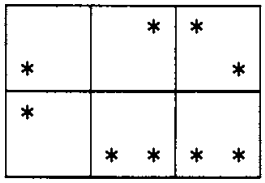
and on the coordinate labels in (3) as (15)(24)(3)(6). Since the twelve codewords $a, a', d, d', \ldots, e, e'$ form a basis for $\mathbb{C}_{24}$, it follows that ψ is a symmetry in M_{24}.

Let $\mathcal{G}$ be the subgroup of M_{24} that permutes the two-dimensional subcodes $\mathcal{A}, \mathcal{B}$, $\mathcal{C}, \mathcal{D}, \mathcal{E}$ and $\mathcal{F}$ among themselves. We have the following lemma.

Lemma 1. $\mathcal{G} = \langle \psi \rangle$.

Proof. If $\mathcal{G} \neq \langle \psi \rangle$ then there exists a nontrivial symmetry $\phi \in \mathcal{G}$ that fixes the set $\{\mathcal{A}, \mathcal{D}\}$. Thus ϕ permutes the columns of the MOG array. The intersection patterns of the coordinates labeled $\boxed{2}$ and $\boxed{3}$ with the columns of the MOG are $(3, 1, 0^4)$ and $(2^2, 0^4)$, while all the other intersection patterns are $(2, 1^2, 0^3)$. Hence ϕ does not interchange $\mathcal{A}$ with $\mathcal{D}$, and in fact ϕ fixes the first three columns of the MOG array. Observe that ϕ cannot act as $(\mathcal{B}, \mathcal{F})(\mathcal{C}, \mathcal{E})$ since the third column of the MOG is included in the support of $\mathcal{F}$ and no column of

the MOG is entirely included in the support of $\mathcal{B}$. It follows that ϕ fixes each two-dimensional subcode. By considering the supports of $\mathcal{F}$ and $\mathcal{B}$, we can now conclude that ϕ fixes the last three columns of the MOG array. Since ϕ respects the coordinate labeling, it follows that ϕ fixes the coordinates shown below



Thus ϕ fixes at least 10 points. This is a contradiction, since it is known [**2**] that a nontrivial symmetry in M_{24} fixes at most 8 points. ∎

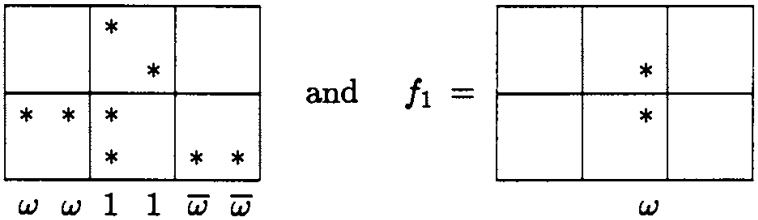
4. Classification and analysis of tail-biting generating sets

We will use the symmetry group M_{24} to bring an arbitrary representation associated with the two-dimensional subcodes $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}$ as close as possible to the particular representation described in § 3. Since M_{24} is transitive on sextets we may assume that $\mathcal{A} = \langle a, a' \rangle$ and $\mathcal{D} = \langle d, d' \rangle$. The symmetries available now belong to the stabilizer of the standard sextet. This is the group $(2^6) : 3 \times \mathsf{S}_6$, in which the subgroup $(2^6) : 3$ consists of permutations that preserve every tetrad, while the quotient group S_6 describes the action on the tetrads. For more details on the sextet stabilizer group, see Conway and Sloane [**2**, Chapter 11.9].

Notice that at least one of the twelve octads in the generating set must be odd. After possibly interchanging $\mathcal{A}$ with $\mathcal{D}$, we may suppose that the subcode $\mathcal{F}$ contains an even octad f_1 and two odd octads f_2 and $f_1 + f_2$. The support of $\mathcal{F}$ meets the support of $\mathcal{A}$, namely columns 1, 2, and 3 of the MOG, in eight points and the support of $\mathcal{D}$, namely columns 4, 5, 6 of the MOG, in four points. We can use the sextet stabilizer group to make f_2 meet column 3 of the MOG in three points, and to make the support of $\mathcal{F}$ meet columns 4, 5, and 6 of the MOG in two, one, and one point(s), respectively.

Now f_1 is associated with a hexacodeword $(aa\ aa\ 00)$ and after multiplication by some power of ω we may suppose that this hexacodeword is $(\overline{\omega\omega}\ \overline{\omega\omega}\ 00)$. Recall that involutions in the elementary abelian subgroup (2^6) of the sextet stabilizer group correspond to addition of hexacodewords, and that this subgroup preserves the set of 32 even Golay codewords associated with a fixed hexacodeword. We use this elementary abelian subgroup to associate f_2 with the hexacodeword $(\omega\omega\ 11\ \overline{\omega\omega})$. At this point we have:

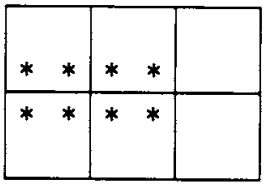
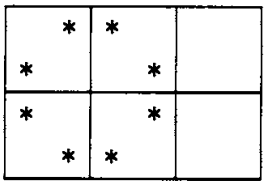
$$f_2 = f' = \quad \text{and} \quad f_1 = \qquad (5)$$



Notice that the other interpretation for $\overline{\omega}$ in column 4 of (5) would imply that the support of $\mathcal{F}$ meets columns 4, 5, 6 of the MOG in a total of five points.

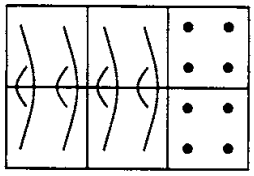
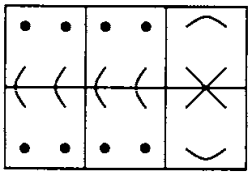
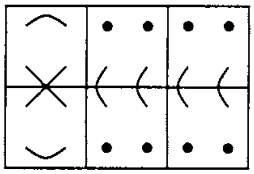
This would contradict the fact that the support of $\mathcal{F}$ meet columns 4, 5, and 6 in two, one, and one point(s), respectively, established in the foregoing paragraph.

After possibly interchanging entries in the left and right couples in (5), we see that there are only two alternatives for f_1, namely:

$f_1 = f =$  and $f_1 =$  (6)

We will refer to these two alternatives as the *rectangular case* and the *circular case*. The two cases are investigated in the following two subsections.

4.1. Completing the classification: the rectangular case. Obviously, the subcode $\mathcal{F} = \langle f, f' \rangle$ determines the sextet associated with $\mathcal{C}, \mathcal{F}$. At this point, eihght of the twelve octads in the generating set are determined, and we have $\mathcal{A} = \langle a, a' \rangle$, $\mathcal{D} = \langle d, d' \rangle$, $\mathcal{C} = \langle c, c' \rangle$ and $\mathcal{F} = \langle f, f' \rangle$. Define:

$\alpha_1 =$  $\alpha_2 =$  $\alpha_3 =$ 

It is easy to see that $\alpha_1, \alpha_2, \alpha_3$ are symmetries in M_{24}. Indeed, recall that the subgroup fixing an octad pointwise is elementary abelian of order 16, and the action on the complementary sixteen points is that of the additive group of a 4-dimensional vector space. For more details, see the description of the octad group $2^4 : A_8$ in Conway and Sloane [2, Chapter 11]. By completing octads from five of their points, it is straightforward to extend any transposition of two of the complementary 16 points to the appropriate involution.

Let $\mathfrak{B}$ be the subgroup of M_{24} that permutes the two-dimensional subcodes $\mathcal{A}, \mathcal{D}, \mathcal{C}, \mathcal{F}$ among themselves. We have the following lemma.

Lemma 2. *$\mathfrak{B} = \langle \psi, \alpha_1, \alpha_2, \alpha_3 \rangle$ is a group of order 16.*

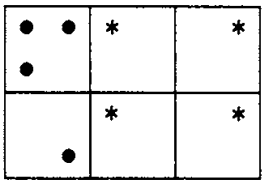
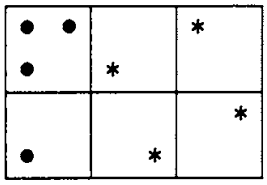
Proof. If $\phi \in \mathfrak{B}$, then after possibly applying ψ we may assume ϕ fixes the standard sextet (associated with $\mathcal{A}$ and $\mathcal{D}$) and the sextet associated with $\mathcal{C}, \mathcal{F}$. If ϕ interchanges $\mathcal{A}$ with $\mathcal{D}$, then ϕ interchanges $\mathcal{C}$ with $\mathcal{F}$, and it follows that coordinates labeled 3 below are interchanged with coordinates labeled 6.

3 3	6	
3 3	6	6 6

However this cannot be achieved by a symmetry that preserves columns of the MOG array. Hence ϕ permutes columns of the MOG array, fixes $\mathcal{A}, \mathcal{D}, \mathcal{C}, \mathcal{F}$, and fixes coordinates labeled $\boxed{3}$ and $\boxed{6}$. After possibly applying α_2, α_3 or $\alpha_2\alpha_3$, we may suppose that ϕ fixes every column of the MOG array. It follows that ϕ

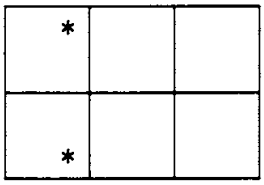
belongs to the elementary abelian subgroup (2^6) of the sextet stabilizer group, where involutions correspond to addition of hexacodewords. We see that ϕ must correspond to the addition of $(\overline{\omega\omega}\ \overline{\omega\omega}\ 00)$, that is $\phi = \alpha_1$. ∎

There must be an octad in the generating set associated with a hexacodeword of type $(a, a{+}1\ b, b{+}1\ c, c{+}1)$ or of type $(a, a{+}\omega\ b, b{+}\omega,\ c, c{+}\omega)$. Furthermore, this octad belongs to subcode $\mathcal{B}$ because it includes an odd number of points labeled $\boxed{3}$. Now consider the restriction of $\mathcal{B}$ to the coordinates labeled $\boxed{3}$. The weight distribution of such restriction is $0, 1, 3, 4$ or $0, 2, 3, 3$, and in either case there is an octad b_1 that includes 3 points labeled $\boxed{3}$. After possibly applying a symmetry in $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$ we may assume that

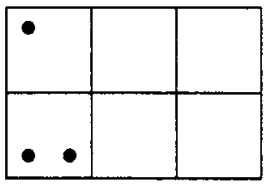
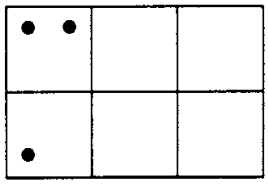
$b_1 = b =$  or $b_1 = b + b' =$ 

for in each case there is only one way to complete the points labeled • to an octad that includes no points labeled $\boxed{6}$. Note that the involution ψ in (4) interchanges the two choices for b_1, so we may assume that $b_1 = b$. At this point nine of the twelve octads in the generating set are determined.

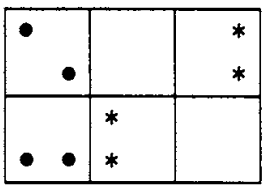
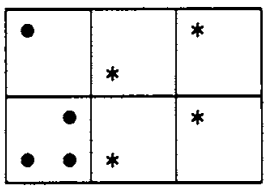
Determining the last three octads in the rectangular case. First suppose that the restriction of $\mathcal{B}$ to the coordinates labeled $\boxed{3}$ includes a vector of weight 4. It follows by inspection that the vector

$v =$ 

is orthogonal to every generator (octads in $\mathcal{E}$ do not include points labeled $\boxed{3}$). This is a contradiction, since $\mathbb{C}_{24}$ is self-dual and the minimum distance of $\mathbb{C}_{24}$ is 8. We conclude that there is a second octad $b_2 \in \mathcal{B}$ that includes three points labeled $\boxed{3}$, and that is not orthogonal to v. There are two cases to consider

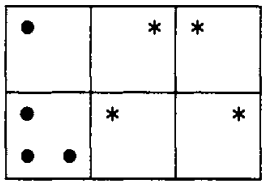
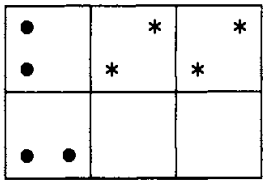
Case ▲: $b_2 =$  **Case ▼:** $b_2 =$ 

Case ▲. Suppose b_2 is even. Then there are only two ways to complete the left brick to an octad that includes no points labeled $\boxed{6}$, namely:

 and 

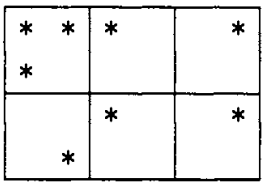
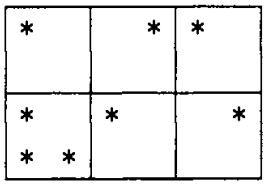
In both cases the support of $\mathcal{B}$ meets columns 1, 2, and 3 of the MOG in more than 8 points. This is a contradiction, since the supports of $\mathcal{B}$ and $\mathcal{A}$ intersect

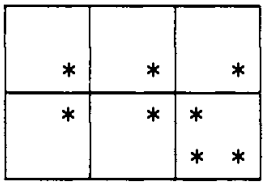
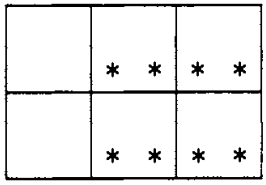
in exactly 8 points. Hence b_2 must be odd. Again, there are only two ways to complete the left brick to an octad that contains no points labeled $\boxed{6}$, namely:

$b_2 =$  and $b_2^* =$ 

It is straightforward to verify that the symmetry $\psi\alpha_1\alpha_2$ maps b to b_2^* and b_2 to b. Thus Case ▲ leads to a single isomorphism type of tail-biting trellis.

Type 1.

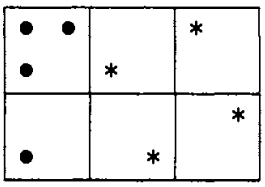
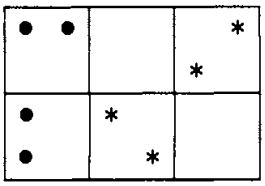
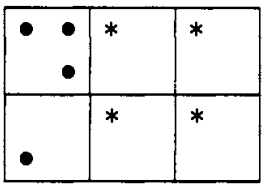
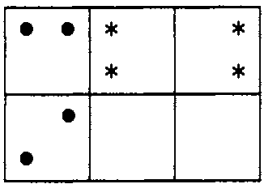
$b =$  $b' =$ 

$e =$  $e' =$ 

The coordinate labeling is:

3	3	2	4	4	4
2	1	1	6	5	5
2	1	2	6	5	4
3	3	1	5	6	6

Case ▼. There are four different ways to add a point to the left brick, and in each case there is only one way to complete the left brick to an octad that contains no points labeled $\boxed{6}$. The four alternatives are:

The last of these alternative is not allowed because then the support of $\mathcal{B}$ meets the support of $\mathcal{A}$, namely columns 1, 2, 3 of the MOG, in more than 8 points. Thus Case ▼ leads to three different isomorphism types of tail-biting trellis.

Type 2. *This is precisely the generating set we have exhibited at the beginning of the previous section. The complete coordinate labeling is given in* (3).

Type 3.

$b =$

* *	*	*
*		
	*	*
*		

$b' =$

* *		*
		*
*	*	
*	*	

$e =$

	*	*
*		*
*	*	
		* *

$e' =$

	*	*
*		*
	*	*
*		*

The coordinate labeling is:

3	3	2	5	5	4
2	1	1	6	4	5
2	1	2	6	5	4
3	3	1	4	6	6

Type 4.

$b =$

* *	*	*
*		
	*	*
*		

$b' =$

* *	*	*
*		
	*	*
*		

$e =$

	* *	* *
	* *	* *

$e' =$

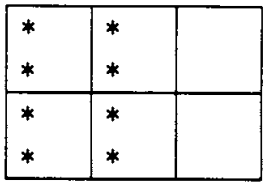
	*	
*		
* *	*	
	*	* *

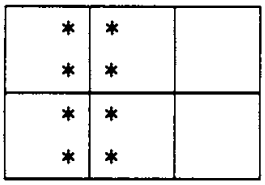
The coordinate labeling is:

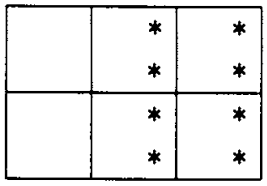
3	3	2	5	4	4
2	2	1	6	5	5
2	1	2	6	4	4
3	3	1	5	6	6

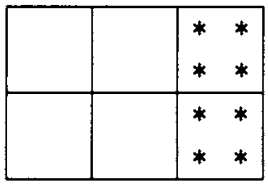
After changing the pairing of generators to $a+a', b$; $b+b', c+c'$; c', d'; $d+d', e$; $e', f'+f$; f, a, we find that this is isomorphic to the generating set of Type 3.

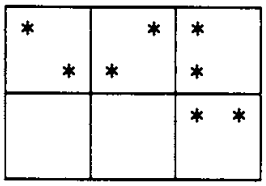
4.2. Completing the classification: the circular case. Once again, the subcode $\mathcal{F} = \langle f_1, f' \rangle$ determines the sextet associated with $\mathcal{C}, \mathcal{F}$. We start by exhibiting a set of generators with the required form below.

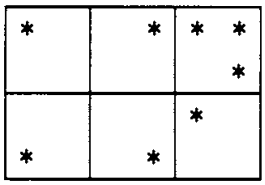
$a =$ 

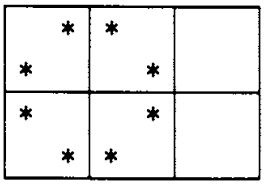
$a' =$ 

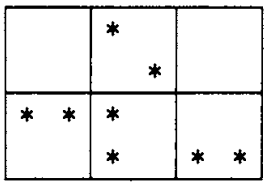
$d =$ 

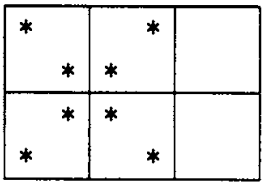
$d' =$ 

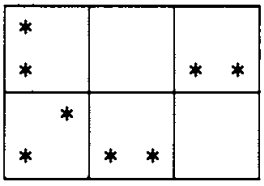
$c =$ 

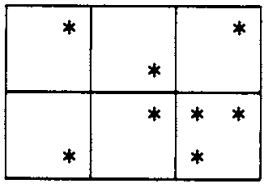
$c' =$ 

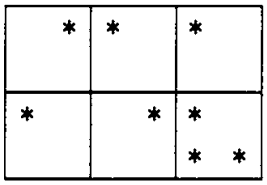
$f =$ 

$f' =$ 

$b =$ 

$b' =$ 

$e =$ 

$e' =$ 

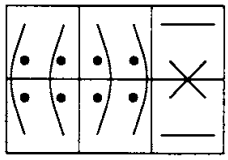
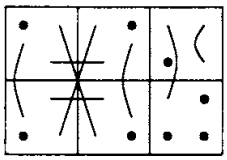
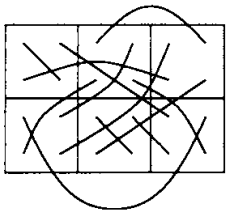
Observe that the hexacodewords associated with these generators span the hexacode and that $\langle a, a', d, d', b + f_1, c + e + e' + f' \rangle$ is the six-dimensional binary space associated with the zero hexacodeword. Thus the above twelve Golay codewords generate $\mathbb{C}_{24}$, and hence are linearly independent.

We obtain the corresponding coordinate labeling by intersecting supports of the two-dimensional subcodes $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}$. This labeling is given by:

3	1	1	4	5	5
2	3	3	6	4	4
1	2	2	6	5	5
3	1	2	4	6	6

(7)

In order to classify the different generating sets that arise in the circular case, we need to understand the structure of the subgroup $\mathfrak{B}$ of M_{24} that permutes the two-dimensional subcodes $\mathcal{A}, \mathcal{D}, \mathcal{C}, \mathcal{F}$ among themselves. To this end, consider the symmetries $\alpha_4, \alpha_5, \delta \in \mathfrak{B}$, defined as follows:

$\alpha_4 =$  $\alpha_5 =$  $\delta =$ 

It is straightforward to verify that δ acts on the generators exhibited on the previous page as $(a,c)(a',c+c')(a+a',c')(d',f')(d,f_1+f')(d+d',f_1)(b)(b,b+b')(e,e')(e+e')$ and on coordinate labels as $(1,5)(2,4)(3)(6)$. Since these generators form a basis for the Golay code, this implies δ is an automorphism in M_{24}. Observe that δ interchanges the sextets $\mathcal{A}, \mathcal{D}$ and $\mathcal{C}, \mathcal{F}$. On the other hand, the symmetries α_4, α_5 both fix an octad pointwise, and we may verify that α_4, α_5 are automorphisms in M_{24} by completing octads as described earlier in § 4.1.

Given an arbitrary symmetry $\phi \in \mathfrak{B}$, we may apply δ and assume that ϕ fixes the sextets $\mathcal{A}, \mathcal{D}$ and $\mathcal{C}, \mathcal{F}$. It is easy to see that ϕ cannot act as $(\mathcal{A}, \mathcal{D})(\mathcal{C}, \mathcal{F})$ since $\mathsf{supp}(\mathcal{A}) \cap \mathsf{supp}(\mathcal{C})$ and $\mathsf{supp}(\mathcal{D}) \cap \mathsf{supp}(\mathcal{F})$ meet the standard sextet in different ways. Hence ϕ fixes $\mathcal{A}$, $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{F}$. Observe that $\mathsf{supp}(\mathcal{F})$ distinguishes column 1 of the MOG from columns 2 and 3. Thus, after possibly applying α_5, we may assume ϕ fixes columns 1, 2 and 3 of the MOG. Once the first three columns are fixed, then generators $f_1, f', f_1 + f'$ must all be fixed, because they are distinguished by the way they meet column 3. It now follows that $\phi = \alpha_4$, since the octad of fixed points can be distinguished from the way generators in $\mathcal{F}$ meet columns 1, 2, 3 and 4, there can be no more fixed points, and specifying a single transposition determines α_4. We have thus proved the following lemma.

Lemma 3. *$\mathfrak{B}$ is a group of order* 8, *and* $\mathfrak{B} = \langle \delta, \alpha_4, \alpha_5 \rangle$.

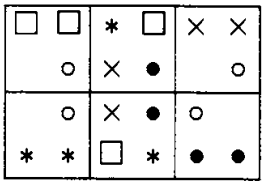
Now consider $w_1, w_2, w_3, w_4, w_5 \in \mathbb{F}_2^{24}$, where w_i is the vector of weight 2, with the two nonzero entries at positions labeled i' in the MOG array below.

$1'$	$2'$ $3'$	$4'$ $5'$
		$5'$
		$4'$
$1'$	$2'$ $3'$	

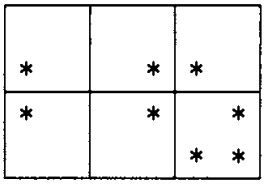
Observe that each of the vectors w_1, w_2, w_3, w_4, w_5 is orthogonal to every octad in the two-dimensional subcodes $\mathcal{A}$, $\mathcal{D}$, $\mathcal{C}$ and $\mathcal{F}$.

Recall that the two-dimensional subcode $\mathcal{B}$ covers points in $\mathsf{supp}(\mathcal{A}) \cap \mathsf{supp}(\mathcal{C})$, labeled $\boxed{3}$ in (7), and misses points in $\mathsf{supp}(\mathcal{D}) \cap \mathsf{supp}(\mathcal{F})$, labeled $\boxed{6}$ in (7). By symmetry, the subcode $\mathcal{E}$ covers points labeled $\boxed{6}$ and misses points labeled $\boxed{3}$. The weight distribution of the restriction of $\mathcal{E}$ to the points labeled $\boxed{6}$ is either $0,1,3,4$ or $0,2,2,4$ or $0,2,3,3$. We will consider these cases separately below.

Case A. Here the weight distribution is $0,1,3,4$. Thus there exists an octad in $\mathcal{E}$ that covers all points labeled $\boxed{6}$. We complete the points labeled $\boxed{6}$ to a sextet as shown below, and observe that there is only one way to choose an octad e missing all points labeled $\boxed{3}$.

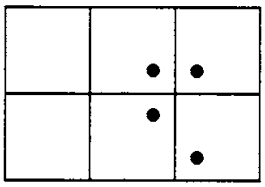


$e =$

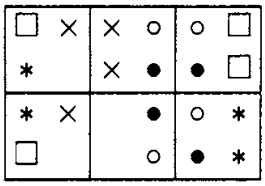


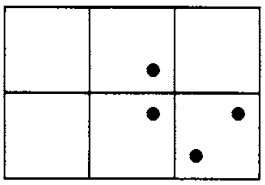
Observe that e is fixed by the entire group $\mathfrak{B}$. Note that at this point nine of the twelve octads in the generating set are determined.

First we suppose there is an octad e' that meets e in four points and involves three points labeled $\boxed{6}$. After applying symmetries from $\mathfrak{B}$ there are three possibilities for $e \cap e'$, and we complete each tetrad to a sextet as shown below:

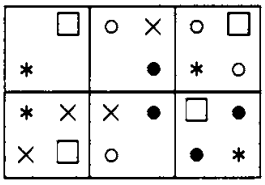


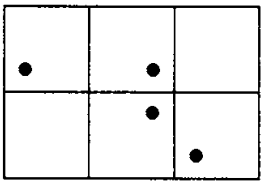
$\rightsquigarrow \quad \mathcal{S}_1 =$



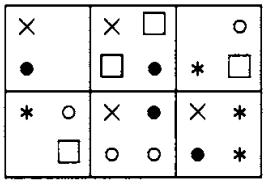


$\rightsquigarrow \quad \mathcal{S}_2 =$



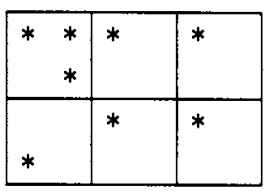


$\rightsquigarrow \quad \mathcal{S}_3 =$

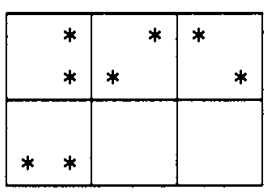


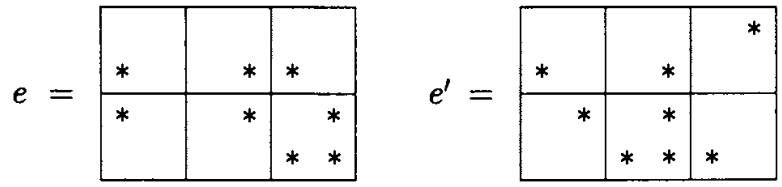
Observe that $w_1 + w_2$ is orthogonal to all octads in $\mathcal{S}_1$ and $\mathcal{S}_2$. Since the generators must span $\mathbb{C}_{24}$, it follows that e' must be an octad in $\mathcal{S}_3$. This observation uniquely determines e', and with it the entire generating set. Thus, if the weight distribution of the restriction of $\mathcal{E}$ to the points labeled $\boxed{6}$ is $0,1,3,4$, then the four octads of the $\mathcal{B},\mathcal{E}$ sextet are given by:

$b =$



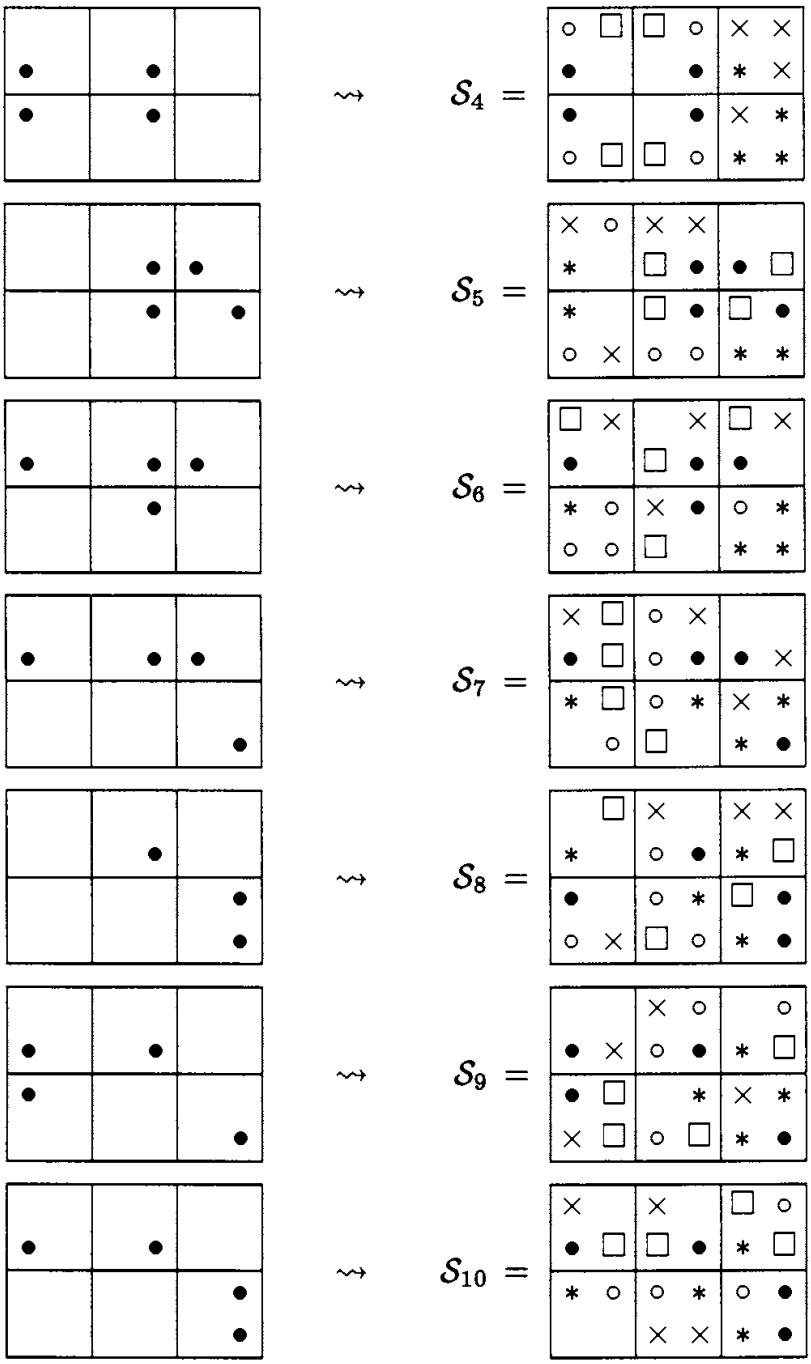
$b' =$





We have used MAGMA to verify that this generating set is isomorphic to the generating set of Type 2, exhibited at the beginning of the previous section.

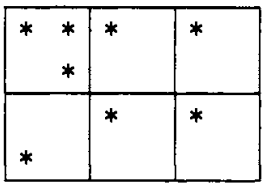
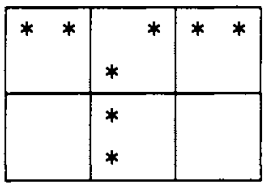
Case B. Here the weight distribution is $0, 2, 2, 4$. We look for an octad e' that meets e in four points and involves exactly two points labeled $\boxed{6}$. After applying symmetries from $\mathfrak{B}$ there are seven possibilities for $e \cap e'$, and we complete each tetrad to a sextet as shown below:

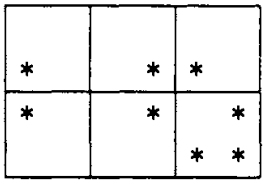
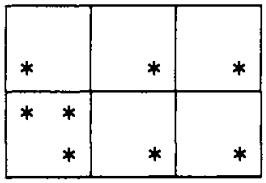


Observe that $w_1 + w_2$ is orthogonal to all octads in $\mathcal{S}_4$, $\mathcal{S}_5$, $\mathcal{S}_7$, $\mathcal{S}_8$ and $\mathcal{S}_{10}$. Hence e' is an octad in $\mathcal{S}_6$ or $\mathcal{S}_9$. This gives rise to the following generating sets.

Type 5. *This is precisely the generating set we have exhibited at the beginning of this subsection The complete coordinate labeling is given in* (7).

Type 6.

$b =$  $b' =$ 

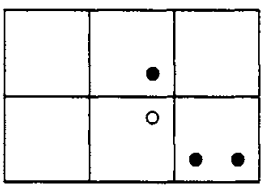
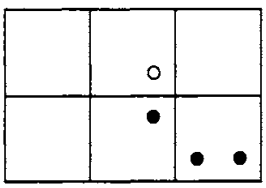
$e =$  $e' =$ 

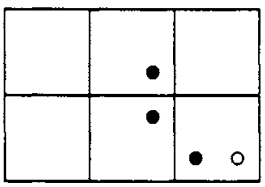
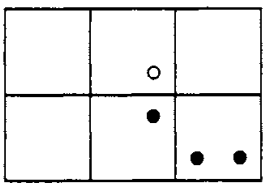
The coordinate labeling is:

3	1	2	5	4	5
1	3	3	6	5	4
1	2	1	6	4	5
3	2	2	4	6	6

We used MAGMA to verify this generating set is isomorphic to Type 1. After changing the pairing of generators to a, b; $b+b', c$; c', d'; $d, e+e'$; e, f_1+f'; f_1, a we used MAGMA to verify this generating set is also isomorphic to Type 4.

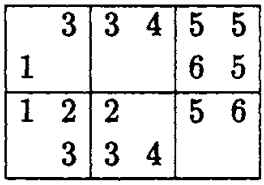
Case C. What remains is the case where the weight distribution of the restriction of the two-dimensional subcode $\mathcal{E}$ to the points labeled $\boxed{6}$ is $0, 2, 3, 3$. Notice that the group $\mathfrak{B}$ acts on the possible $(4, 2, 2)$ binary codes. There are two orbits, and we distinguish orbit representatives, as follows.

Case C(i): $e =$  $e' =$ 

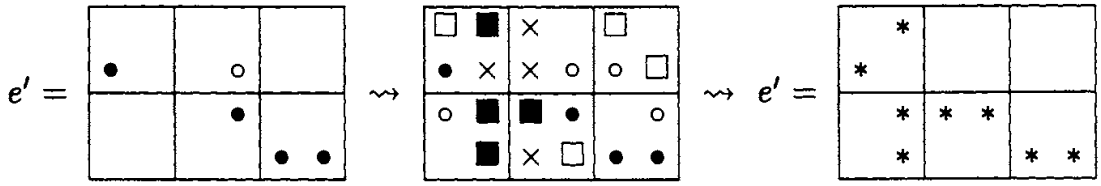
Case C(ii): $e =$  $e' =$ 

where the symbol ∘ indicates the absence of a point. Octads e and e' must meet in four points, and we enumerate the different ways to add this 4-th point.

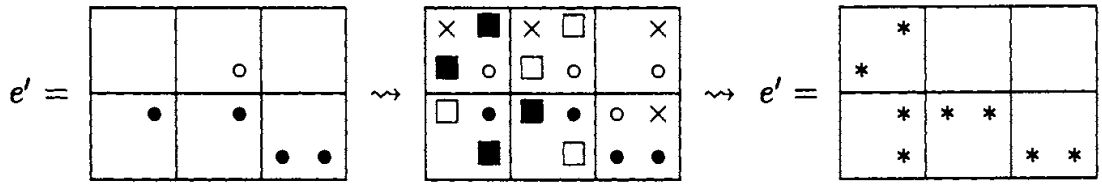
First we deal with Case C(i), where it is sufficient to consider adding a single point from each of the orbits of $\langle \alpha_4, \alpha_5 \rangle$ shown below.



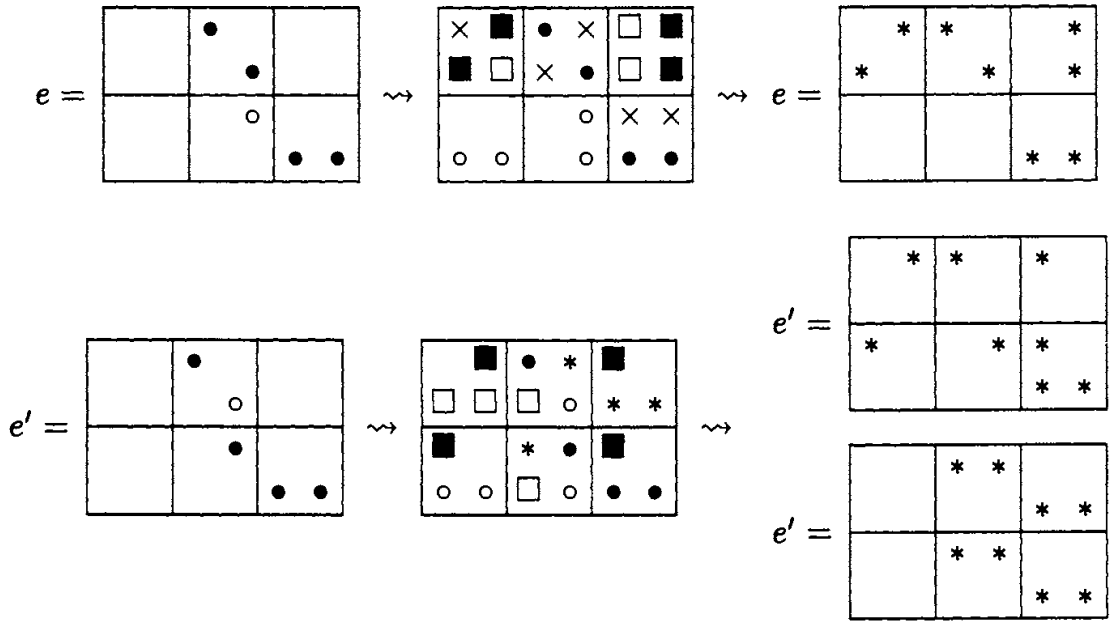
Orbit 1. We calculate the sextet $\mathcal{S}$ containing e'. There is only one choice for the octad e' and this choice meets $\mathsf{supp}(\mathcal{A})$ in five points, which is too many.



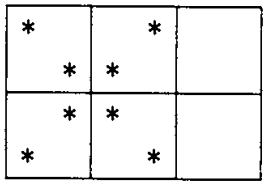
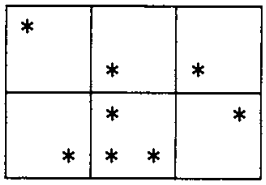
Orbit 2. Once again, we calculate the sextet containing e', and observe that this leads to a single choice for e' that meets $\mathsf{supp}(\mathcal{A})$ in too many (5) points.

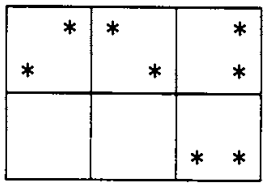
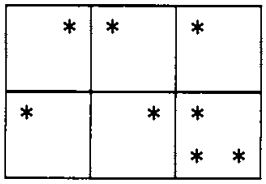


Orbit 3. We calculate the sextets $\mathcal{S}, \mathcal{S}'$ containing e, e'. There is one choice for e (since $\mathsf{supp}(\mathcal{E})$ meets $\mathsf{supp}(\mathcal{A})$ in at most four points) and two choices for e'.



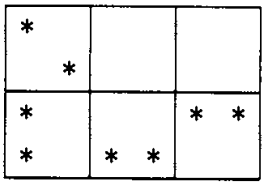
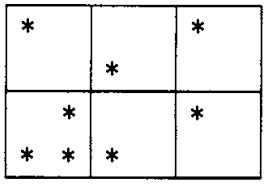
The first choice for e' leads to a completion of the $\mathcal{B}, \mathcal{E}$ sextet as follows.

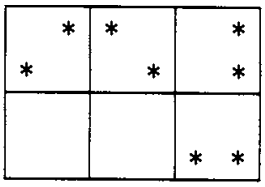
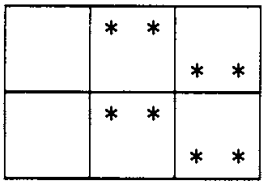
$b =$  $b' =$ 

$e =$  $e' =$ 

Since all these octads are orthogonal to $w_1 + w_2$, this is not a generating set for the Golay code. The second choice for e' leads to the following generating set.

Type 7.

$b =$  $b' =$ 

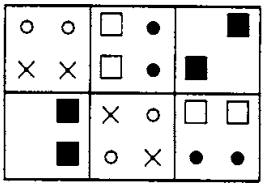
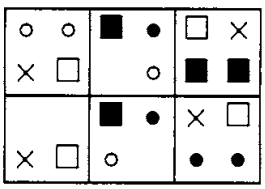
$e =$  $e' =$ 

The coordinate labeling is:

3	1	1	5	4	5
1	3	3	6	5	5
2	2	1	6	4	4
3	2	2	4	6	6

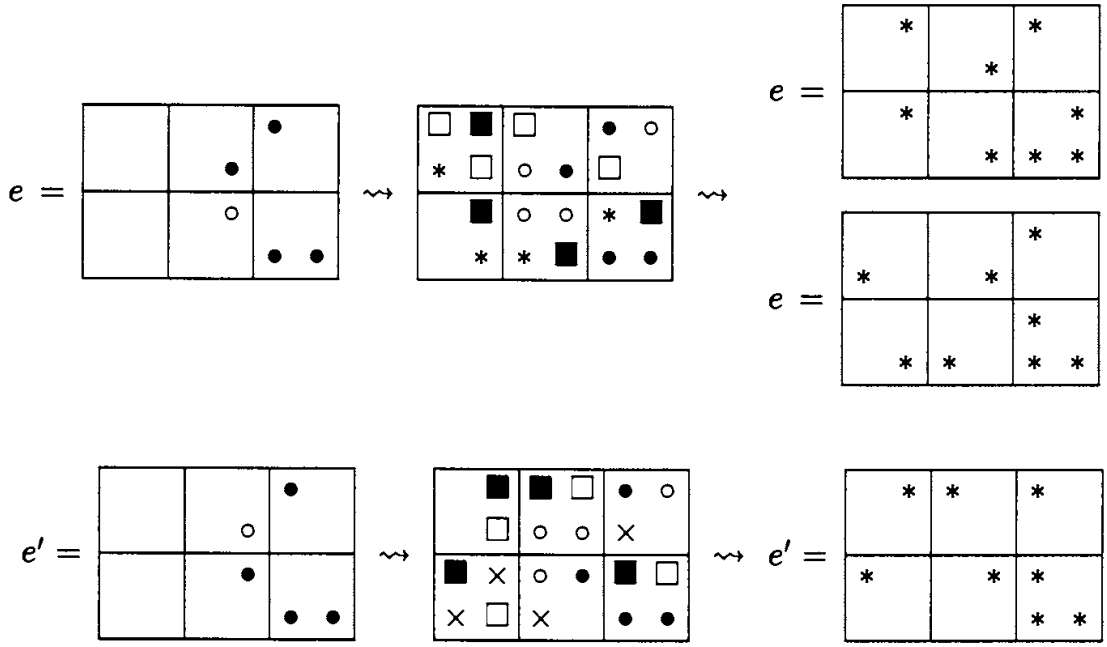
Changing the pairing of generators to $a + a', b'$; b, c; c', d'; d, e'; $e, f_1 + f'$; f_1, a', we see, using MAGMA, that this generating set is isomorphic to Type 3.

Orbit 4. We calculate the sextets $\mathcal{S}, \mathcal{S}'$ containing e, e', and observe that there is one choice for both e and e'. The two sextets are given by:

$\mathcal{S} =$  $\mathcal{S}' =$ 

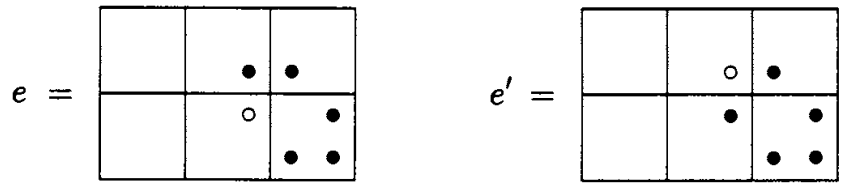
This leads to a completion of the $\mathcal{B}, \mathcal{E}$ sextet which turns out to be isomorphic to the generating set of Type 4, as we have verified using MAGMA.

Orbit 5. Once again, we calculate the sextets $\mathcal{S}, \mathcal{S}'$ containing e, e'. There is one choice for e' and two choices for e.



The first choice for e meets e' in a point from Orbit 3, so it has been considered previously. The second choice for e is not possible since it forces $\mathsf{supp}(\mathcal{E})$ to meet $\mathsf{supp}(\mathcal{A})$ in six points.

Orbit 6. If the generating set has not been considered previously, then e and e' contain both points from orbit 6. Thus we have:



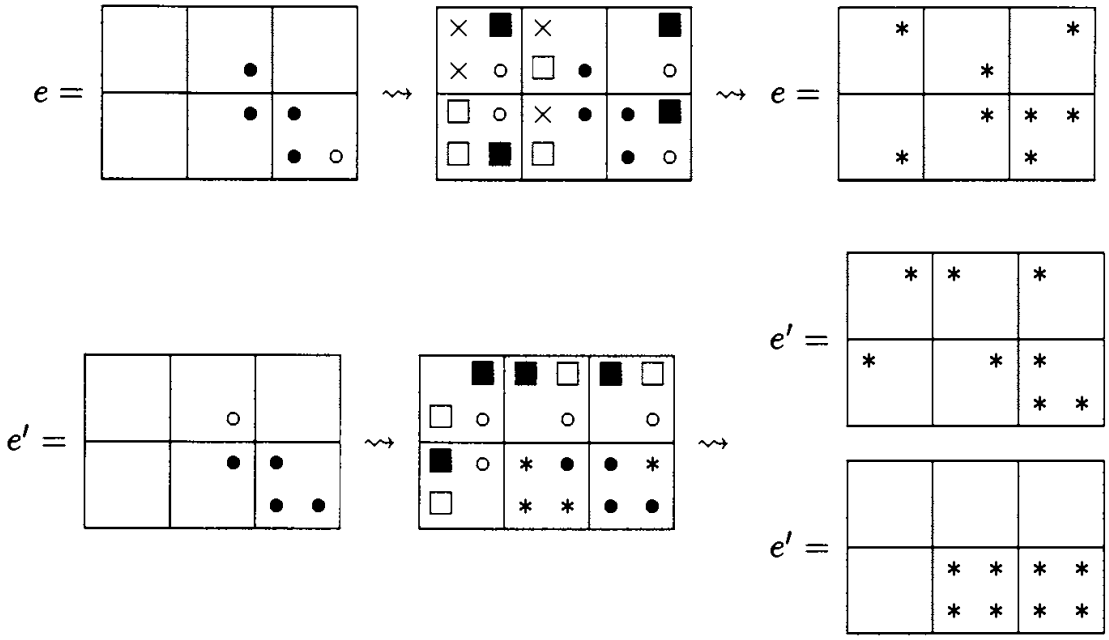
It is easy to see that there is no way to complete either e or e' to an octad.

Finally we deal with Case C(ii), where it is sufficient to consider adding a single point from each of the orbits of $\langle \delta \rangle$ shown below.

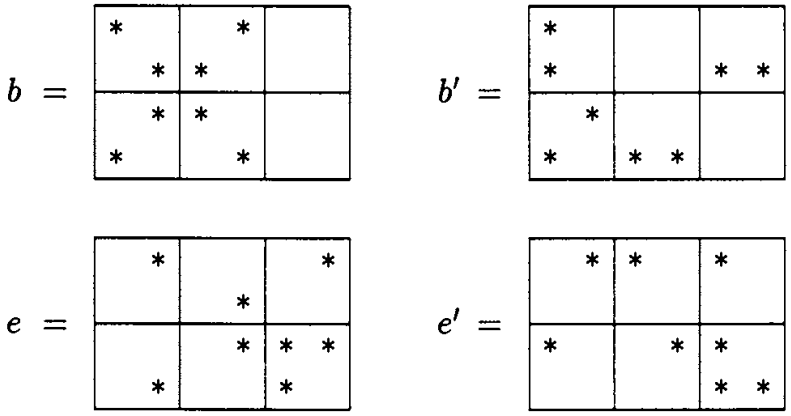
	1	6	7	5	6
4				4	3
2	7	8		1	2
	5	3	8		

This case thus gives rises to eight different orbits. We will proceed with each of these orbits, using arguments similar to those used for Case C(i).

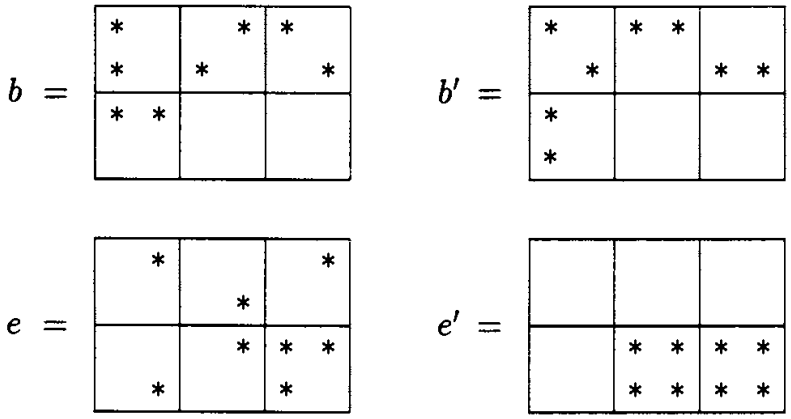
Orbit 1. We calculate the sextets $\mathcal{S}, \mathcal{S}'$ containing e, e', and find there is one choice for e and two choices for e'. Note that $e \neq d + d'$.



The first choice for e' leads to:

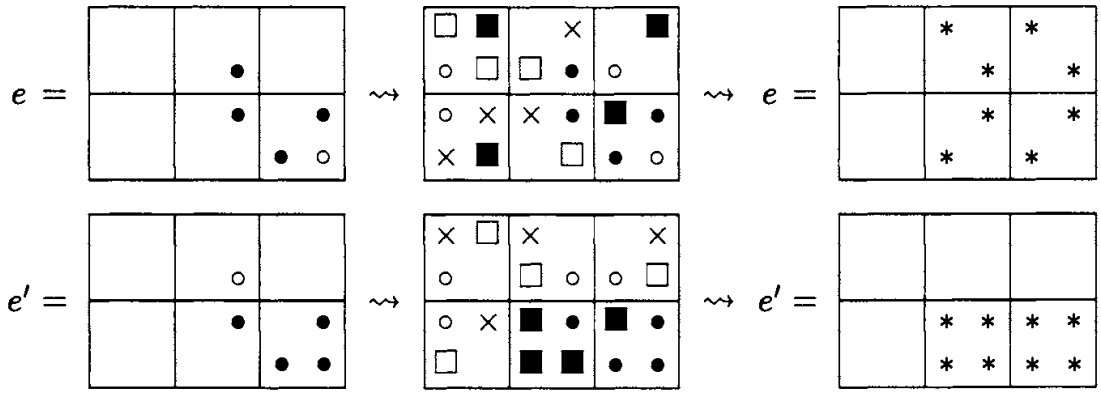


and we used MAGMA to verify that this generating set is isomorphic to Type 5. The second choice for e' leads to:



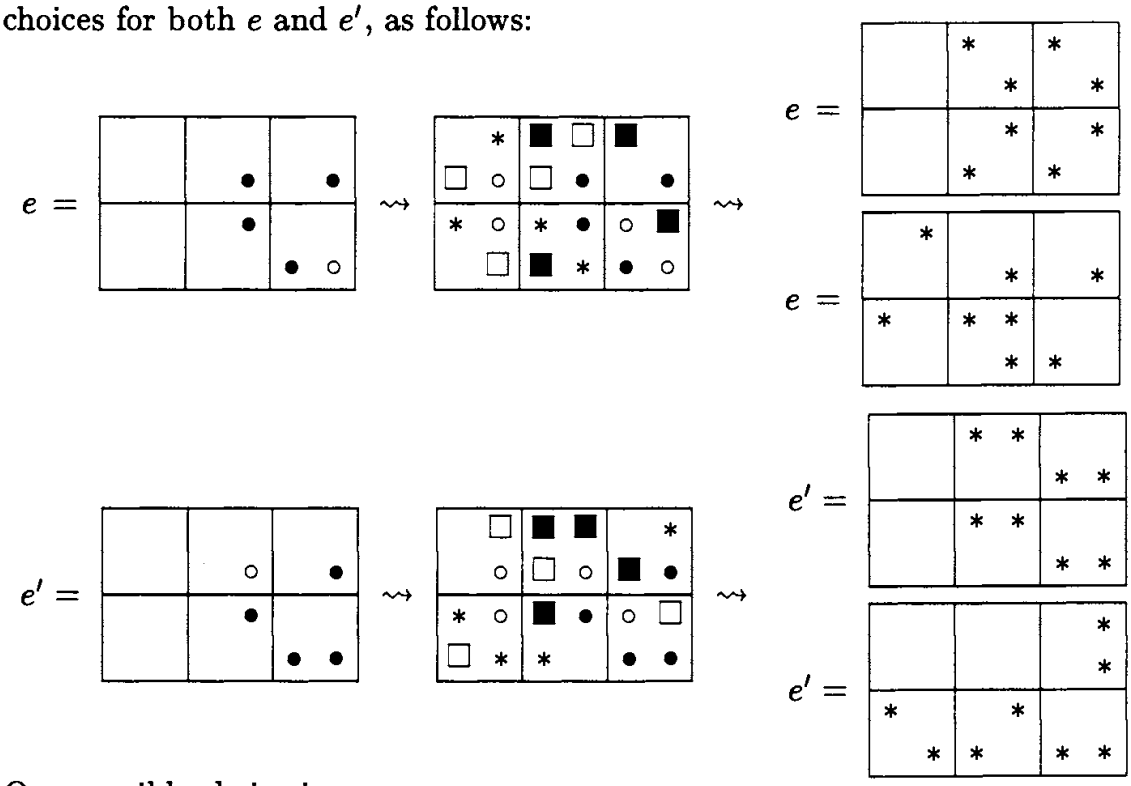
and since all these octads are orthogonal to w_1 this is not a generating set.

Orbit 2. We calculate the sextets containing e, e' and find there is one choice for e'. There are two choices for e, but one was considered previously. Thus:

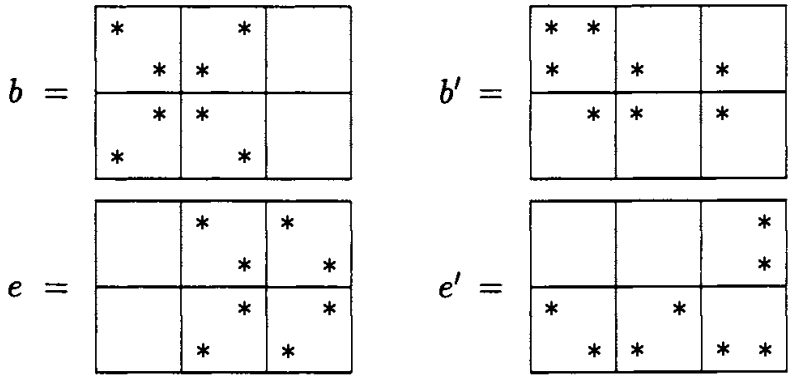


Now we have a contradiction because $\mathsf{supp}(\mathcal{E})$ meets $\mathsf{supp}(\mathcal{A})$ in only three points.

Orbit 3. We calculate the sextets $\mathcal{S}, \mathcal{S}'$ containing e, e' and find there are two choices for both e and e', as follows:

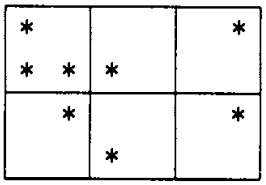
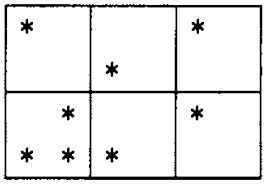


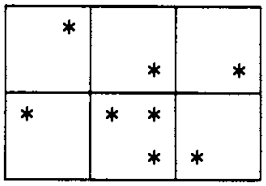
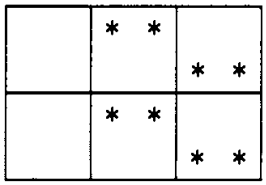
One possible choice is:



We used MAGMA to show that this generating set is isomorphic to Type 2.

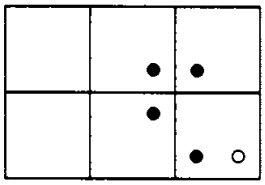
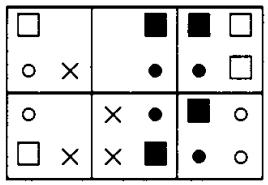
Since $\mathsf{supp}(\mathcal{E})$ must meet $\mathsf{supp}(\mathcal{A})$ in exactly four points, there is only one more possibility for e and e'. The second possible choice is as follows:

$b =$  $b' =$ 

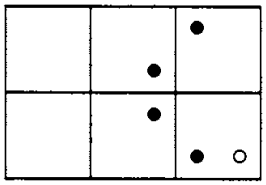
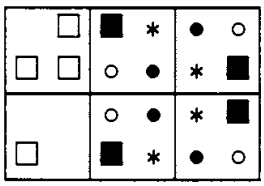
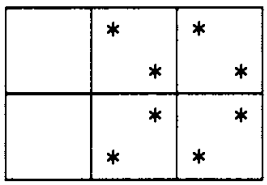
$e =$  $e' =$ 

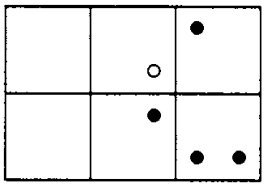
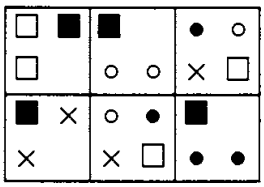
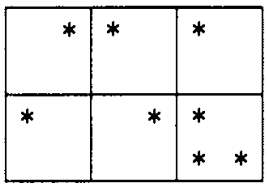
However, since all octads here are orthogonal to w_4, this is not a generating set.

Orbit 4. We calculate the sextet $\mathcal{S}$ containing e. The only choice is $e = d + d'$, which violates linear independence of the generators.

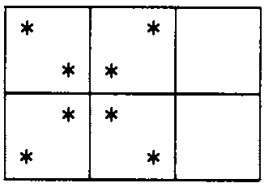
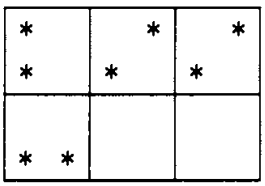
$e =$  $\rightsquigarrow$ 

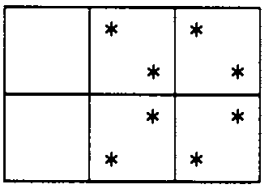
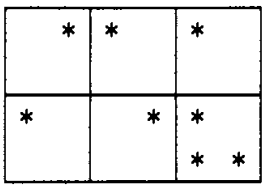
Orbit 5. We calculate the sextets $\mathcal{S}, \mathcal{S}'$ containing e, e' and find there is one choice for e, e' (note that $e \neq d + d'$).

$e =$  $\rightsquigarrow$  $\rightsquigarrow$ $e =$ 

$e' =$  $\rightsquigarrow$  $\rightsquigarrow$ $e' =$ 

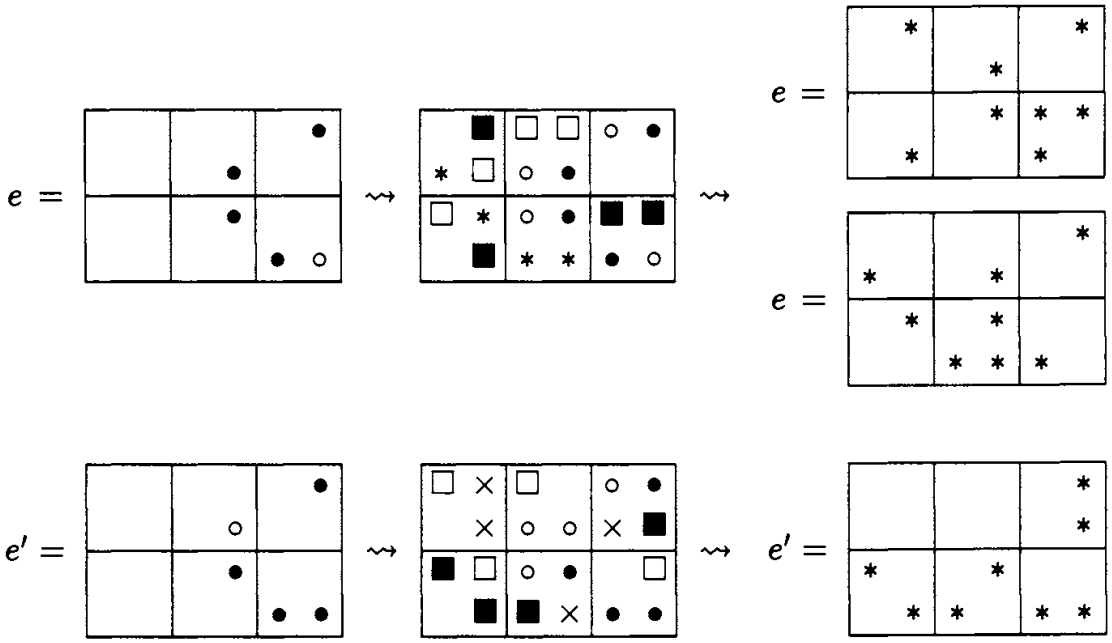
This leads to:

$b =$  $b' =$ 

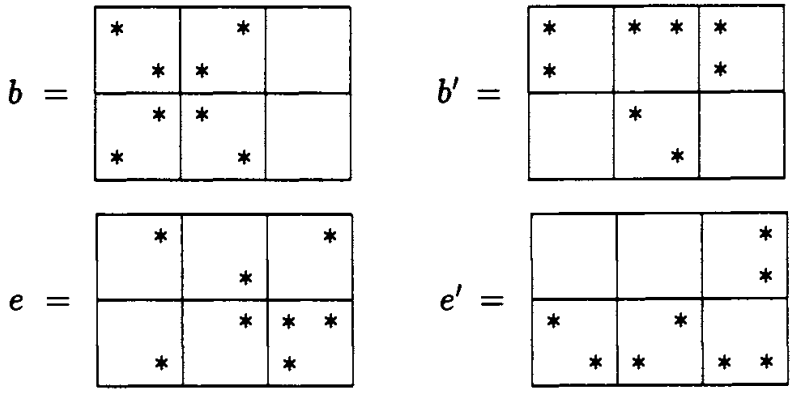
$e =$  $e' =$ 

and we used MAGMA to verify that this generating set is isomorphic to Type 2.

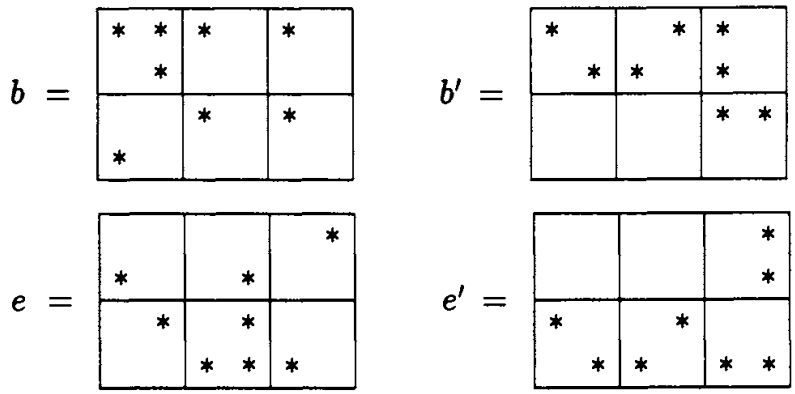
Orbit 6. We calculate the sextets $\mathcal{S}, \mathcal{S}'$ containing e, e' and we find that there are two choices for e and one choice for e'.



The first choice leads to:

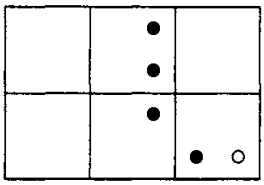
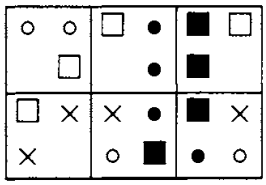


We used MAGMA to verify that this generating set is isomorphic to Type 2. The second choice leads to:

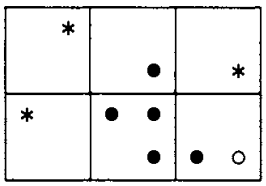
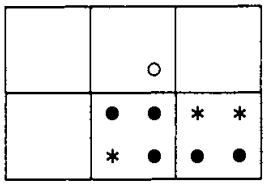


and since all these octads are orthogonal to w_4 this is not a generating set.

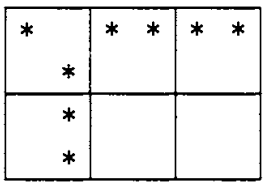
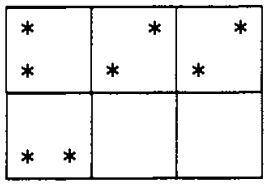
Orbit 7. The situation here is exactly the same as for Orbit 4. The only choice is $e = d + d'$, which violates linear independence of the generators.

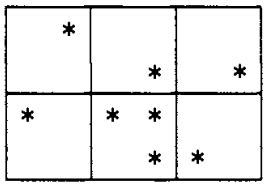
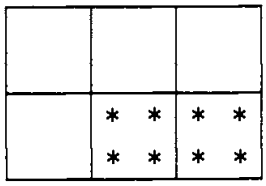
$e =$  $\rightsquigarrow$ 

Orbit 8. If the generating set has not been previously considered, then e, e' must contain both points from orbit 8. Thus we have

$e =$  $e' =$ 

which leads to:

$b =$  $b' =$ 

$e =$  $e' =$ 

Since $w_2 + w_4$ is orthogonal to all the above octads, this is not a generating set.

5. A taxonomy of tail-biting generating sets

The analysis provided in the foregoing section implies that there are at most three distinct isomorphism types of generating sets for the Golay code that have the structure of (1). These are Types 2, 5 and 7. In this section, we characterize the symmetry group of each of these three generating sets. Automorphisms in M_{24} that permute the subcodes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ will be given as permutations with respect to the following labeling of the MOG array:

1	5	9	13	17	21
2	6	10	14	18	22
3	7	11	15	19	23
4	8	12	16	20	24

Furthermore, we investigate which of the three generating sets may be refined to produce the structure of (2). We will conclude that such a generating set is *unique*, up to the symmetries discussed at the beginning of this paper.

To find out whether a given generating set that complies with (1) may be further refined to comply with (2), we study the projections of the subcodes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ onto the four positions labeled $\boxed{1}$, $\boxed{2}$, $\cdots$, $\boxed{6}$. As noted in the foregoing section, such projections may have one of three possible weight distributions:

Type I:	0, 2, 2, 4
Type II:	0, 1, 3, 4
Type III:	0, 2, 3, 3

Thus a generating set may be associated with the 6×6 matrix of types obtained by noting the weight distribution of the projection of each subcode $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ onto each tetrad that consist of points which receive the same coordinate label. Observe that the matrix of types is not an invariant, since changing the pairings in a generating set may also change the matrix of types.

Type 2. This is the generating set that was exhibited at the beginning of § 3. The coordinate labeling and the matrix of types are as follows:

3	3	2	5	4	4
2	1	2	6	5	5
1	1	2	6	5	4
3	3	1	4	6	6

	$\boxed{1}$	$\boxed{2}$	$\boxed{3}$	$\boxed{4}$	$\boxed{5}$	$\boxed{6}$
$\mathcal{A}$	III	II	I			
$\mathcal{B}$		III	III	III		
$\mathcal{C}$			I	II	III	
$\mathcal{D}$				III	III	III
$\mathcal{E}$	III				III	I
$\mathcal{F}$	III	III				III

The symmetry group is the subgroup of M_{24} that permutes the subcodes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ among themselves. It has order 2, and is generated by:

(1,5)(2,23)(3,18)(4,8)(6,22)(7,19)(9,16)(10,21)(11,17)(12,13)(14,20)(15,24)

This generating set cannot be refined to produce the structure of (2). This would require an octad in $\mathcal{E}$ with two points labeled $\boxed{5}$, and a second octad in $\mathcal{E}$ with two points labeled $\boxed{1}$. The only octad in $\mathcal{E}$ with either property is e.

Type 5. This is the generating set that was exhibited at the beginning of §4.2. The coordinate labeling and the matrix of types are as follows:

3	1	2	5	4	5
1	3	3	6	5	4
1	2	1	6	4	5
3	2	2	4	6	6

	$\boxed{1}$	$\boxed{2}$	$\boxed{3}$	$\boxed{4}$	$\boxed{5}$	$\boxed{6}$
$\mathcal{A}$	III	I	III			
$\mathcal{B}$		III	III	III		
$\mathcal{C}$			III	I	III	
$\mathcal{D}$				III	III	III
$\mathcal{E}$	III				III	I
$\mathcal{F}$	III	III				III

The symmetry group of this generating set is the subgroup of M_{24} that permutes the subcodes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ among themselves. This group has order 6, acts

as the symmetric group S_3 on the three sextets $(\mathcal{A},\mathcal{D})$, $(\mathcal{C},\mathcal{F})$, and $(\mathcal{B},\mathcal{E})$. The group is generated by the following permutations:

(1,2)(3,4)(5,10)(6,11)(7,9)(8,12)(13,18)(14,19)(15,17)(16,20)(21,23)(22,24)

(1,21)(2,5)(3,11)(4,13)(6,18)(7,14)(8,15)(9,20)(10,23)(12,24)(16,19)(17,22)

This generating set also cannot be refined to produce the structure of (2). Again, the only octad in $\mathcal{E}$ with two points labeled $\boxed{5}$ or two points labeled $\boxed{1}$ is e, whereas we need two $\mathcal{E}$-octads with this property. Similarly, for $\mathcal{A}$ and $\mathcal{C}$.

Type 7. This is the isomorphism class of all the other generating sets found in this paper, namely Types 1, 3, 4, 6, and 7. The coordinate labeling and the matrix of types are as follows:

3 1	1 5	4 5				
1 3	3 6	5 5				
2 2	1 6	4 4				
3 2	2 4	6 6				

	$\boxed{1}$	$\boxed{2}$	$\boxed{3}$	$\boxed{4}$	$\boxed{5}$	$\boxed{6}$
$\mathcal{A}$	III	III	III			
$\mathcal{B}$		III	III	III		
$\mathcal{C}$			III	III	III	
$\mathcal{D}$				III	III	III
$\mathcal{E}$	III				III	III
$\mathcal{F}$	III	III				III

The subgroup of M_{24} that permutes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ among themselves has order 6, acts as the symmetric group S_3 on the three sextets, and is generated by

$\sigma_1 = (1,24)(2,7)(3,11)(4,20)(5,8)(6,15)(9,12)(10,14)(13,23)(16,18)(17,21)(19,22)$

$\sigma_2 = (1,19)(2,20)(3,18)(4,17)(5,24)(6,23)(7,21)(8,22)(9,15)(10,16)(11,14)(12,13)$

Unlike the pervious cases, it is possible to refine this generating set to the 12-section structure of (2). In fact, there is a unique way to do so. The corresponding labeling of the MOG and the resulting set of generators are shown below.

$3'$ $1'$	1 5	4 $5'$				
$1'$ 3	3 $6'$	5 $5'$				
$2'$ 2	1 $6'$	4 $4'$				
$3'$ 2	$2'$ $4'$	6 6				

	11	$1'1'$	22	$2'2'$	33	$3'3'$	44	$4'4'$	55	$5'5'$	66	$6'6'$
a'	11	01	11	01	11	00	00	00	00	00	00	00
$a+a'$	00	11	11	10	10	11	00	00	00	00	00	00
b'	00	00	11	01	01	11	11	00	00	00	00	00
b	00	00	00	11	10	11	01	11	00	00	00	00
c	00	00	00	00	11	10	11	01	11	00	00	00
c'	00	00	00	00	00	11	11	10	10	11	00	00
d'	00	00	00	00	00	00	11	01	01	11	11	00
d	00	00	00	00	00	00	00	11	10	11	01	11
e'	11	00	00	00	00	00	00	00	11	01	11	01
e	10	11	00	00	00	00	00	00	00	11	11	10
f_1+f'	01	11	11	00	00	00	00	00	00	00	11	01
f_1	10	11	01	11	00	00	00	00	00	00	00	11

Observe that σ_1 acts as $(a',f_1)(a+a',f_1+f')(b',e)(b,e')(c,d)(c',d')$ on these generators, whereas σ_2 acts as $(a',d)(a+a',d')(b',c')(b,c)(e,f_1+f')(e',f_1)$.

References

[1] A.R. CALDERBANK, G.D. FORNEY, JR., AND A. VARDY, Minimal tail-biting trellises: the Golay code and more, submitted to the *IEEE Trans. Inform. Theory*, August 1997.

[2] J.H. CONWAY AND N.J.A. SLOANE, *Sphere Packings, Lattices and Groups*, New York: Springer-Verlag, 1993.

[3] G.D. FORNEY, JR., Coset codes — Part II: Binary lattices and related codes, *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, 1988.

[4] ———, The forward-backward algorithm, in *Proc. 34th Allerton Conference on Comm., Control, and Computing*, Monticello, IL., pp. 432–446, October 1996.

[5] R. KÖTTER AND A. VARDY, Construction of minimal tail-biting trellises, in *Proc. IEEE Int. Workshop on Inform. Theory*, pp. 73–75, Killarney, Ireland, June 1998.

[6] ———, The theory of tail-biting trellises, in preparation, 1998.

[7] B.D. KUDRYASHOV AND T.G. ZAKHAROVA, Block codes from convolutional codes, *Problemy Peredachi Informatsii*, vol. 25, pp. 98–102, 1989.

[8] S. LIN AND D.J. COSTELLO, JR., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

[9] J.H. MA AND J.K. WOLF, On tail-biting convolutional codes, *IEEE Trans. Commun.*, vol. 34, pp. 104–111, 1986.

[10] D.J. MUDER, Minimal trellises for block codes, *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, 1988.

[11] G. SOLOMON AND H.C.A. VAN TILBORG, A connection between block and convolutional codes, *SIAM J. Applied Math.*, vol. 37, pp. 358–369, 1979.

[12] N. WIBERG, H.-A. LOELIGER, AND R. KÖTTER, Codes and iterative decoding on general graphs, *Euro. Trans. Telecommun.*, vol. 6, pp. 513–526, 1995.

On the Weight Distribution of Some Turbo-Like Codes

Daniel J. Costello, Jr.

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
costello@saturn.ee.nd.edu

Jiali He

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
jhe@chaucer.helios.nd.edu

Abstract. Parallel concatenated turbo codes are capable of near-capacity performance at moderate bit-error-rates, while serial concatenated turbo codes are capable of almost error-free performance at higher signal-to-noise ratios. It is desirable to have codes that achieve a combination of these characteristics. In attempting to lower the "error floor" of parallel concatenated turbo codes, we consider several hybrid concatenation schemes with parallel concatenated inner turbo codes serially concatenated with outer block and convolutional codes. The objective is for the outer code to "filter out" the bad input sequences for the inner turbo code and for the outer decoder to collaborate in iterative decoding with the inner turbo decoder. As one example, we consider using a rate $^3/_4$ outer convolutional code and a rate $^2/_3$ inner turbo code in an overall rate $^1/_2$ hybrid concatenation scheme, and we obtain the exact weight distribution of this scheme for some typical interleavers and very short block lengths and compare these results to conventional rate $^1/_2$ turbo codes. The hybrid scheme clearly has a superior weight distribution. We also consider another very interesting turbo-like coding scheme, a serial concatenation of three convolutional codes, and obtain the weight distribution of this scheme with an overall rate of $^1/_4$, for some typical interleavers and very short block lengths. It is again clear that this code has a superior weight distribution compared to both a parallel concatenation of three convolutional codes and a serial concatenation of two convolutional codes with overall rate $^1/_4$. Because of their exceptional weight distributions, these schemes have the potential to outperform conventional turbo codes.

1. Introduction

Four years ago, a new class of codes, called turbo codes, was introduced [7] and shown to achieve near-capacity performance at moderate bit-error-rates (BERs). Since then these codes have received a great deal of attention from the research

This work was supported in part by NSF Grant NCR95-22939, NASA Grant NAG 5-557, and by the Lockheed-Martin Corporation.

community. Computer simulations by various researchers have confirmed the claimed performance of achieving a BER of 10^{-5} with a rate $1/2$ code using BPSK modulation on an additive white Gaussian noise (AWGN) channel at a signal-to-noise ratio (SNR) of 0.7 dB, which is only 0.5 dB away from channel capacity. However this performance is only achieved with a very large block size, on the order of 65,000 bits, and with as many as eighteen iterations of decoding. Also, there is an *error floor* at BER's below 10^{-5}, which causes performance to degrade dramatically. In this paper, we consider several approaches to overcoming these drawbacks.

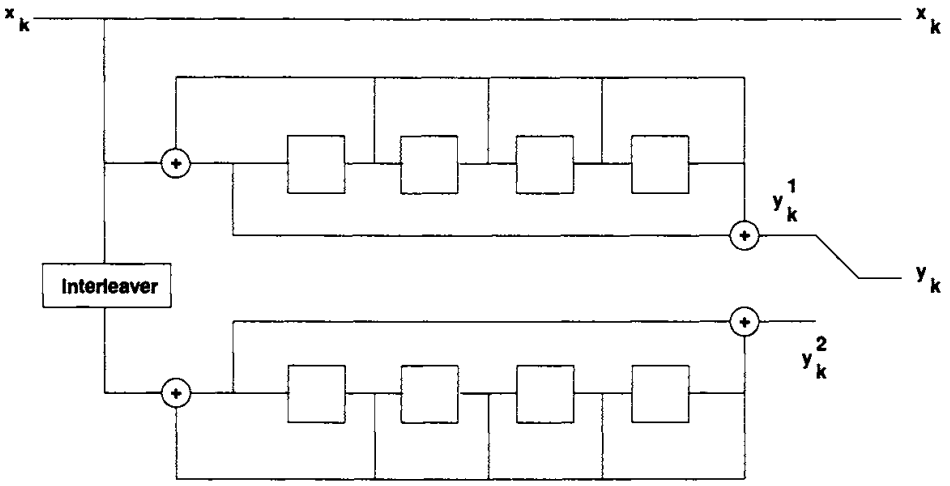


Figure 1. *Turbo encoder*

The turbo code presented in the original paper [7] consists of a parallel concatenation of two recursive (37,21) convolutional codes (see Figure 1), where 37 and 21 are the octal representations of the feedback and the feedforward connections, respectively, of the encoder. Appropriate puncturing is applied to make the overall code rate $1/2$. The random interleaver and the recursive property of the component codes (or constituent codes) are very critical to the outstanding performance achieved by turbo codes. The interleaver insures, with high probability, that the parity sequence $\{y_k^2\}$ generated by the second encoder is different from the parity sequence $\{y_k^1\}$ generated by the first encoder, thus making it unlikely that both encoders will generate low weight parity sequences in response to a particular information sequence. The recursive component code, on the other hand, insures that the weight of the parity sequence corresponding to a very low weight information sequence, say, a weight one information sequence, is very high. This is very important since a weight-one information sequence is still a weight-one sequence after interleaving. Both the interleaver and the recursive code function together in a process called *spectral thinning*, which explains [18] the exceptional performance of turbo codes at low SNR's.

Shannon's noisy channel coding theorem suggests that capacity achieving performance can be approached by randomly constructing long block codes. Turbo codes are long block codes, considering the fact that in the original paper the

interleaver size was 65536, and are also somewhat "random" due to the random interleaver. However, there is no known method to decode a randomly constructed code. The main contribution of turbo codes is *turbo decoding*, which is an iterative decoding process. Two decoders, which employ MAP decoding [3] to give reliability information about each decoded bit, exchange so-called "extrinsic information" through the iterative decoding process. The performance typically improves after each decoding iteration until it reaches a limit, which approximates the performance of the code with maximum-likelihood decoding.

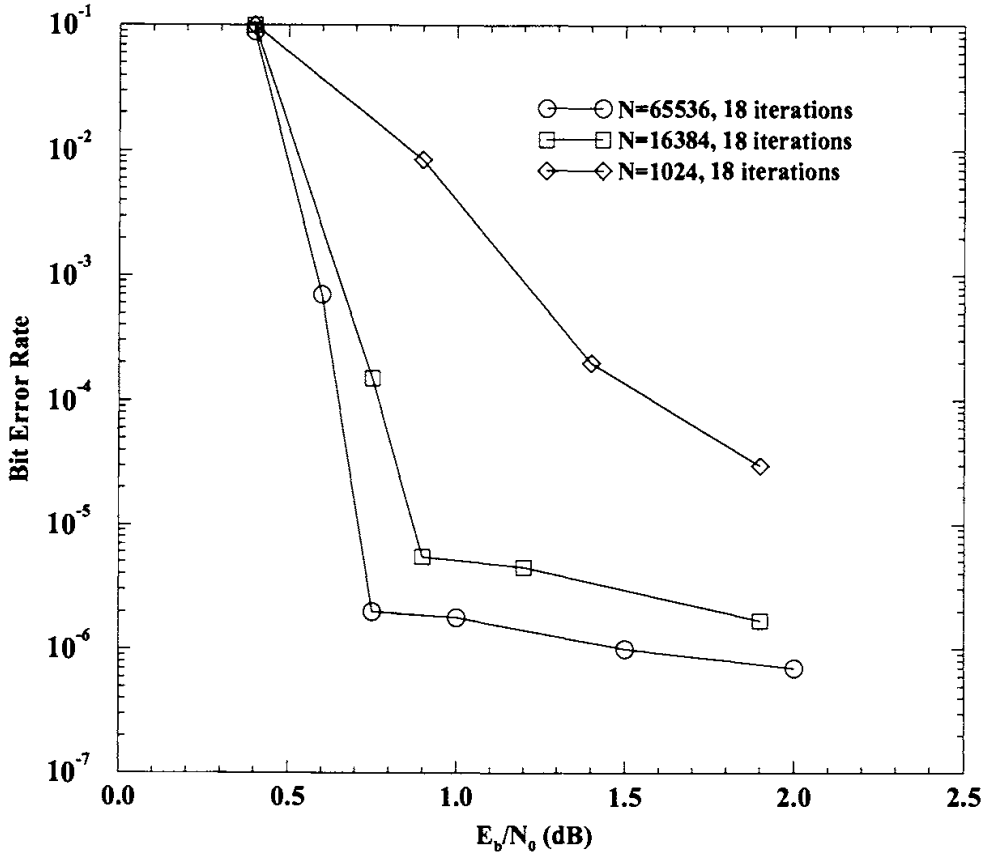


Figure 2. *Performance of turbo codes*

In the four years since the appearance of turbo codes, much research has been done on the structure and performance of this coding scheme. The good performance of turbo codes is explained fairly well by examining their "thin" distance spectrum [**18**], and the code performance at moderate and high SNR's is estimated using union bounds on the BER assuming uniform interleaver and maximum-likelihood decoding [**6**]. Also, the turbo code concept has been extended in a variety of ways: block codes have been used as component codes [**6, 11**]; different codes have been used for each of the component codes [**19**]; high rate convolutional codes have been used as component codes [**5, 15**]; primitive polynomials have been used as the encoding feedback polynomial to improve code performance at moderate to high SNR's in [**19, 18**]; and multiple turbo codes, which have more than two component codes, have been investigated and corresponding decoding strategies given in [**10**]. Still

more recently, a natural extension of turbo codes, serially concatenated turbo codes, have been proposed as a valid, and in some cases superior, alternative to parallel concatenated turbo codes [4]. Although this is not a wholly new coding scheme, it can only achieve superior performance with interleaving and iterative decoding similar to turbo decoding.

Simulations have revealed a characteristic break in the performance curve of parallel concatenated turbo codes. At low SNR's, the curve is very steep, while at higher SNR's, the curve flattens out (see Figure 2). The flat part of the curve has been called the *error floor*. It is caused by a few codewords with very small weight. It was shown in [**19, 20**] that, for a particular interleaver of size 65536 and the $(37, 21)$ component code, the turbo code only has a minimum distance of 6. This can be seen from an example.

Example. Consider the (37,21) component code. The generator polynomial for this code is given by:

$$\left[1, \frac{1+D^4}{1+D+D^2+D^3+D^4}\right]$$

A weight-two input sequence, such as $\cdots 01000010 \cdots = D^i(1+D^5)$ is self terminating. It will generate the codeword $[D^i(1+D^5), D^i(1+D+D^4+D^5)]$ so that the parity sequence is $\cdots 01100110 \cdots$. Now, an interleaver such as the one shown in Figure 3 will map the information sequence to $D^j(1+D^5)$. This interleaved sequence will also produce a parity sequence of weight four, namely: $D^j(1+D+D^4+D^5)$. Puncturing with the puncturing matrix

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

generates a codeword with weight only six. Fortunately, for a random interleaver, input sequences of the form $D^i(1+D^5)$ are permuted into sequences which are not of the form $D^j(1+D^5)$ with very high probability.

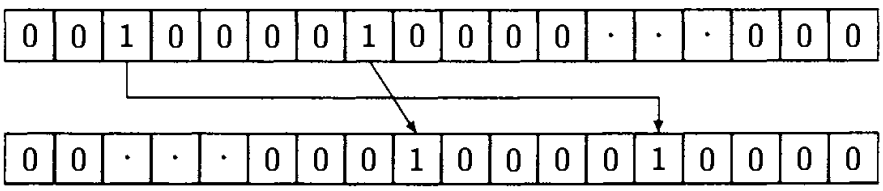


Figure 3. *Example of bad interleaver mapping leading to a small distance*

Thus, although turbo codes have some very low weight codewords, the multiplicity of these low weight codewords is much smaller than, say, in a block code with same block length obtained by terminating a conventional convolutional code. This is the reason for the remarkable performance of turbo codes. □

Various schemes with outer block codes and inner turbo codes are presented in § 2. In § 3, we propose a new hybrid coding scheme with a parallel concatenated inner turbo code and a serially concatenated outer convolutional code. Exact weight distributions of this coding scheme for short block lengths are given and compared to those of standard turbo codes with the same overall rate. In § 4 we propose a new triple serial concatenation scheme with three convolutional codes and two interleavers. Exact weight distributions for short block lengths are again calculated and compared to the weight distributions of triple parallel concatenation and double serial concatenation schemes of the same overall rate. Several possible decoding structures are also proposed.

2. Concatenated turbo codes with outer block codes

Despite their outstanding performance, several problems still exist with turbo codes. One is the existence of an error floor. Others include the fact that near capacity performance requires considerable decoding effort and a long decoding delay. As a result, we propose various serial concatenation schemes using a parallel concatenated turbo code as the inner code with the following goals in mind: (1) eliminate (or reduce) the error floor; (2) achieve similar performance with less decoding effort; and (3) achieve similar performance with less decoding delay. Performance results are summarized in this section.

The problem of the error floor has been tackled by several different methods in the literature. The first was an attempt to improve the interleaver [**19**]. As was shown in the previous section, the low weight codewords are due to *bad mappings* in the interleaver. This approach tries to reduce the number of bad mappings in a particular interleaver: it examines the interleaver and break up mappings like those shown in Figure 3. Due to computational limitations, such improvements can only be applied to very low weight information sequences, and sometimes the breaking up of a bad mapping may result in an even worse mapping. But after several such operations the interleaver usually becomes better, and the error floor is lowered. In [**2**], it was shown that an improved 400 bit interleaver is better than an unimproved interleaver of size 1000 for SNR's above 2 dB.

A similar method was proposed in [**16**]. Due to the bad mappings in a particular interleaver, some positions are more prone to errors than others. The study of [**2**] shows an example of this phenomenon. The method of [**16**] examines the interleaver, determines the error-prone positions, and then uses a BCH code to give these positions additional protection. Compared to the first method, this one is less elegant because it adds more complexity to the encoder and decoder without achieving better performance. Both methods lower the error floor to some degree, but it still exists at lower BER's.

Another similar method was proposed in [**17**], where the error-prone positions are identified by a distance spectrum analysis. Rather than protecting these positions with an outer BCH code, as in [**16**], in [**17**] the error-prone positions are filled with dummy bits instead of information bits, and the dummy bits are

then discarded at the receiver. For large interleaver sizes, the resulting rate reduction is very slight, and a similar lowering of the error floor is achieved.

As claimed in [15], when an improved interleaver is used, the error distribution is almost uniform, so it is impossible to see that some positions are more prone to errors than others by using simulation. In this case it makes more sense to use an outer code to protect the whole block. Reed-Solomon (RS) codes have been used [9], for example, and BCH codes can also be used [1]. A reason for preferring RS codes is if we want to use feedback from the outer code to help the inner code. For an RS code, the decoding error probability can be made extremely small and this can provide very reliable feedback [14].

As explained in the previous section, the presence of an error floor is due to the low free distance of turbo codes. Because of this, a small number of errors, typically an even number (most likely, two) in a block, often "fall through" the iterative decoding process and are not corrected. With an outer (255,239) Reed-Solomon code, for example, the error floor can be lowered, while the code rate only decreases from 0.50 to 0.47. This is due to combining a code with a good overall distance spectrum (the turbo code) with a code which is optimized for minimum distance (the RS code).

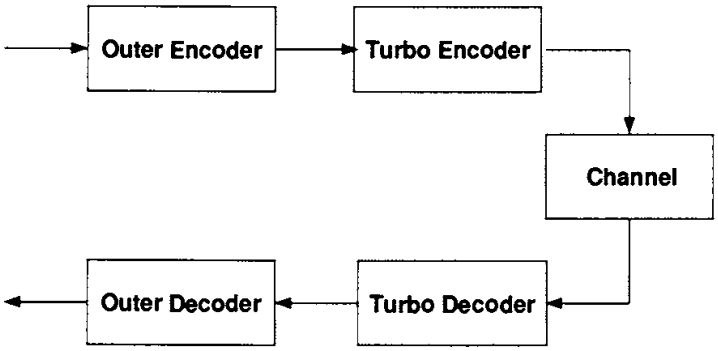


Figure 4. *Turbo code extended with an outer code*

A concatenation scheme with an outer RS code and an inner turbo code is illustrated in Figure 4. The information enters the RS encoder first, and then the RS codeword is encoded by the turbo encoder. The turbo code block size is the length of one RS codeword in bits. For the (255, 239) RS code, the block size is 2040. At the receiver, a codeword is first decoded by the turbo decoder and then by the RS decoder. The RS decoder takes the output of the turbo decoder and, after regrouping the output bits into symbols, corrects the residual errors in the decoded turbo block, thus greatly reducing the error floor. To obtain an estimate of the BER at the RS decoder output, we use the following bound:

$$P_b < \frac{2^{\tilde{K}-1}}{2^{\tilde{K}}-1} \sum_{j=t+1}^{n} \frac{j+t}{n} \binom{n}{j} p_s^j (1-p_s)^{n-j}$$

for a t-error-correcting RS code, where $\tilde{K}$ is the number of bits per RS code symbol and n is the RS code blocklength in symbols. Turbo code simulation

results are used to estimate the symbol error rate p_s at the RS decoder input. The estimated BER of this concatenation system indicates a substantial lowering of the error floor to about 10^{-16}. For more details, see [9].

To further improve the performance of this concatenation scheme, an intra-code bit interleaver can be placed between the two codes, since the distribution of the erroneously decoded bits within a turbo code block has some structure. For example, if there is an error at position k, there is a high probability of another error at position $k+j^*n_{\text{cycle}}$, where n_{cycle} is the cycle length of the recursive convolutional code. The intra-code bit interleaver takes advantage of the distribution of erroneously decoded bits within a turbo code block by placing positions that are the same modulo n_{cycle} in the same RS code symbol, thus decreasing the symbol error rate at the input to the RS decoder. Results show that with an intra-code bit interleaver, there is about a 0.2 dB performance improvement at BER's below the error floor with no increase in decoding complexity [9].

The formula used above to estimate the BER of the turbo/RS concatenation scheme assumes that the symbol errors are random, that is, it assumes ideal symbol interleaving between the inner and outer codes, which, unlike the intra-code bit interleaver above, adds to the decoding delay. Unfortunately, turbo code simulations show that the symbol errors are not randomly distributed, i.e., most decoded blocks are error free, many blocks have a small number of symbol errors, and a few blocks have many symbol errors (see Table 1). This makes the results obtained using this formula somewhat suspect.

#E:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
#B:	19041	401	738	202	168	81	45	42	26	18	17	16	13	15	12
#E:	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
#B:	12	6	6	9	8	7	6	3	5	7	5	3	6	3	3
#E:	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
#B:	1	1	4	1	0	4	2	2	4	3	3	2	2	0	4
#E:	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
#B:	0	3	1	0	0	2	2	0	0	4	2	1	1	1	0

Table 1. *Example of symbol error distribution*

Here #E denotes the number of symbol errors, while #B denotes the number of blocks. The results correspont to SNR of 1.2 dB, five decoding iterations, with a block size of 2040, simulated for 21000 blocks.

In Table 1, the blocks which have more than 8 symbol errors cannot be corrected by the (255,239) RS code. Thus a symbol iterleaver is required to spread the symbol errors in one block into several blocks, thus generating more correctable blocks. Simulations show that, for a block size of 2040, the performance of

the concatenated system improves significantly with a moderate depth symbol interleaver at moderate to high channel SNR's. For example, for the case shown in Table 1 (SNR of 1.2 dB), when a depth-15 symbol interleaver is used, a significant number of additional blocks can be corrected by the outer RS code (see Table 2). At channel SNR's above 1.4 dB, a depth-5 symbol interleaver is sufficient. When the channel SNR is lower, however, a large interleaving depth is necessary to achieve the same performance obtained under the ideal symbol interleaving assumption. The required interleaving depth can be reduced somewhat if we use reliability information from the turbo decoder to transform some symbol errors into erasures (errors and erasures decoding), at a cost of additional decoding complexity, or if we use an outer (255, 223) RS code, at a cost of additional rate loss.

#E:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
#B:	15042	3596	1146	484	273	169	123	82	36	23	15	6	1	1	1

#E:	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
#B:	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2. *Example of symbol error distribution improvement with depth-15 symbol interleaving*

Again #E denotes the number of symbol errors, while #B denotes the number of blocks. The results correspont to SNR of 1.2 dB, five decoding iterations, with a block size of 2040, simulated for 21000 blocks.

From another viewpoint, symbol interleaving introduces delay, and the delay increases with interleaving depth. On the other hand, the performance of turbo codes improves with block size, which also increases delay. As far as delay is concerned, a symbol interleaving depth of 5 is equivalent to one turbo code block with block size of 10200 containing five RS codewords. Simulations show that the scheme with a larger turbo code block is much better than the scheme with symbol interleaving, at least at low SNR's. (Note that in a standard concatenation scheme with an inner convolutional code and an outer RS code, a symbol interleaver is also used between the inner and outer codes. But for convolutional codes, unlike turbo codes, there is no way to improve performance by increasing the block size without also increasing decoder complexity.) Thus the value of symbol interleaving to improve the performance of a turbo/RS concatenation scheme is questionable.

The performance of the turbo/RS concatenation scheme can be further improved by using feedback from the RS decoder to the turbo decoder. With feedback from the outer code, we obtain better performance with the same number of turbo decoding iterations, or similar performance with fewer turbo decoding iterations. Different rate RS codes can be used in the feedback scheme. For example, with two RS code rates, after a few iterations of turbo decoding, the strongest (lowest rate) RS codes are decoded and those bits are fed back to

assist subsequent iterations of turbo decoding; after some additional iterations of turbo decoding, the same process is repeated for the weaker (higher rate) RS codes (see Figure 5). There are different ways to use the feedback information in turbo decoding [**15**]. One is to modify the soft channel output values of those bits in the received information sequence that have been decoded by the RS code. Another is to modify the extrinsic information used as inputs for each iteration of decoding. To give positive results, the feedback bits must have high reliability. Furthermore, two encoding strategies can be employed: (1) only a small fraction of the information bits in a block are encoded using one low rate RS code (or several low rate codes with different rates) and then interleaved into the information sequence prior to turbo encoding; and (2) all the information bits in a block are encoded using one high rate RS code (or several high rate codes with different rates) and then interleaved and turbo encoded. In the first case, the rate loss is very small. The main purpose is to reduce the number of iterations of turbo decoding or to achieve better performance with the same number of iterations, and some lowering of the error floor may also be achieved. This scheme can be quite natural in applications where some bits in a block need extra protection, such as a compressed image transmission system where the header information is very important. In the second case, the number of iterations of turbo decoding can also be reduced. Furthermore, after turbo decoding, the outer code can pick up most of the residual errors and lower the error floor significantly. But the rate loss is higher.

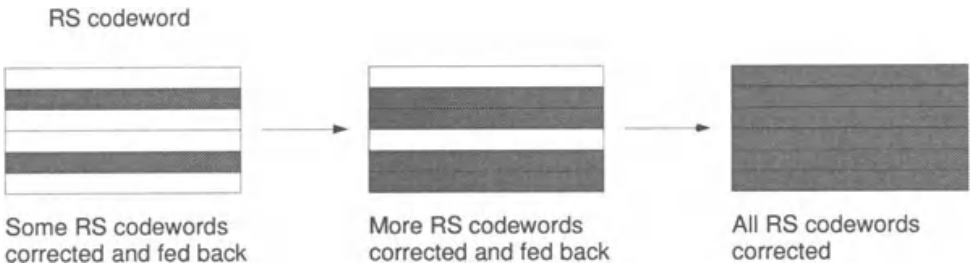


Figure 5. *Iterative decoding scheme with feedback from outer RS codes*

In the compressed image transmission example [**12**], we tested the performance improvement using the first approach mentioned above. Results show that at a channel SNR of 0.9 dB (namely E_p/N_0, where E_p is the energy per pixel), after ten iterations, the BER is $8.9 \cdot 10^{-6}$ with feedback back versus BER of $2.9 \cdot 10^{-5}$ without feedback, an improvement factor of more than three. This improvement is achieved with almost no added decoding complexity.

A comparison of a non-feedback scheme, with two RS(255,239) codes covering all the information bits in one turbo block, and a feedback scheme with a single RS(255,223) code covering roughly half the information bits in one turbo block (both schemes have the same block length of 4080 and the same number 3824 of information bits) shows that at lower SNR's the scheme with the stronger RS code and feedback is better. This further confirms the effectiveness of using feedback from the outer code.

3. A new hybrid coding scheme

As discussed in the previous section, the scheme with feedback from the outer code to aid the turbo decoder, similar to the "state pinning" idea of Collins and Hizlan [8] in conventional concatenated coding, is quite effective. However, this scheme still has several disadvantages: (1) the outer decoder begins decoding only after the inner turbo code has finished several decoding iterations, i.e., most of the time, the outer decoder is idle; (2) if the outer decoder uses a hard-decision decoding algorithm, or errors and erasures decoding, the soft information from the inner decoder output is not fully utilized; and (3) the turbo decoder output is bursty, so that the number of errors in a block is often beyond the error correcting capability of the outer code. Also, when the outer code cannot correct the block, we cannot use feedback from the outer code to aid the inner decoder. Therefore it is desirable to have a concatenation scheme in which the outer decoder fully exploits the soft information from the inner decoder and provides feedback to the inner decoder throughout the entire iterative decoding process. Convolutional outer codes can make full use of the soft information from the inner decoder and can interact fully with the inner decoder through an iterative decoding process. This leads to a hybrid coding structure consisting of both serial and parallel concatenation.

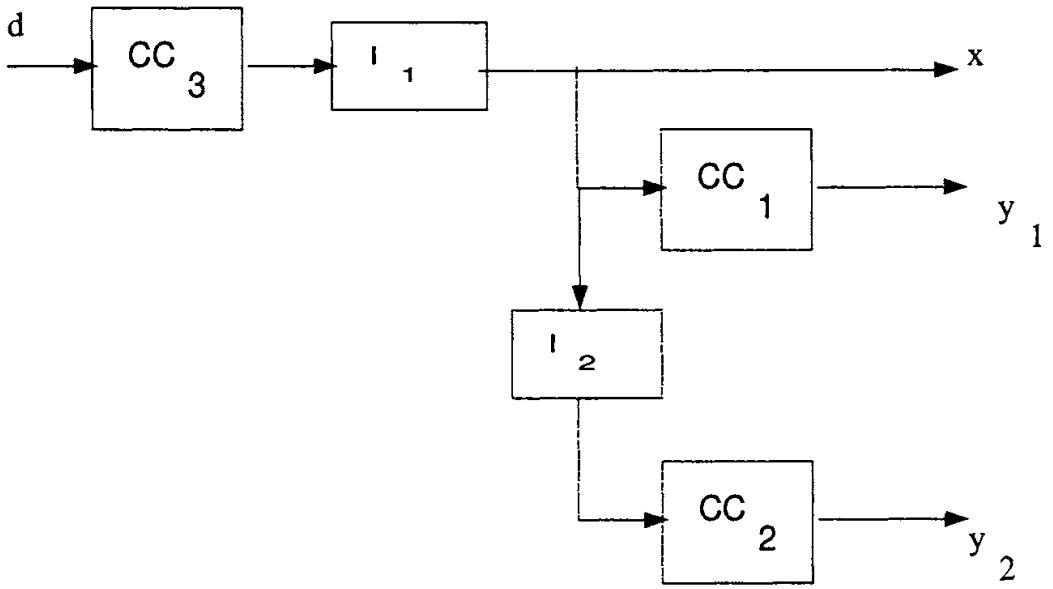


Figure 6. *A new hybrid encoding structure*

It is well known that parallel concatenated turbo codes are designed for near-capacity performance at moderate BER's, while serial concatenated turbo codes are designed for almost error-free performance at higher SNR's. Thus a proper combination of serial and parallel concatenation should maintain the advantages of both. A new encoding structure for such a hybrid coding scheme is illustrated in Figure 6. The interleaver I_1 between the outer convolutional code and the

inner turbo code is more important for decoding than for encoding. The outer code "filters out" the bad input sequences for the inner code, thus giving the overall code a large minimum distance.

Weight	Multiplicity
0	1
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	3
11	6
12	14
13	60
14	150
15	452
16	1356
17	3507
18	8839
19	21247
20	48161
⋮	⋮

Table 3. *Weight distribution of the new hybrid coding scheme*

Weight	Multiplicity
0	1
1	0
2	0
3	0
4	0
5	0
6	1
7	0
8	14
9	19
10	53
11	142
12	364
13	943
14	2488
15	6458
16	15889
17	37458
18	82773
19	174154
20	345741
⋮	⋮

Table 4. *Weight distribution of a standard rate $1/2$ turbo code*

We have calculated the weight distribution of this hybrid coding scheme for very short block lengths and compared it to the weight distribution of a standard turbo code with the same overall code rate and similar decoding complexity. Using a rate $3/4$ outer convolutional code and a rate $2/3$ inner turbo code as an example, we obtained the weight distribution in Table 3 for block length $N = 27$. The outer code has 8-states and is non-recursive, with generator matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1+D & D & 1 \\ 0 & D & 1+D^2 & 1+D^2 \end{bmatrix}$$

See [**13**] for more details on this code. The turbo code is the primitive (23,35) code with additional puncturing. The overall code rate is $1/2$.

The weight distribution of a standard rate $\frac{1}{2}$ parallel concatenated turbo code with $N = 27$ is shown in Table 4. The component code is the primitive (23,35) code. The weight distributions in these tables are "averaged" over several random interleavers. However, for the random interleavers tested, the weight distributions are quite similar. The distributions shown are typical of those tested.

As can be seen from a comparison of the two weight distributions, the hybrid structure has a "thinner" distance spectrum and a larger minimum distance, as expected, since the outer code "filters out" the bad input sequences of the inner turbo code, i.e., the input sequences which produce low weight codewords. It is expected that, at longer block lengths, this hybrid coding structure will have the characteristics of both parallel concatenation (a "thin" overall distance spectrum) and serial concatenation (a large minimum distance).

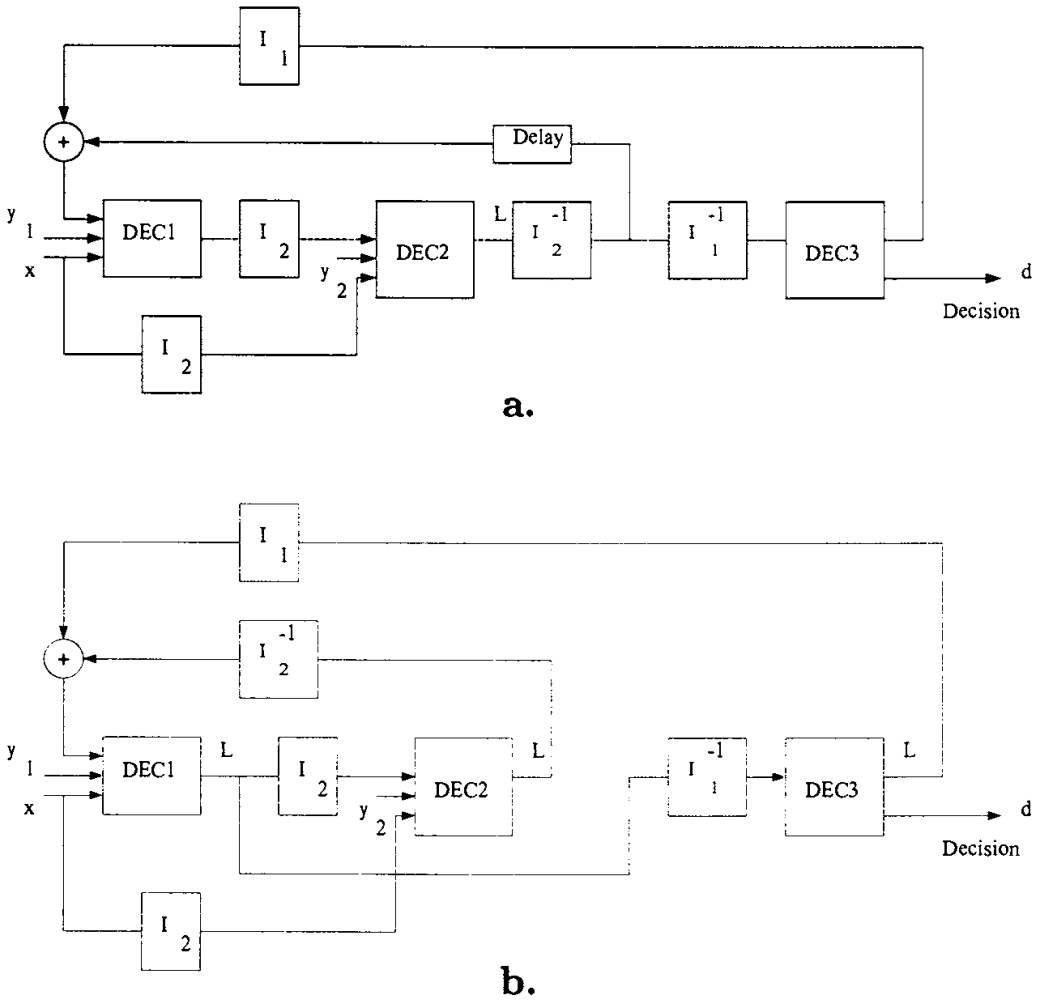


Figure 7. *Two possible decoding structures for the new hybrid coding scheme*

Two possible decoding structures based on the turbo decoding principle, or the iterative decoding principle, are illustrated in Figures 7a and 7b. The difference is that in the decoding structure of Figure 7b, the units labeled DEC2 and DEC3 can work simultaneously, thus resulting in less overall decoding delay.

4. Triple serial concatenation

Parallel concatenated turbo codes have been extended to more than two component codes. These schemes are known as *multiple turbo codes*. Different decoding strategies are given in [**15**] and [**10**]. Serial concatenated turbo codes can also have more than two component codes. Due to the serial structure, a very large minimum distance can be expected from this coding structure, and thus these codes may be very useful in extremely low BER applications. The encoding structure for a triple serial concatenation scheme is shown in Figure 8. There are three codes and two interleavers in this scheme, and in this case the interleavers are important for both encoding and decoding. Due to the interleavers, it is not easy to terminate all three component codes at the same time, so we terminate the outer code only.

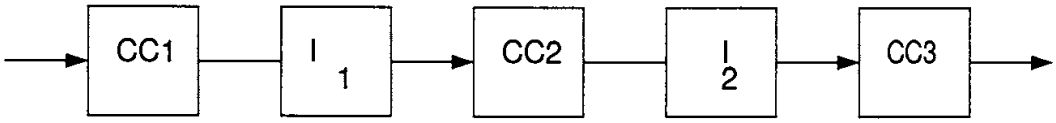


Figure 8. *A triple concatenation encoding structure*

As for the hybrid scheme discussed in the previous section, the weight distribution of triple serial concatenation is calculated for very short block lengths and compared to the weight distributions of a parallel concatenation of three codes and a serial concatenation of two codes with the same overall code rate and similar decoding complexity. The weight distributions are shown in Tables 5, 6, and 7. In each case, the block length $N = 27$ and the overall code rate is $1/4$. For the serial concatenation of three codes, the component codes have rates $1/2$, $2/3$, and $3/4$, and numbers of states 4, 4 and 8, respectively. All three codes are non-recursive, with the following generator matrices:

$$\begin{bmatrix} 1+D^2 & 1+D+D^2 \end{bmatrix}$$

$$\begin{bmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1+D & D & 1 \\ 0 & D & 1+D^2 & 1+D^2 \end{bmatrix}$$

See [**13**] for more details on these codes. For the parallel concatenation of three codes, the component codes are the same rate $1/2$, 16-state, primitive (23,35) code. For the serial concatenation of two codes, the component codes are the same rate $1/2$, 8-state, non-recursive code, with generator matrix

$$\begin{bmatrix} 1+D+D^3 & 1+D^2+D^3 \end{bmatrix}$$

Further information on this code can be also found in [**13**]. The weight distributions shown here are, again, typical of several random interleavers.

Weight	Multiplicity
0	1
1	0
2	0
⋮	⋮
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	2
29	2
30	10
31	18
⋮	⋮

Table 5.
Weight distribution of a serial concatenation of three convolutional codes

Weight	Multiplicity
0	1
1	0
2	0
⋮	⋮
16	0
17	0
18	0
19	0
20	0
21	0
22	1
23	4
24	8
25	4
26	7
27	14
28	20
29	24
30	55
31	95
⋮	⋮

Table 6.
Weight distribution of a parallel concatenation of three convolutional codes

Weight	Multiplicity
0	1
1	0
2	0
⋮	⋮
16	0
17	0
18	1
19	0
20	1
21	1
22	0
23	3
24	5
25	6
26	8
27	9
28	11
29	26
30	52
31	81
⋮	⋮

Table 7.
Weight distribution of a serial concatenation of two convolutional codes

Compared to the schemes with three parallel concatenated codes and two serial concatenated codes, the weight distribution of the three serial concatenated codes is very encouraging, especially considering the fact that the three component codes are quite simple, with 4, 4, and 8 states, respectively. The best rate $1/4$ convolutional codes with free distance around 28, for example, have about $2^9 \sim 2^{10}$ states [**13**]. (In the example with block length $N = 27$, the triple serial concatenated code is actually a linear $(116, 27)$ block code. The Gilbert-Varshamov bound for a $(116, 27)$ code is $d \geq 25$, or 3 less than the minimum distance of the new code.) Thus this triple serial concatenation scheme should give very good performance, if it can be effectively decoded.

It has been observed that, at very short block lengths, the weight distribution of the triple serial concatenation scheme with two recursive inner codes is a little worse than with the equivalent non-recursive inner codes, since the disadvantages of non-recursive inner codes do not manifest themselves at short block

lengths. However, at the more interesting large block lengths, recursive codes should be chosen as the two inner codes, since if non-recursive codes are used, the minimum distance of the overall code will not increase with increasing block length, thus resulting in no interleaver gain.

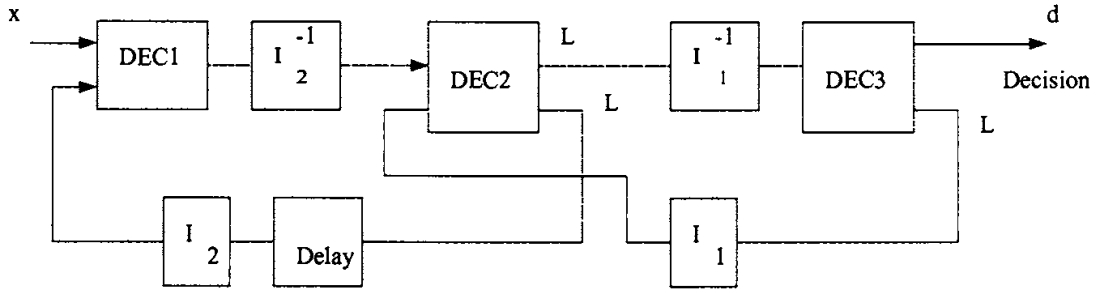


Figure 9. *A decoding structure for the serial concatenation of three codes*

A possible decoding structure for the triple serial concatenation scheme which uses the turbo decoding principle is shown in Figure 9.

5. Conclusions

Various concatenation schemes with inner turbo codes and outer block codes have been proposed in an attempt to overcome some of the drawbacks of turbo codes. They are successful to some degree, but still have several disadvantages. A new hybrid coding scheme, with a parallel concatenated turbo code as the inner code and a convolutional outer code, and another new turbo-like coding scheme, a triple serial concatenation of convolutional codes, have also been proposed. Weight distributions for these new coding scheme have been calculated for very short block lengths and compared to the weight distributions of comparable existing turbo coding schemes. These calculations show that the new schemes have the potential for very good performance and several possible decoding structures have been proposed.

References

[1] J.D. ANDERSON, Turbo codes extended with outer BCH code, *Electronics Letters*, vol. 32, pp. 2059–2060, October 1996.

[2] D. ARNOLD AND G. MEYERHANS, The realization of the turbo-coding system, Semester Project Report, Swiss Federal Institute of Technology, July 1995.

[3] L. BAHL, J. COCKE, F. JELINEK, AND J. RAVIV, Optimal decoding of linear codes for minimizing symbol error rate, *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, March 1974.

[4] S. BENEDETTO, D. DIVSALAR, G. MONTORSI, AND F. POLLARA, Serial concatenation of interleaved codes: performance analysis, design and iterative decoding, *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, May 1998.

[5] S. BENEDETTO, R. GARELLO, AND G. MONTORSI, Canonical structure for systematic k/n convolutional encoders and its application to turbo codes, *Electronics Letters*, vol. 32, pp. 981–982, May 1996.

[6] S. BENEDETTO AND G. MONTORSI, Unveiling turbo codes: some results on parallel concatenated coding schemes, *IEEE Trans. Inform. Theory*, vol. 42, pp. 409-428, March 1996.

[7] C. BERROU, A. GLAVIEUX, AND P. THITIMAJSHIMA, Near Shannon limit error-correcting coding: Turbo codes, in *Proceedings IEEE International Conf.* on *Communications*, pp. 1064-1070, Geneva, Switzerland, July 1993.

[8] O. COLLINS AND M. HIZLAN, Determinate state convolutional codes, *IEEE Trans. Comm.*, vol. 41, pp. 1785–1794, December 1993.

[9] D.J. COSTELLO AND G. MEYERHANS, Concatenated turbo codes, in *Proceedings ISITA'96* pp. 571–574, Victoria, BC, Canada, September 1996.

[10] D. DIVSALAR AND F. POLLARA, Multiple turbo codes for deep-space communications, TDA Progress Report 42-121, pp. 66–77, May 15, 1995.

[11] J. HAGENAUR, E. OFFER, AND L. PAPKE, Iterative decoding of binary block and convolutional codes, *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, March 1996.

[12] J. HE, D.J. COSTELLO, JR., Y.-F. HUANG, AND R.L. STEVENSON, On the application of turbo codes to the robust transmission of compressed images, in *Proceedings ICIP'97*, Santa Barbara, CA, October 1997.

[13] S. LIN AND D.J. COSTELLO, *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

[14] R.J. MCELIECE AND L. SWANSON, On the decoder error probability for Reed-Solomon codes, *IEEE Trans. Inform. Theory*, vol. 32, pp. 701–703, September 1986.

[15] G. MEYERHANS, Interleaver and code design for parallel and serial concatenated convolutional codes, M.Sc. Thesis, University of Notre Dame, March 1996.

[16] K.R. NARAYANAN AND G.L. STUBER, Selective serial concatenation of turbo codes, *IEEE Communications Letters*, vol. 1, pp. 136–139, September 1997.

[17] M. OBERG AND P.H. SIEGEL, Application of distance spectrum analysis to turbo code performance improvement, in *Proceedings 35-th Annual Allerton Conf.* on *Commun., Control, and Computing*, Monticello, IL, September 1997.

[18] L.C. PEREZ, J. SEGHERS, AND D.J. COSTELLO, A distance spectrum interpretation of turbo codes, *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698–1709, November 1996.

[19] P. ROBERTSON, Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes, in *Proceedings GLOBECOM'94*, pp. 1298–1303, San Francisco, CA, December 1994.

[20] J. SEGHERS, On the free distance of turbo codes and related product codes, Diploma Project Report, Swiss Federal Institute of Technology, August 1995.

Part III

SIGNALS AND INFORMATION

Alternating Minimization Algorithms:

From Blahut-Arimoto to Expectation-Maximization

Joseph A. O'Sullivan

DEPARTMENT OF ELECTRICAL ENGINEERING
WASHINGTON UNIVERSITY
ST. LOUIS, MO 63130, USA
jao@ee.wustl.edu

ABSTRACT. In his doctoral thesis and in a prize-winning paper in the *IEEE Transactions on Information Theory*, R.E. Blahut described two computational algorithms that are important in information theory. The first, which was also described by Arimoto, computes channel capacity and the distribution that achieves it. The second computes the rate-distortion function and distributions that achieve it. Blahut derived these algorithms as alternating maximization and minimization algorithms. Two closely related algorithms in estimation theory are the expectation-maximization algorithm and the generalized iterative scaling algorithm, each of which can be written as alternating minimization algorithms. Algorithmic and historical connections between these four algorithms are explored.

1. Introduction

This paper describes a unified framework in which several well-known algorithms are described in terms of alternating maximization or minimization problems. The paper is inspired by the sixtieth birthday of Richard E. Blahut and the twenty-fifth anniversary of the publication of two prize-winning papers by R.E. Blahut and S. Arimoto. Both papers describe the computation of channel capacity and the distribution that achieves channel capacity in terms of an alternating maximization over transition probabilities from the output alphabet to the input alphabet and over the input distribution. Blahut also describes an alternating minimization algorithm for computing the rate-distortion function.

Blahut's algorithm for computing the rate-distortion function is closely related to the expectation-maximization algorithm used in estimation theory. This algorithm can also be described as an alternating minimization algorithm, as discussed by many authors, but in particular by Csiszár and Tusnady [**10**]. The Blahut-Arimoto algorithm for computing channel capacity and the distribution that achieves capacity is closely related to the generalized iterative scaling algorithm of Darroch and Ratcliff [**11**], which can also be described as an alternating minimization algorithm (see Byrne [**4**]). This paper attempts to make the relationship between these algorithms more transparent.

Fundamental papers on each of the four algorithms appeared in 1972: the papers by Blahut [**2**] and Arimoto [**1**] that introduce the algorithms for channel capacity and rate-distortion computations; the paper by Darroch and Ratcliff [**11**] that introduces the generalized iterative scaling algorithm; and the paper by Richardson [**19**] that introduces a version of the expectation-maximization algorithm in the form discussed here. Darroch-Ratcliff and Richardson did not write their algorithms as alternating minimization algorithms; this came much later.

Richardson's algorithm [**19**] was designed for use in astronomical imaging and does not appear to have been known within the statistics community. It was not described as an expectation-maximization algorithm, but is clearly an iterative algorithm designed to estimate an input distribution consistent with a measured distribution. It is a nonparametric estimation algorithm in that no assumptions (other than nonnegativity and a sum constraint) are assumed about the input distribution. The definitive paper on the expectation-maximization algorithm was written by Dempster, Laird, and Rubin [**12**] in 1977. In their paper, several previous iterative algorithms that may be considered expectation-maximization algorithms are presented, and the general framework for deriving expectation-maximization algorithms is given. Their derivation can be reinterpreted as being an alternating minimization algorithm; for more details, see Vardi, Shepp, and Kaufman [**24**] or Csiszár and Tusnady [**10**].

Recent papers have applied the alternating minimization technique, analogous to the expectation-maximization algorithm, to deterministic linear inverse problems subject to nonnegativity constraints [**22, 23, 4**]. The I-divergence or relative entropy is defined as the objective function to be minimized. The iterations are shown to decrease the I-divergence monotonically, and assumptions about the problem guarantee uniqueness of the resulting estimate. The algorithm is the same as the expectation-maximization algorithm applied to emission tomography problems [**20, 24, 18**], and much of the convergence analysis is similar.

The generalized iterative scaling algorithm was originally introduced to find the distribution that maximizes entropy subject to a set of linear (mean-value) constraints [**11**]. It has been shown to minimize the I-divergence or cross-entropy between a linear model and nonnegative data, using an alternating minimization approach [**4**]. Byrne referred to this algorithm as SMART for the simultaneous multiplicative algebraic reconstruction technique.

The common framework in which the four algorithms are presented is motivated by the approach of Blahut [**2**]. Solution to each problem is written as the mini-

mum of an objective function. The objective function is written as the minimum of a relaxed objective function over an auxiliary variable. The solution is then described by a double minimization. An alternating minimization algorithm results by alternately minimizing the objective function over each of the variables.

Some analysis related to the convergence of these algorithms is also presented. This analysis is based on Csiszár and Tusnady's discussion of alternating minimization algorithms [**10**] and Hero and Fessler's discussion of A-algorithms [**13**].

In order to keep the analysis straightforward, some simplifications are made. First, only the finite dimensional case is treated. There are extensions of many of the results here to countably infinite spaces and to function spaces. Secondly, in order to emphasize the connection between the estimation-theoretic algorithms and the algorithms from the 1972 paper of Blahut [**2**], nonparametric estimation algorithms are considered. That is, the vectors being estimated are unconstrained other than being nonnegative or being probability vectors.

It is hoped that the connections between the two sets of algorithms may yield new insights. For example, using the analogy to the estimation problems, algorithms for solving constrained or parameterized channel capacity problems may be obtained. Algorithms for finding constrained or parameterized rate-distortion problems may also result. The results here form a bridge between traditional information theory and estimation theory.

2. Notation and information measures

All of the vectors and matrices discussed in this paper have nonnegative entries (in $\mathbb{R}_+$). All logarithms are natural logarithms. In some cases it is necessary to restrict the entries to be positive. Many of the results involve probability vectors and probability transition matrices. Let the sets of nonnegative $n \times 1$ and $m \times 1$ vectors be denoted by $\mathbb{R}_+^n$ and $\mathbb{R}_+^m$, respectively. Let

$$\mathcal{P} = \left\{\mathbf{P} = [P_{j|k}] \in \mathbb{R}_+^{n \times m}\right\} \qquad \mathcal{Q} = \left\{\mathbf{Q} = [Q_{k|j}] \in \mathbb{R}_+^{m \times n}\right\}$$

We will also need notation for the restriction of the above to probability vectors and to channel transition probability matrices, namely:

$$\Pi^n = \left\{\mathbf{p} \in \mathbb{R}_+^n : \textstyle\sum_{j=1}^n p_j = 1\right\} \qquad \Pi^m = \left\{\mathbf{q} \in \mathbb{R}_+^m : \textstyle\sum_{k=1}^m q_k = 1\right\}$$

$$\mathcal{P}^\Delta = \left\{\mathbf{P} \in \mathcal{P} : \sum_{j=1}^n P_{j|k} = 1, \forall k\right\} \qquad \mathcal{Q}^\Delta = \left\{\mathbf{Q} \in \mathcal{Q} : \sum_{k=1}^m Q_{k|j} = 1, \forall j\right\}$$

There are two channels that are of interest: a forward channel, and a backward channel. These channels are illustrated in Figure 1.

If $\mathbf{p} \in \Pi^n$ and $\mathbf{P} \in \mathcal{P}^{\Delta}$, while $\mathbf{q} \in \Pi^m$ and $\mathbf{Q} \in \mathcal{Q}^{\Delta}$, then these channel models describe two different probabilistic models for $(\boldsymbol{X}, \boldsymbol{Y})$ pairs. This special case is referred to as the probability distribution model.

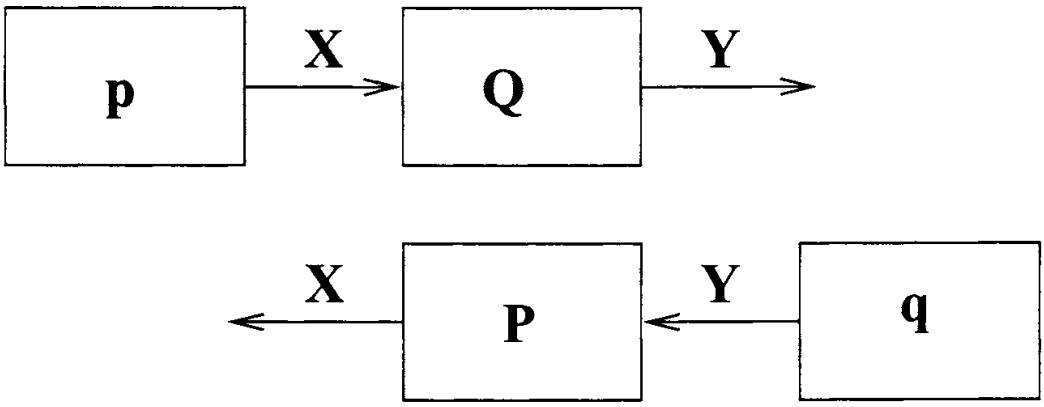


Figure 1. *Forward and backward channel models*

In general, the forward and backward channel models are two possible descriptions for the $(\boldsymbol{X}, \boldsymbol{Y})$ pairs. These descriptions are matrices $\mathbf{Q} \circ \mathbf{p} = [Q_{k|j} p_j]$ and $\mathbf{P} \circ \mathbf{q} = [P_{j|k} q_k]$. Each of these descriptions has corresponding vectors that are the marginal probabilities in the probability distribution model case, and are simply the matrix-vector products in general: $\mathbf{Qp}$ and $\mathbf{Pq}$.

Definition 1. *The I-divergence between two positive vectors* $\mathbf{p}$ *and* $\mathbf{r}$ *of length* n *is defined as follows:*

$$I(\mathbf{p}||\mathbf{r}) \stackrel{\text{def}}{=} \sum_{j=1}^{n} \left(p_j \log \frac{p_j}{r_j} - p_j + r_j \right) \tag{1}$$

Note that if both $\mathbf{p}$ and $\mathbf{r}$ are vectors in Π^n, then the I-divergence equals the relative entropy between the distributions. This quantity is given many names in the literature. Kullback and Leibler define this in [**16**, p.80] and in [**14**, p.5] as the mean information for discrimination, or simply the discrimination information, and Kullback notes [**14**, pp. 6–7] that it can be considered to be a directed divergence. Other names for this quantity include the Kullback-Leibler distance [**5**, p.18], cross entropy [**21**], discrimination [**3**, p.17], information divergence or I-divergence [**9**], Kullback-Leibler information number, and Kullback-Leibler divergence. The terminology *I-divergence* is adopted here due to the axiomatic derivation given by Csiszár [**9**] and his emphasis on including the last two terms in (1) when the vectors are not probability vectors.

There are many important properties of the I-divergence. We now briefly discuss some of these properties.

Property 1. *$I(\mathbf{p}||\mathbf{r}) \geq 0$. Furthermore $I(\mathbf{p}||\mathbf{r}) = 0$ if and only if $\mathbf{p} = \mathbf{r}$.*

Proof. This follows directly from the inequality $\log x = \ln x \geq 1 - 1/x$, with equality iff $x = 1$. In fact each term in the sum in (1) is nonnegative. ∎

Property 2. *Divide the sum (1) in the definition of I-divergence into two parts, say A and $\bar{A}$, and define $P(A) = \sum_{j \in A} p_j$ and $P(\bar{A}) = \sum_{j \in \bar{A}} p_j$. Similarly define $R(A)$ and $R(\bar{A})$. Then*

$$I(\mathbf{p}||\mathbf{r}) \geq \left[P(A) \log \frac{P(A)}{R(A)} - P(A) + R(A)\right] + \left[P(\bar{A}) \log \frac{P(\bar{A})}{R(\bar{A})} - P(\bar{A}) + R(\bar{A})\right]$$

Proof. This follows directly from the log-sum inequality and leads directly to the next property [5, pp. 29–30]. ∎

Property 3. *The I-divergence $I(\mathbf{p}||\mathbf{r})$ is convex in the pair $(\mathbf{p}, \mathbf{r})$.*

Property 4. *Let $\mathcal{L} \subset \mathbb{R}_+^n$ be any nonempty subset satisfying a linear constraint such as $\mathcal{L} = \{\mathbf{p} \in \mathbb{R}_+^n : \mathbf{Q}\mathbf{p} = \mathbf{q}\}$, where $\mathbf{Q} \in \mathcal{Q}$ and $\mathbf{q} \in \mathbb{R}_+^m$ is a positive vector. Further define $\mathbf{p}^* = \operatorname{argmin}_{\mathbf{p} \in \mathcal{L}} I(\mathbf{p}||\mathbf{r})$. Then for any $\mathbf{p} \in \mathcal{L}$, we have:*

$$I(\mathbf{p}||\mathbf{r}) = I(\mathbf{p}||\mathbf{p}^*) + I(\mathbf{p}^*||\mathbf{r}) \tag{2}$$

Proof. The proof of this property dates at least to Kullback [14] and Kullback and Khairat [15] when $\mathbf{p}$ and $\mathbf{r}$ are probability vectors. The generalized iterative scaling algorithm discussed below solves this problem as well [11]. Let

$$p_j^* = r_j \exp\left(\sum_k Q_{k|j} \lambda_k\right)$$

where λ_k are Lagrange multipliers chosen so that $\mathbf{p}^* \in \mathcal{L}$. Then:

$$\begin{aligned} I(\mathbf{p}^*||\mathbf{r}) &= \sum_j p_j^* \left(\sum_k Q_{k|j} \lambda_k\right) + \sum_j (r_j - p_j^*) && (3) \\ &= \sum_k \lambda_k \sum_j Q_{k|j} p_j^* + \sum_j (r_j - p_j^*) && (4) \\ &= \sum_k \lambda_k \sum_j Q_{k|j} p_j + \sum_j (r_j - p_j^* + p_j - p_j) && (5) \end{aligned}$$

The first term in (5) also appears in

$$I(\mathbf{p}||\mathbf{p}^*) = \sum_j p_j \left(\log p_j - \log r_j - \sum_k Q_{k|j} \lambda_k\right) + \sum_j (p_j^* - p_j) \tag{6}$$

Adding the results in (5) and (6), we get the result stated in the property. ∎

This definition of I-divergence may be extended to matrices, so that the I-divergence between forward and backward channel models equals:

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}) = \sum_j \sum_k \left(p_j Q_{k|j} \log \frac{p_j Q_{k|j}}{q_k P_{j|k}} - p_j Q_{k|j} + q_k P_{j|k} \right) \quad (7)$$

In the context of of alternating minimization algorithms, it will be useful to rearrange the expression above as follows:

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}) =$$

$$\sum_j \sum_k \left(p_j Q_{k|j} \log \frac{p_j}{P_{j|k}} + p_j Q_{k|j} \log \frac{Q_{k|j}}{q_k} - p_j Q_{k|j} + q_k P_{j|k} \right) \quad (8)$$

When restricted for use with probability vectors and probability transition matrices, the last two terms cancel each other. The two remaining terms play a central role in Blahut's derivations. Define mutual information as

$$I(\mathbf{p}, \mathbf{Q}) = \sum_j \sum_k p_j Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} Q_{k|j'} p_{j'}}$$

Blahut presented two variational representations of mutual information,

$$I(\mathbf{p}, \mathbf{Q}) = \max_{\mathbf{P} \in \mathcal{P}^\Delta} \sum_j \sum_k p_j Q_{k|j} \log \frac{P_{j|k}}{p_j}$$

and

$$I(\mathbf{p}, \mathbf{Q}) = \min_{\mathbf{q} \in \Pi^m} \sum_j \sum_k p_j Q_{k|j} \log \frac{Q_{k|j}}{q_k}$$

Thus, in (8), the minimum of the first term over $\mathbf{P}$ is $-I(\mathbf{p}, \mathbf{Q})$ and the minimum of the second term over $\mathbf{q}$ is $I(\mathbf{p}, \mathbf{Q})$. From the way that the joint I-divergence is separated into terms, it is also apparent that $\mathbf{P}$ only contributes to the first term and $\mathbf{q}$ only contributes to the second.

These two variational representations of mutual information play central roles in the derivation of the Blahut-Arimoto algorithm and Blahut's algorithm for computing the rate-distortion function, respectively.

3. Minimization of I-divergence

The joint I-divergence has four arguments when written as in (7). We have the following simple but useful lemma.

Lemma 5. *$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q})$ is convex in each of its four arguments.*

Proof. This follows directly from Property 3 of I-divergence. ∎

The results of minimizing over these variables are discussed next. Each of these minimizations plays a part in the algorithms.

3.1. Minimization over Q. Let $\widetilde{\mathbf{Q}}^* = \operatorname{argmin}_{\mathbf{Q}\in\mathcal{Q}} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$. Then $\widetilde{Q}^*_{k|j} = P_{j|k}q_k/p_j$ and we have:

$$I(\widetilde{\mathbf{Q}}^*\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q}) \;=\; 0$$

Let $\mathbf{Q}^* = \operatorname{argmin}_{\mathbf{Q}\in\mathcal{Q}^\Delta} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$. Then $Q^*_{k|j} = P_{j|k}q_k/\sum_{k'} P_{j|k'}q_{k'}$ and

$$I(\mathbf{Q}^*\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q}) \;=\; I(\mathbf{p}||\mathbf{Pq})$$

This equality plays a key role in our analysis of Blahut's rate-distortion algorithm and of the expectation-maximization algorithm.

This statement on minimization over $\mathbf{Q}$ may be generalized. Let $\mathcal{Q}_1$ be the intersection of $\mathcal{Q}^\Delta$ with any linear space $\mathcal{E}$, and assume that $\mathcal{Q}_1$ is not empty. If

$$\mathbf{Q}^* = \operatorname*{argmin}_{\mathbf{Q}\in\mathcal{Q}_1} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$$

then for all $\mathbf{Q}\in\mathcal{Q}_1$, we have:

$$I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q}) \;=\; I(\mathbf{Q}\circ\mathbf{p}||\mathbf{Q}^*\circ\mathbf{p}) + I(\mathbf{Q}^*\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q}) \tag{9}$$

This is just an application of Property 4 of I-divergence and is used extensively in analysis of convergence (see [11, Lemma 2] and [7, Theorem 3.1] for more details). For the rate distortion algorithm, the set $\mathcal{Q}_1$ becomes the set of transition probabilities that satisfy the distortion constraint.

3.2. Minimization over p. Let $\mathbf{p}^* = \operatorname{argmin}_{\mathbf{p}\in\mathbb{R}^n_+} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$. Then

$$p^*_j \;=\; \exp\left(\sum_k \frac{Q_{k|j}}{\sum_{k'} Q_{k'|j}} \log\frac{q_k P_{j|k}}{Q_{k|j}}\right)$$

and

$$\begin{aligned} I(\mathbf{Q}\circ\mathbf{p}^*||\mathbf{P}\circ\mathbf{q}) \;&=\; \sum_j\sum_k \left(-p^*_j Q_{k|j} + q_k P_{j|k}\right) \\ &=\; -\sum_j\left(\sum_{k'} Q_{k'|j}\right)\exp\left(\sum_k \frac{Q_{k|j}}{\sum_{k'} Q_{k'|j}}\log\frac{q_k P_{j|k}}{Q_{k|j}}\right) \\ &\quad + \sum_j\sum_k q_k P_{j|k} \end{aligned}$$

Now consider the restriction of $\mathbf{p}$ to probability distributions. In this case, we have $\mathbf{p}^* = \operatorname{argmin}_{\mathbf{p}\in\Pi^n} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$, and

$$p^*_j \;=\; \frac{\exp\left(\sum_k \frac{Q_{k|j}}{\sum_{k'} Q_{k'|j}}\log\frac{q_k P_{j|k}}{Q_{k|j}}\right)}{\sum_{j'}\exp\left(\sum_k \frac{Q_{k|j'}}{\sum_{k'} Q_{k'|j'}}\log\frac{q_k P_{j'|k}}{Q_{k|j'}}\right)}$$

By Property 4 of I-divergence, for either of the minimizations in this subsection,

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}) \;=\; I(\mathbf{Q} \circ \mathbf{p} || \mathbf{Q} \circ \mathbf{p}^*) + I(\mathbf{Q} \circ \mathbf{p}^* || \mathbf{P} \circ \mathbf{q})$$

For the probability distribution model, the first term simplifies so that

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}) \;=\; I(\mathbf{p} || \mathbf{p}^*) + I(\mathbf{Q} \circ \mathbf{p}^* || \mathbf{P} \circ \mathbf{q})$$

3.3. Minimization over P. Let $\tilde{\mathbf{P}} = \text{argmin}_{\mathbf{P} \in \mathcal{P}} I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q})$. It is easy to see that $\tilde{\mathbf{P}} = Q_{k|j} p_j / q_k$ and therefore

$$I(\mathbf{Q} \circ \mathbf{p} || \tilde{\mathbf{P}} \circ \mathbf{q}) \;=\; 0$$

As in the first minimization over $\mathbf{Q}$, this unconstrained minimization is not interesting. However, the identity $\tilde{\mathbf{P}} = Q_{k|j} p_j / q_k$ does play a role below.

Let $\mathbf{P}^* = \text{argmin}_{\mathbf{P} \in \mathcal{P}^\Delta} I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q})$. Then we have $\mathbf{P}^* = Q_{k|j} p_j / \sum_{j'} Q_{k|j'} p_{j'}$ and therefore

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P}^* \circ \mathbf{q}) \;=\; I(\mathbf{Q}\mathbf{p} || \mathbf{q}) \tag{10}$$

There is a representation of the joint I-divergence for the minimization over $\mathbf{P}$ that is similar to that obtained in (9) for minimization over $\mathbf{Q}$. Property 4 does not apply since the minimization is over the second variable in the I-divergence. However, a similar decomposition holds. Let $\tilde{\mathbf{P}} = Q_{k|j} p_j / q_k$ as before. Also, let $\tilde{\mathbf{q}}$ be the output distribution consistent with $\mathbf{Q}$ and $\mathbf{p}$, namely $\tilde{\mathbf{q}} = \sum_j Q_{k|j} p_j$. Then, for all $\mathbf{P} \in \mathcal{P}^\Delta$, we have

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}) \;=\; I(\tilde{\mathbf{P}} \circ \mathbf{q} || \mathbf{P} \circ \mathbf{q}) \;=\; I(\tilde{\mathbf{P}} \circ \mathbf{q} || \mathbf{P}^* \circ \mathbf{q}) + I(\tilde{\mathbf{P}} \circ \tilde{\mathbf{q}} || \mathbf{P} \circ \tilde{\mathbf{q}})$$

Further noting that:

$$\tilde{\mathbf{P}} \circ \mathbf{q} \;=\; \mathbf{Q} \circ \mathbf{p} \;=\; \mathbf{P}^* \circ \tilde{\mathbf{q}}$$

the I-divergence may be decomposed as

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}) \;=\; I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P}^* \circ \mathbf{q}) + I(\mathbf{P}^* \circ \tilde{\mathbf{q}} || \mathbf{P} \circ \tilde{\mathbf{q}}) \tag{11}$$

For the Blahut-Arimoto algorithm, it is worth noting that the sum in (11) can be also decomposed in a second way. Assume the probability distribution model, and note that

$$\log \frac{Q_{k|j} p_j}{P^*_{j|k} q_k} \;=\; \log \frac{p_j \sum_{j'} Q_{k|j'} p'_j}{Q_{k|j} p_j} + \log \frac{Q_{k|j} p_j}{p_j q_k} \tag{12}$$

Denote the product distribution by $\mathbf{p}\mathbf{q}^T$. The first term in the expansion in (11) may be rewritten, using the decomposition in (12), as

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P}^* \circ \mathbf{q}) \;=\; -I(\mathbf{Q} \circ \mathbf{p} || \mathbf{p}(\mathbf{Q}\mathbf{p})^T) + I(\mathbf{Q} \circ \mathbf{p} || \mathbf{p}\mathbf{q}^T) \tag{13}$$

This equality in fact holds in the general case, where of course the restriction $\mathbf{P} \in \mathcal{P}^\Delta$ remains. Finally, note that the first term in (13) is in fact the mutual information for the forward channel, so

$$I(\mathbf{Q} \circ \mathbf{p} || \mathbf{P}^* \circ \mathbf{q}) \;=\; -I(\mathbf{p}, \mathbf{Q}) + I(\mathbf{Q} \circ \mathbf{p} || \mathbf{p}\mathbf{q}^T)$$

3.4. Minimization over q. Let $\mathbf{q}^* = \operatorname{argmin}_{\mathbf{q}\in\mathbb{R}_+^m} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$. Then

$$q_k^* = \frac{\sum_j p_j Q_{k|j}}{\sum_j P_{j|k}}$$

The minimum value of the joint I-divergence does not really simplify from the original. However, there is a decomposition into the sum of two I-divergences as in the other cases, as follows:

$$I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q}) = I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q}^*) + I(\mathbf{P}\circ\mathbf{q}^*||\mathbf{P}\circ\mathbf{q}) \tag{14}$$

For the probability distribution model, $\sum_j P_{j|k} = 1$ for all k, so $\mathbf{q}^* = \mathbf{Q}\mathbf{p}$ and

$$I(\mathbf{P}\circ\mathbf{q}^*||\mathbf{P}\circ\mathbf{q}) = I(\mathbf{q}^*||\mathbf{q})$$

The first term on the right side of (14) can be rearranged as follows:

$$I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q}^*) = -\sum_j\sum_k p_j Q_{k|j}\log\frac{P_{j|k}}{p_j} + I\left(\mathbf{Q}\circ\mathbf{p}||\mathbf{p}(\mathbf{Q}\mathbf{p})^T\right) \tag{15}$$

$$= -\sum_j\sum_k p_j Q_{k|j}\log\frac{P_{j|k}}{p_j} + I(\mathbf{p},\mathbf{Q}) \tag{16}$$

In this form, we see that the second term is the mutual information between the input and the output for the forward channel model.

4. Blahut-Arimoto algorithm

It is well-known that the capacity of a channel with transition probability matrix $\mathbf{Q}$ is defined as follows:

$$C = \max_{\mathbf{p}\in\Pi^n} I(\mathbf{p},\mathbf{Q})$$

The variational representation of the mutual information is actually one term in (11) and yields

$$C = \max_{\mathbf{p}\in\Pi^n}\max_{\mathbf{P}\in\mathcal{P}^\Delta}\sum_j\sum_k p_j Q_{k|j}\log\frac{P_{j|k}}{p_j} \tag{17}$$

$$= \max_{\mathbf{p}\in\Pi^n}\max_{\mathbf{P}\in\mathcal{P}^\Delta}\left\{\sum_j\sum_k p_j Q_{k|j}\log\frac{P_{j|k}}{p_j} - \sum_j\sum_k q_k P_{j|k}\right\} \tag{18}$$

This double minimization leads to the Blahut-Arimoto algorithm for computing capacity. The Blahut-Arimoto algorithm is an alternating maximization algorithm, that alternates between maximizing (18) over $\mathbf{p}$ and over $\mathbf{P}$.

Blahut-Arimoto algorithm:

- **Choose initial guess for** $\mathbf{p}^0 \in \Pi^n$, $p_j^0 > 0$ **for all** j; **set** $r = 0$.
- **P-Step** (or channel estimation step)

$$P_{j|k}^r = \frac{p_j^r Q_{k|j}}{\sum_j p_j^r Q_{k|j}}$$

- **p-step** (or input distribution step)

$$p_j^{r+1} = \frac{\exp\left(\sum_k Q_{k|j} \log P_{j|k}^r\right)}{\sum_j \exp\left(\sum_k Q_{k|j} \log P_{j|k}^r\right)}$$

- **If convergence is achieved, stop; else set** $r = r+1$ **and iterate.**

The Blahut-Arimoto algorithm may be written as an A-algorithm or a one-step algorithm. Define the term in (18) over which the minimization takes place as

$$J(\mathbf{p}, \mathbf{Q}, \mathbf{P}) = \sum_j \sum_k p_j Q_{k|j} \log \frac{P_{j|k}}{p_j}$$

Then

$$A(\mathbf{p}|\mathbf{p}^r) = -J(\mathbf{p}, \mathbf{Q}, \mathbf{P}^r) = \sum_j \sum_k p_j Q_{k|j} \log \frac{p_j (\sum_{j'} p_{j'}^r Q_{k|j'})}{p_j^r Q_{k|j}}$$

and

$$\mathbf{p}^{r+1} = \operatorname*{argmin}_{\mathbf{p} \in \Pi^n} A(\mathbf{p}|\mathbf{p}^r) \tag{19}$$

$$p_j^{r+1} = p_j^r \frac{\exp\left(\sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} p_{j'}^r Q_{k|j'}}\right)}{\sum_j p_j^r \exp\left(\sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} p_{j'}^r Q_{k|j'}}\right)} \tag{20}$$

The convergence has been analyzed by Blahut, Arimoto, and Csiszár and Tusnady [10] in the context of alternating minimization algorithms. Note that the function $J(\cdot, \cdot, \cdot)$ is convex in both $\mathbf{p}$ and $\mathbf{P}$. Following Arimoto [1], we denote

$$C(r+1, r) = J(\mathbf{p}^{r+1}, \mathbf{Q}, \mathbf{P}^r) \quad \text{and} \quad C(r, r) = J(\mathbf{p}^r, \mathbf{Q}, \mathbf{P}^r)$$

The successive minimizations yield

$$-\mathcal{C}(0,0) \geq -\mathcal{C}(1,0) \geq -\mathcal{C}(1,1) \geq \cdots \geq -\mathcal{C}(r,r-1) \geq -\mathcal{C}(r,r) \geq \cdots \geq -\mathcal{C}$$

From (11), for any $\mathbf{q}$, we have:

$$\begin{aligned} I(\mathbf{Q} \circ \mathbf{p}^r || \mathbf{P}^{r-1} \circ (\mathbf{Q}\mathbf{p}^r)) &= I(\mathbf{Q} \circ \mathbf{p}^r || \mathbf{P}^{r-1} \circ \mathbf{q}) - I(\mathbf{Q} \circ \mathbf{p}^r || \mathbf{P}^r \circ \mathbf{q}) \\ &= -\mathcal{C}(r,r-1) + \mathcal{C}(r,r) \\ &= -J(\mathbf{p}^r, \mathbf{Q}, \mathbf{P}^{r-1}) + J(\mathbf{p}^r, \mathbf{Q}, \mathbf{P}^r) \end{aligned}$$

Let $\mathbf{p}^*$ be a distribution that achieves capacity. Then

$$\begin{aligned} \sum_j p_j^* \log \frac{p_j^{r+1}}{p_j^r} &= \sum_j p_j^* \sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} p_{j'}^r Q_{k|j'}} - \mathcal{C}(r+1,r) \\ &= -\mathcal{C}(r+1,r) + \sum_j p_j^* \sum_k Q_{k|j} \log \frac{Q_{k|j} \sum_{j'} p_{j'}^* Q_{k|j'}}{\sum_{j'} p_{j'}^* Q_{k|j'} \sum_{j'} p_{j'}^r Q_{k|j'}} \\ &= \mathcal{C} - \mathcal{C}(r+1,r) + I(\mathbf{Q}\mathbf{p}^* || \mathbf{Q}\mathbf{p}^r) \end{aligned} \tag{21}$$

where

$$\mathcal{C}(r+1,r) = \log \sum_j p_j^r \exp \left(\sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} p_{j'}^r Q_{k|j'}} \right)$$

Thus

$$\mathcal{C} - \mathcal{C}(r+1,r) \leq \sum_j p_j^* \log \frac{p_j^{r+1}}{p_j^r} \tag{22}$$

Summing both sides of (22), we obtain:

$$\sum_{r=0}^{N} \mathcal{C} - \mathcal{C}(r+1,r) \leq \sum_{r=0}^{N} \sum_j p_j^* \log \frac{p_j^{r+1}}{p_j^r} = \sum_j p_j^* \log \frac{p_j^{N+1}}{p_j^0} \tag{23}$$

But the right-hand side of (23) is bounded by:

$$\sum_j p_j^* \log \frac{p_j^{N+1}}{p_j^0} = I(\mathbf{p}^* || \mathbf{p}^0) - I(\mathbf{p}^* || \mathbf{p}^N) \leq I(\mathbf{p}^* || \mathbf{p}^0)$$

Since this holds for all N, the sum of these nonnegative values is finite and

$$\lim_{r \to \infty} \mathcal{C}(r+1,r) = \mathcal{C}$$

Note that this also implies [1] from (21) that $\lim_{N \to \infty} I(\mathbf{Q}\mathbf{p}^* || \mathbf{Q}\mathbf{p}^r) = 0$. Also note that if there is a unique $\mathbf{p}^*$ that achieves capacity, then $\mathbf{p}^r$ converges to it.

Blahut [2] extended these results to include the case where $\mathbf{p}$ is constrained to satisfy a linear constraint, such as

$$\mathbf{p} \in \mathbf{P}_E = \left\{ \mathbf{p} \in \mathbf{P}^n : \textstyle\sum_j p_j e_j \leq E \right\}$$

The strategy is to alternate as before, but with an additional factor. There is a one-to-one relationship between a Lagrange multiplier used to enforce the

constraint and the constrained value E. The iteration uses a fixed value of the Lagrange multiplier; after convergence, this multiplier may be changed and the iteration repeated. The Lagrange multiplier trades off values of E for capacity C.

5. Generalized iterative scaling

Darroch and Ratcliff [11] introduced the technique of generalized iterative scaling for finding the solution of a linear inverse problem that maximizes entropy. Later, this algorithm was reintroduced [4] as the simultaneous multiplicative algebraic reconstruction technique. The basic properties of this algorithm were laid out by Darroch and Ratcliff, although they did not discuss it as an alternating minimization algorithm. Csiszár [8] showed that generalized iterative scaling can be interpreted as alternating I-projections and the convergence thus follows from his more general results [7]. Byrne [4] showed that this algorithm is in fact an alternating minimization algorithm, whose convergence is covered by Csiszár and Tusnady [10]. The algorithm and its basic properties are introduced, then its relationship to the Blahut-Arimoto algorithm is discussed.

Suppose that a vector of data $\mathbf{q}$ is given, along with a forward channel model $\mathbf{Q}$. The problem addressed is to find the vector $\mathbf{p}$ that minimizes $I(\mathbf{Qp}||\mathbf{q})$. This problem becomes an alternating minimization problem by using the variational representation in (10) to get

$$\hat{\mathbf{p}} = \operatorname*{argmin}_{\mathbf{p}\in\mathbb{R}^n_+} I(\mathbf{Qp}||\mathbf{q}) = \operatorname*{argmin}_{\mathbf{p}\in\mathbb{R}^n_+} \min_{\mathbf{P}\in\mathcal{P}^\Delta} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$$

Generalized iterative scaling algorithm:

- **Choose an initial guess** $\mathbf{p}^0 \in \mathbb{R}^n_+$, $p^0_j > 0$ **for all** j; **set** $r = 0$.
- **P-Step**

$$P^r_{j|k} = \frac{p^r_j Q_{k|j}}{\sum_{j'} p^r_{j'} Q_{k|j'}}$$

- **p-step**

$$p^{r+1}_j = \exp\left(\sum_k Q_{k|j} \log \frac{P^r_{j|k} q_k}{Q_{k|j}}\right)$$

Rearranging

$$p^{r+1}_j = p^r_j \prod_k \left(\frac{q_k}{\sum_{j'} p^r_{j'} Q_{k|j'}}\right)^{Q_{k|j}} \tag{24}$$

- **If convergence is achieved, stop; else set** $r = r+1$ **and iterate.**

Theorem 6. *If there exists a vector* $\hat{\mathbf{p}} \in \mathbb{R}_+^n$ *such that* $\mathbf{Q}\hat{\mathbf{p}} = \mathbf{q}$, *then:*

a. *The sequence* $\mathbf{p}^r$ *converges to the maximum entropy solution subject to the equality constraint* $\mathbf{Qp} = \mathbf{q}$.

b. *Each vector in the sequence is of the exponential form*

$$p_j^r = \pi_j \exp\left(\sum_k Q_{k|j} s_k\right)$$

where the s_k *are Lagrange multipliers, and* π_j *is the initial guess.*

The generalized iterative scaling algorithm may be written as an A-algorithm by defining

$$\begin{aligned} p_j^{r+1} &= \operatorname*{argmin}_{\mathbf{p}\in\mathbb{R}_+^n} A(\mathbf{p}|\mathbf{p}^r) = \operatorname*{argmin}_{\mathbf{p}\in\mathbb{R}_+^n} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}(\mathbf{p}^r)\circ\mathbf{q}) \\ &= \operatorname*{argmin}_{\mathbf{p}\in\mathbb{R}_+^n} I(\mathbf{p}||\mathbf{p}^r) + \sum_j p_j \sum_k Q_{k|j} \log \frac{\sum_{j'} p_{j'}^r Q_{k|j'}}{q_k} \end{aligned}$$

A link to the Blahut-Arimoto algorithm may be seen by comparing (20) and (24). These equations have similar forms. In fact, they would be the same if $\mathbf{q}$ had the property that for all j such that $p_j > 0$,

$$\sum_k Q_{k|j} \log \frac{Q_{k|j}}{q_k}$$

is a constant. That is, the I-divergence between $\mathbf{q}$ and each row of $\mathbf{Q}$ should be constant. This is in fact related to the Kuhn-Tucker conditions for convergence of the Blahut-Arimoto algorithm. If $\mathbf{q} = \mathbf{Qp}^*$, where $\mathbf{p}^*$ is the capacity achieving distribution, then for all j such that $p_j > 0$,

$$\sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} p_{j'}^* Q_{k|j'}} = C$$

and for all other j,

$$\sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} p_{j'}^* Q_{k|j'}} \leq C$$

An outline of the convergence analysis is now given. Let

$$\begin{aligned} \mathcal{G}(r+1, r) &= I(\mathbf{Q}\circ\mathbf{p}^{r+1}||\mathbf{P}^r\circ\mathbf{q}) \\ \mathcal{G}(r, r) &= I(\mathbf{Q}\circ\mathbf{p}^r||\mathbf{P}^r\circ\mathbf{q}) \end{aligned}$$

Then there is a string of inequalities

$$\mathcal{G}(0,0) \geq \mathcal{G}(1,0) \geq \mathcal{G}(1,1) \geq \cdots \geq \mathcal{G}(r,r) \geq \mathcal{G}(r+1,r) \geq \cdots \geq \mathcal{G}^*$$

where $\mathcal{G}^* = \min_{\mathbf{p}\in\mathbb{R}_+^n} I(\mathbf{Q}\mathbf{p}||\mathbf{q})$. Let $\mathbf{p}^*$ achieve the minimum of $I(\mathbf{Q}\mathbf{p}||\mathbf{q})$. From the minimization over $\mathbf{p}$, we have:

$$\mathcal{G}(r+1,r) = -\log\left(\sum_m p_m^r \prod_k \left(\frac{q_k}{\sum_{j'} p_{j'}^r Q_{k|j'}}\right)^{Q_{k|m}}\right)$$

Using the same strategy we used for the Blahut-Arimoto algorithm, we get:

$$\begin{aligned}\sum_j p_j^* \log \frac{p_j^{r+1}}{p_j^r} &= \sum_j p_j^* \sum_k Q_{k|j} \log \frac{q_k}{\sum_{j'} p_{j'} Q_{k|j'}} + \mathcal{G}(r+1,r) \\ &= \mathcal{G}(r+1,r) - I(\mathbf{Q}\mathbf{p}^*||\mathbf{q}) + I(\mathbf{Q}\mathbf{p}^*||\mathbf{Q}\mathbf{p}^r) \\ &= \mathcal{G}(r+1,r) - \mathcal{G}^* + I(\mathbf{Q}\mathbf{p}^*||\mathbf{Q}\mathbf{p}^r) \qquad (25)\end{aligned}$$

Thus

$$\mathcal{G}(r+1,r) - \mathcal{G}^* \leq \sum_j p_j^* \log \frac{p_j^{r+1}}{p_j^r}$$

and

$$\begin{aligned}\sum_{r=0}^N \mathcal{G}(r+1,r) - \mathcal{G}^* &\leq \sum_{r=0}^N \sum_j p_j^* \log \frac{p_j^{r+1}}{p_j^r} = \sum_j p_j^* \log \frac{p_j^{N+1}}{p_j^0} \\ &= I(\mathbf{p}^*||\mathbf{p}^0) - I(\mathbf{p}^*||\mathbf{p}^N) \leq I(\mathbf{p}^*||\mathbf{p}^0)\end{aligned}$$

The right side does not depend on N, and the sum of the nonnegative values on the left side is bounded. Thus

$$\lim_{r\to\infty} \mathcal{G}(r+1,r) = \mathcal{G}^*$$

The same argument applied to (25) shows that $\lim_{r\to\infty} I(\mathbf{Q}\mathbf{p}^*||\mathbf{Q}\mathbf{p}^r) = 0$. If in addition, the solution for $\mathbf{p}^*$ is unique, then $\mathbf{p}^r$ converges to that unique value. If the solution is not unique, then the results of Darroch and Ratcliff stated in Theorem 6 imply that $\mathbf{p}^r$ converges to the solution that minimizes $I(\mathbf{p}||\mathbf{p}^0)$.

6. Expectation-maximization

Let $\mathbf{p}$ = observed distribution. The problem addressed in this section is to find the distribution

$$\hat{\mathbf{q}} = \operatorname*{argmin}_{\mathbf{q}\in\mathbb{R}_+^m} I(\mathbf{p}||\mathbf{P}\mathbf{q})$$

This is equivalent to a nonparametric maximum-likelihood problem with log-likelihood given by

$$L(\mathbf{p}) = \sum_j p_j \log\left(\sum_k P_{j|k} q_k\right)$$

For this stochastic problem, the p_j may be relative frequencies of each outcome (n_j/n, where there are n trials) or simply the number of times a particular

outcome occurs (n_j). The probability distribution model for the data is assumed to be given in terms of some unknown underlying distribution $\mathbf{q}$ and known transition probabilities $\mathbf{P}$.

The variational representation for $I(\mathbf{p}||\mathbf{Pq})$ given by minimizing over $\mathbf{Q}$ motivates the alternating minimization algorithm that is the nonparametric expectation-maximization algorithm for this problem. In particular,

$$\hat{\mathbf{q}} = \operatorname*{argmin}_{\mathbf{q}\in\mathbb{R}_+^m} \min_{\mathbf{Q}\in\mathcal{Q}^\Delta} I(\mathbf{Q}\circ\mathbf{p}||\mathbf{P}\circ\mathbf{q})$$

This double minimization also leads directly to the algorithms for solving linear inverse problems subject to nonnegativity constraints: see Snyder, Schulz, and O'Sullivan [**22**], Byrne [**4**], and Vardi and Lee [**23**]. Richardson [**19**] derived this algorithm in 1972 for use in astronomical imaging; analysis for this application was extended by Lucy [**17**] in 1974.

Expectation-maximization algorithm:

- Choose an initial guess $\mathbf{q}^0 \in \mathbb{R}_+^m$, $q_k > 0$ for all k; set $r = 0$.
- E-Step

$$Q_{k|j}^r = \frac{q_k^r P_{j|k}}{\sum_k q_k^r P_{j|k}}$$

- M-Step

$$q_k^{r+1} = \frac{\sum_j Q_{k|j}^r p_j}{\sum_j P_{j|k}} = \frac{q_k^r}{\sum_j P_{j|k}} \sum_j \frac{P_{j|k} p_j}{\sum_k q_k^r P_{j|k}}$$

- If convergence is achieved, stop; else set $r = r+1$ and iterate.

If in fact $\mathbf{P} \in \mathcal{P}^\Delta$, then $\sum_j P_{j|k} = 1$ and the M-step simplifies to:

$$q_k^{r+1} = q_k^r \sum_j \frac{P_{j|k} p_j}{\sum_k q_k^r P_{j|k}}$$

This algorithm has a somewhat intuitive interpretation. The current guess $\mathbf{q}^r$ is used to predict the data through $\sum_k q_k^r P_{j|k}$. This is compared to the actual data $\mathbf{p}$ by taking the ratio between the two, then used to update the guess by back-projecting using $\mathbf{P}$. The iteration may also be written in a form that explicitly demonstrates that it is a gradient-descent type of algorithm:

$$q_k^{r+1} = q_k^r + \frac{q_k^r}{\sum_j P_{j|k}} \sum_j P_{j|k} \left(\frac{p_j - \sum_k q_k^r P_{j|k}}{\sum_k q_k^r P_{j|k}} \right)$$

One final way to rewrite the iteration is helpful when comparing the algorithm to Blahut's rate-distortion algorithm. Let

$$\tilde{q}_k^r = q_k^r \sum_j P_{j|k}$$

and

$$\tilde{P}_{j|k} = \frac{P_{j|k}}{\sum_{j'} P_{j'|k}}$$

Then

$$\tilde{q}_k^{r+1} = \tilde{q}_k^r \sum_j \frac{\tilde{P}_{j|k} p_j}{\sum_k \tilde{q}_k^r \tilde{P}_{j|k}} \tag{26}$$

To express the expectation-maximization algorithm as an A-algorithm, define

$$\begin{aligned} A(\mathbf{q}|\mathbf{q}^r) &= I(\mathbf{Q}(\mathbf{q}^r) \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}) \\ &= I(\mathbf{p}||\mathbf{P}\mathbf{q}^r) + \sum_j \sum_k \left(\frac{q_k^r P_{j|k}}{\sum_k q_k^r P_{j|k}} p_j \log \frac{q_k^r}{q_k} - P_{j|k} q_k^r + P_{j|k} q_k \right) \end{aligned}$$

Then

$$\mathbf{q}^{r+1} = \operatorname*{argmin}_{\mathbf{q}} A(\mathbf{q}|\mathbf{q}^r)$$

To outline a convergence proof, the form in (26) is examined. Note that for all $r \geq 1$, we have

$$\sum_k \tilde{q}_k^r = \sum_j p_j$$

Define

$$\begin{aligned} L(r+1, r) &= I(\mathbf{Q}^r \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}^{r+1}) \\ L(r, r) &= I(\mathbf{Q}^r \circ \mathbf{p} || \mathbf{P} \circ \mathbf{q}^r) = I(\mathbf{p}||\mathbf{P}\mathbf{q}^r) \end{aligned}$$

Suppose that $\mathbf{q}^*$ achieves $\min_{\mathbf{q} \in \mathbb{R}_+^m} I(\mathbf{p}||\mathbf{P}\mathbf{q})$, and denote the minimum by L^*. The inequalities

$$L(0,0) \geq L(1,0) \geq L(1,1) \geq \cdots \geq L(r,r) \geq L(r+1,r) \geq \cdots \geq L^*$$

hold as before. Examine the sum

$$\begin{aligned} \sum_k q_k^* \log \frac{q_k^{r+1}}{q_k^r} &= \sum_k q_k^* \log \sum_j P_{j|k} \frac{p_j}{\sum_{k'} P_{j|k'} q_{k'}^r} \\ &= \sum_k q_k^* \sum_j P_{j|k}^* \log \frac{P_{j|k}^* P_{j|k}^{r+1}}{P_{j|k}^* P_{j|k}^{r+1}} \sum_j P_{j|k} \frac{p_j}{\sum_{k'} P_{j|k'} q_{k'}^r} \end{aligned} \tag{27}$$

where

$$P_{j|k}^{r+1} = \frac{Q_{k|j}^r p_j}{q_k^{r+1}}$$

Key facts used here are

$$\sum_k q_k^{r+1} P_{j|k}^{r+1} = p_j \qquad \text{and} \qquad q_k^* \sum_j P_{j|k}^* = q_k^*$$

where the second equality is a version of the Kuhn-Tucker conditions due to the fact that $\mathbf{q}^*$ achieves the minimum. Continuing the expansion in (27), we have

$$\sum_k q_k^* \log \frac{q_k^{r+1}}{q_k^r} = L(r,r) + I(\mathbf{P}^* \circ \mathbf{q}^* || \mathbf{P}^{r+1} \circ \mathbf{q}^*) - L^*$$

This gives the inequality

$$L(r,r) - L^* \leq \sum_k q_k^* \log \frac{q_k^{r+1}}{q_k^r}$$

Summing both sides, we see that $L(r,r)$ converges to L^* and that $\mathbf{P}^{r+1}$ converges to $\mathbf{P}^*$. For more details on the proof of convergence, see [**3, 22, 24, 25**].

More typically, in estimation problems, there is some parametric description of the model $\mathbf{q}(\theta)$. The goal is then to estimate θ. The variational representation and therefore the equation for $\mathbf{Q}^r$ are the same. However, finding θ may be difficult. The A-algorithm formulation may be used to analyze the convergence properties. This situation is dealt with in [**12, 13, 18, 25**].

7. Blahut's rate-distortion algorithm

The algorithm for computing the rate-distortion function was presented in Blahut's paper [**2**] in 1972. Its derivation uses a variational representation of the mutual information function that is distinct from that used to derive the Blahut-Arimoto algorithm. Nevertheless, this algorithm often is mistakenly referred to as a Blahut-Arimoto algorithm.

Throughout this section, the probability distribution model is assumed. We let $\rho_{jk} \geq 0$ be a distortion measure. The rate-distortion function is defined as

$$R(D) = \min_{\mathbf{Q} \in \mathbf{Q}_D} I(\mathbf{p}, \mathbf{Q}) = \min_{\mathbf{Q} \in \mathbf{Q}_D} \min_{\mathbf{q} \in \Pi^m} \sum_j \sum_k p_j Q_{k|j} \log \frac{Q_{k|j}}{q_k} \tag{28}$$

where $\mathbf{Q}_D$ is the set of transition probabilities that satisfy the distortion constraint, namely:

$$\mathbf{Q}_D = \left\{ \mathbf{Q} \in \mathcal{Q}^\Delta : \sum_k \sum_j p_j Q_{k|j} \rho_{jk} \leq D \right\}$$

This formulation leads to Blahut's algorithm for computing the rate-distortion function. The algorithm is an alternating minimization algorithm, which alternates between minimizing (28) over $\mathbf{q}$ and over $\mathbf{Q} \in \mathbf{Q}_D$. A Lagrange multiplier s is introduced and the iteration proceeds for that multiplier. Each value of the Lagrange multiplier s corresponds to a specific distortion constraint D_s.

By varying s, the complete rate-distortion curve is obtained. Thus the Lagrange multiplier trades off distortion D for rate $R(D)$. For a specific s, we have

$$\begin{aligned} R(D_s) &= \min_{\mathbf{Q}\in\mathcal{Q}^\Delta} \sum_j \sum_k p_j Q_{k|j} \log \frac{Q_{k|j}}{\sum_{j'} Q_{k|j'} p_{j'}} - s\left(\sum_j \sum_k p_j Q_{k|j}\rho_{jk} - D_s\right) \\ &= sD_s + \min_{\mathbf{Q}\in\mathcal{Q}^\Delta} \min_{\mathbf{q}\in\Pi^m} \sum_j \sum_k p_j Q_{k|j} \log \frac{Q_{k|j}}{q_k e^{s\rho_{jk}}} \end{aligned}$$

Blahut's rate-distortion algorithm:

- Choose an initial guess $\mathbf{q}^0 \in \Pi^m$, $q_k > 0$ for all k; set $r = 0$.
- **Q-step** (or channel estimation step)

$$Q^r_{k|j} = \frac{q^r_k e^{s\rho_{jk}}}{\sum_{k'} q^r_{k'} e^{s\rho_{jk'}}}$$

- **q-step** (or output probability step)

$$q^{r+1}_k = \sum_j Q^r_{k|j} p_j = q^r_k \sum_j \frac{e^{s\rho_{jk}} p_j}{\sum_{k'} e^{s\rho_{jk'}} q^r_{k'}} \tag{29}$$

- If convergence is achieved, stop; else set $r = r+1$ and iterate.

It is interesting to note a difference between Blahut's rate-distortion algorithm and the expectation-maximization algorithm. In this regard, observe that in (29), there is no $\sum_j e^{s\rho_{jk}}$ in the denominator multiplying each term. When the expectation-maximization algorithm is written in terms of iterations on the product $\sum_j P_{j|k} q^r_k$, as in (26), the iterations are the same as those in Blahut's rate-distortion algorithm and the convergence analysis is equivalent.

A connection between the two algorithms occurs if the sum $\sum_j e^{s\rho_{jk}}$ is independent of k. Then, denoting this constant by b, Blahut's rate-distortion algorithm may be written in terms of the transition probabilities

$$P_{j|k} = \frac{e^{s\rho_{jk}}}{b}$$

and the algorithm is the same as the expectation-maximization algorithm. This constant sum condition occurs frequently in problems due to symmetry.

The A-algorithm formulation uses

$$A(\mathbf{q}|\mathbf{q}^r) = \sum_k p_j \frac{q_k^r e^{s\rho_{jk}}}{\sum_{k'} q_{k'}^r e^{s\rho_{jk'}}} \log \frac{\frac{q_k^r}{\sum_{k'} q_{k'}^r e^{s\rho_{jk'}}}}{q_k}$$

Then the algorithm may be written as $q_k^{r+1} = \operatorname{argmin}_{\mathbf{q}} A(\mathbf{q}|\mathbf{q}^r)$.

8. Convergence

The convergence analysis of the algorithms has been outlined. Blahut's original paper [**2**] contained a gap in the proof of convergence of his rate-distortion algorithm, that he closed in his thesis. Csiszár [**6**] also presented a complete proof of convergence. The proof of the convergence of the expectation-maximization algorithm in Dempster, Laird and Rubin [**12**] also had a gap. Because of its wide use in the estimation literature, the convergence of the expectation-maximization algorithm has been addressed extensively in the literature, starting with Wu [**25**]. See also [**13**] and [**24**].

Convergence analyses generally take one of two forms. In the first form, championed extensively by Csiszár [**7, 10, 8**], geometric arguments are used to show convergence. To get the rates of convergence, the algorithm is usually written in a second form as an A-algorithm, namely:

$$\hat{\boldsymbol{x}}^{r+1} = \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} A(\boldsymbol{x}|\boldsymbol{x}^r) \tag{30}$$

and properties of the function A are explored in order to show convergence [**25, 24, 13**]. The vector $\boldsymbol{x}$ in (30) is either $\mathbf{p}$ or $\mathbf{q}$ in one of the algorithms earlier.

9. Conclusions

Alternating minimization algorithms play a key role in information theory and estimation theory problems. Important papers appeared in 1972 introducing four alternating minimization algorithms. Common threads in the analysis of these algorithms are brought out herein, focusing on the I-divergence between a forward channel model and a backward channel model, as illustrated in Figure 1.

Acknowledgment. Don Snyder provided comments on an early draft of this paper and feedback on some of the results. Dick Blahut provided the motivation for this work, and has influenced my career in many positive ways.

References

[1] S. ARIMOTO, An algorithm for computing the capacity of an arbitrary discrete memoryless channel, *IEEE Trans. Inform. Theory*, vol. 18, pp. 14-20, 1972.

[2] R.E. BLAHUT, Computation of channel capacity and rate distortion functions, *IEEE Trans. Inform. Theory*, vol. 18, pp. 460–473, 1972.

[3] ———, *Principles and Practice of Information Theory*, Addison-Wesley 1987.

[4] C.L. BYRNE, Iterative image reconstruction algorithms based on cross-entropy minimization, *IEEE Trans. Image Processing*, vol. 2, pp. 96–103, 1993.

[5] T.M. COVER AND J.A. THOMAS, *Elements of Information Theory*, New York: Wiley 1991.

[6] I. CSISZÁR, On the computation of rate-distortion functions, *IEEE Trans. Inform. Theory*, vol. 20, pp. 122–124, 1974.

[7] ———, I-divergence geometry of probability distributions and minimization problems, *Annals of Probability*, vol. 3, pp. 146–158, 1975.

[8] ———, A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling, *Annals of Statistics*, vol. 17, pp. 1409–1413, 1989.

[9] ———, Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems, *Annals of Statistics*, vol. 19, pp. 2032–2066, 1991.

[10] I. CSISZÁR AND G. TUSNADY, Information geometry and alternating decisions, *Statistical Decisions*, Suppl. issue #1, pp. 205–207, 1984.

[11] J.N. DARROCH AND D. RATCLIFF, Generalized iterative scaling for log-linear models, *Annals Math. Statistics*, vol. 43, pp. 1470–1480, 1972.

[12] A.P. DEMPSTER, N.M. LAIRD, AND D.B. RUBIN, Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Stat. Society, Series B*, vol. 39, pp. 1–37, 1977.

[13] A.O. HERO AND J.A. FESSLER, Convergence in norm for alternating expectation-maximization (EM) type algorithms, *Statistica Sinica*, vol. 5, pp. 41–54, 1995.

[14] S. KULLBACK, *Information Theory and Statistics*, Wiley, New York, 1959.

[15] S. KULLBACK AND M.A. KHAIRAT, A note on minimum discrimination information, *Annals Math. Statistics*, vol. 37, pp. 279–280, 1966.

[16] S. KULLBACK AND R.A. LEIBLER, On information and sufficiency, *Annals Math. Statistics*, vol. 22, pp. 79–86, 1951.

[17] L.B. LUCY, Iterative technique for the rectification of observed distributions, *Astronomical Journal*, vol. 79, pp. 745–765, 1974.

[18] M.I. MILLER AND D.L. SNYDER, The role of likelihood and entropy in incomplete-data problems: Applications to estimating point-process intensities and Toeplitz and constrained covariances, *Proc. IEEE*, vol. 75, pp. 892–907, 1987.

[19] W.H. RICHARDSON, Bayesian-based iterative method of image restoration, *J. Optical Society of America*, vol. 62, pp. 55–59, 1972.

[20] L.A. SHEPP AND Y. VARDI, Maximum-likelihood reconstruction for emission tomography, *IEEE Trans. Medical Imaging*, vol. 1, pp. 113–122, 1982.

[21] J.E. SHORE AND R.W. JOHNSON, Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy, *IEEE Trans. Inform. Theory*, vol. 26, pp. 26–37, 1980.

[22] D.L. SNYDER, T.J. SCHULZ, AND J.A. O'SULLIVAN, Deblurring subject to nonnegativity constraints, *IEEE Trans. Signal Processing*, vol. 40, pp. 1143–1150, 1992.

[23] Y. VARDI AND D. LEE, From image deblurring to optimal investments: Maximum-likelihood solutions to positive linear inverse problems, *J. Royal Stat. Society, Series B*, pp. 569–612, 1993.

[24] Y. VARDI, L.A. SHEPP, AND L. KAUFMANN, A statistical model for positron emission tomography, *J. Amer. Statistical Society*, vol. 80, pp. 8–35, 1985.

[25] C.F.J. WU, On the convergence properties of the EM algorithm, *Annals Statistics*, vol. 11, pp. 95–103, 1983.

Information-Theoretic Reflections on PCM Voiceband Modems

Gottfried Ungerboeck

IBM Research Division
Zürich Research Laboratory
8803 Rüschlikon, Switzerland
gu@zurich.ibm.com

Abstract. This paper deals with the downstream transmission in "56 kbit/s" PCM modems. The channel consists of a digital 64 kbit/s PCM voice channel extending from a digital server modem to a local telephone office (LTO) and an analog subscriber line from the LTO to a client modem. We compute the capacity of the ideal PCM downstream channel, which is represented as a perfectly equalized AWGN channel over which A-law or μ-law symbol levels are transmitted at 8 kbaud. The Blahut–Arimoto algorithm is employed to determine the capacity-achieving symbol probabilities, either with or without a constraint on average symbol power. Coded-modulation methods for downstream transmission are still being studied. In the absence of agreement on a suitable coded-modulation method, first-generation PCM modems employ uncoded modulation and thereby do not achieve reliable 56 kbit/s transmission. We examine the performance gap between capacity-achieving modulation and uncoded modulation, and conclude that true 56 kbit/s transmission may become possible by applying suitable shaping and coding techniques. We also offer an observation that may stimulate further thought on the virtue of the quantization-noise avoidance approach adopted for PCM modems.

1. PCM modem background

The advent of "56 kbit/s" PCM voiceband modem technology was first announced by modem manufacturers in October of 1996. PCM modems exploit the specific telephone-network situation, which nowadays is usually encountered when an analog client modem communicates with a server modem of an Internet Service Provider (ISP). A 64 kbit/s PCM-channel connection will exist between the server modem and the client's local telephone office. Only the last-leg transmission over the client's local loop will be analog. This suggests that a server modem encodes information directly into a stream of 8-bit PCM-code octets at a rate of 8000 octets per second. In the local telephone office the PCM octets are converted into an analog signal by an A-law or μ-law codec. The signal is sent to the client modem over the analog subscriber line, and the client modem has to determine the sequence of transmitted PCM signal levels [4].

Analog subscriber lines introduce several signal impairments: spectral null at dc; nonlinear distortion at low frequencies; linear distortion; and additive noise. In addition, 64 kbit/s channels in the general telephone network are not necessarily fully transparent. Two "digital impairments" are encountered in the digital network: robbed bit signaling (RBS) and digital attenuation circuits (PADs). Moreover, country-specific constraints are in force, which limit the average power of the "analog signal content" of the digital signals. In view of these impairments and limitations, the industry goal for the downstream rate was set at 56 kbit/s. This industry goal has not yet been achieved.

A first limitation of the achievable downstream rate to less than 64 kbit/s stems from dc-free line coupling and nonlinear distortion introduced by line transformers at low frequencies. This requires a zero-dc constraint to hold for permissible sequences of PCM signal levels. Other limitations result from additive noise, residual echoes from the simultaneously transmitted upstream signal, residual distortion after equalization, and imperfect synchronization. The combined effect of these impairments prevents reliable decoding of closely spaced PCM signal levels around the origin. It is therefore necessary to eliminate some levels from the set of all possible PCM signal levels such that the remaining signal levels are spaced at least by some distance $d_{\min}$. The value of $d_{\min}$ depends on the measured signal–to–interference ratio. The digital impairments and constrained power of digital signals further reduce the number of PCM-code levels available for information encoding.

Prior to entering the data transmission phase, PCM modems perform a startup procedure. During this phase, the analog modem determines the set of modulation symbols to be used by the server modem.

The PCM modems currently on the market are based on two incompatible industry specifications called x2 and k56flex. In February 1998, technical agreement on "issue-1" of an international PCM modem standard was reached. The draft specification [2] is expected to become ratified as ITU Recommendation V.90. The x2, k56flex, and V.90 specifications define various forms of uncoded modulation for downstream transmission. For upstream transmission, V.34-like modulation and coding is used. Achieved downstream transmission rates are typically between 40 and 50 kbit/s. Rates marginally higher than 50 kbit/s are rarely obtained. Work on an advanced international PCM modem standard (issue-2) is still in a beginning phase. The specific form of coded-modulation that may be employed in issue-2 PCM modems remains to be determined. It seems unlikely that reliable downstream transmission at a rate of 56 kbit/s can be achieved without coded modulation.

The next section deals with the capacity of an ideal downstream channel. The results given for unconstrained signal power are equivalent to those given in [4]. We then present the capacity for constrained signal power and compare the performances obtained by capacity-achieving modulation and uncoded modulation. We conclude that there exists a significant margin for achieving shaping and coding gains, specifically in the case of constrained power.

In § 3 we offer another information-theoretic viewpoint on PCM modems, which is somewhat contrary to the current PCM modem philosophy of "quantization-noise avoidance" by allowing only PCM signal levels as modulation symbols. We are able to make our observation only for uniformly-spaced signals or uniform signal quantization. For a given constraint on average signal power, we show that the following two capacities are identical: the capacity of a noiseless channel with symbols chosen from a q-spaced integer lattice, and the capacity of an ideal channel with unconstrained inputs (except power) and additive white Gaussian noise (AWGN) with variance $q^2/12$.

2. Capacity of the ideal downstream channel

Figure 1 illustrates the assumed ideal discrete-time channel with discrete symbol inputs and continuous signal outputs. The transmitted symbols are chosen from a subset $A(d_{\min})$ of the set of 256 A-law or 255 μ-law PCM-code levels defined in ITU Recommendation G.711. Note that the signal spacing between A-law levels ranges from $d = 2$ for signals closest to the origin to $d = 128$ for the largest signals, whereas the spacing between μ-law levels ranges from $d = 2$ to $d = 256$. The signal set $A(d_{\min})$ is the largest subset obtained when PCM-code levels are removed such that the remaining levels have a distance of at least $d_{\min}$.

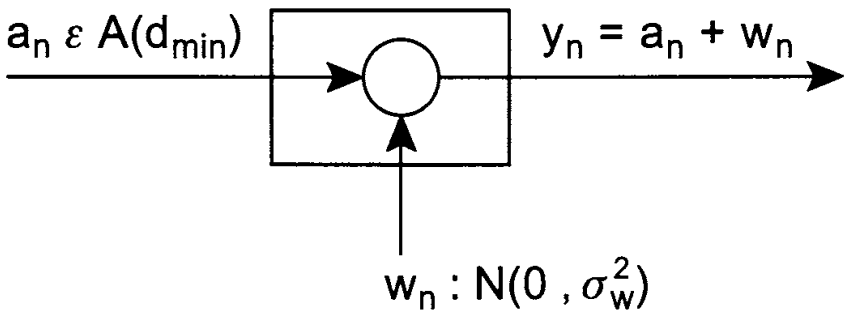


Figure 1. *Ideal downstream channel model*

ITU Rec. G.711, 256 A-law levels: {−4032,−3804,...,−5,−3,−1,1,3,5,...,3804,4032}
signal spacing: $d = 2, 4, 8, 16, 32, 64, 128$

ITU Rec. G.711, 255 μ-law levels: {−8031,−7775,...,−4,−2,±0,2,4,...,7775,8031}
signal spacing: $d = 2, 4, 8, 16, 32, 64, 128, 256$

Transmitted symbols are disturbed by AWGN with zero mean and variance σ_w^2. The noise power is expressed in terms of a 0dBm0-*to-noise ratio* $E_{0\text{dBm0}}/\sigma_w^2$, where $E_{0\text{dBm0}}$ is the symbol energy corresponding to a signal level of 0dBm0 as defined in G.711 (A-law: $E_{0\text{dBm0}} = 2853^2$, μ-law: $E_{0\text{dBm0}} = 5664^2$).

Let the set of modulation symbols be $A(d_{\min}) = \{\alpha_0, \alpha_1, \ldots, \alpha_{M-1}\}$, with $M = |A(d_{\min})| \leq 256$, and let p_i be the probability of sending the symbol α_i. Then the average energy of these symbols is $E = \sum_i p_i e_i$, where $e_i = \alpha_i^2$.

The capacity per channel use of the discrete-input/continuous-output channel, for an energy constraint $E \leq E^*$, is given by:

$$C = \max_{\boldsymbol{p} \in P^*} I(\boldsymbol{p}) = \max_{\boldsymbol{p} \in P^*} \sum_i p_i u_i$$

where:

$$u_i = -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{w^2}{2\sigma_w^2}\right) \log \sum_j p_j \exp\left(-\frac{(2w+\alpha_i-\alpha_j)(\alpha_i-\alpha_j)}{2\sigma_w^2}\right) dw$$

and P^* denotes the set of probability vectors $\boldsymbol{p}$, whose elements satisfy the constraints $p_i \geq 0$ for all i, $\sum_i p_i = 1$, and $\sum_i p_i e_i \leq E^*$. The calculation of the capacity-achieving probability vector $\boldsymbol{p}$ by the Blahut–Arimoto algorithm [**3, 1**] is briefly explained in an appendix. For more details on the Blahut-Arimoto algorithm, we refer the reader to the foregoing chapter by Joseph A. O'Sullivan.

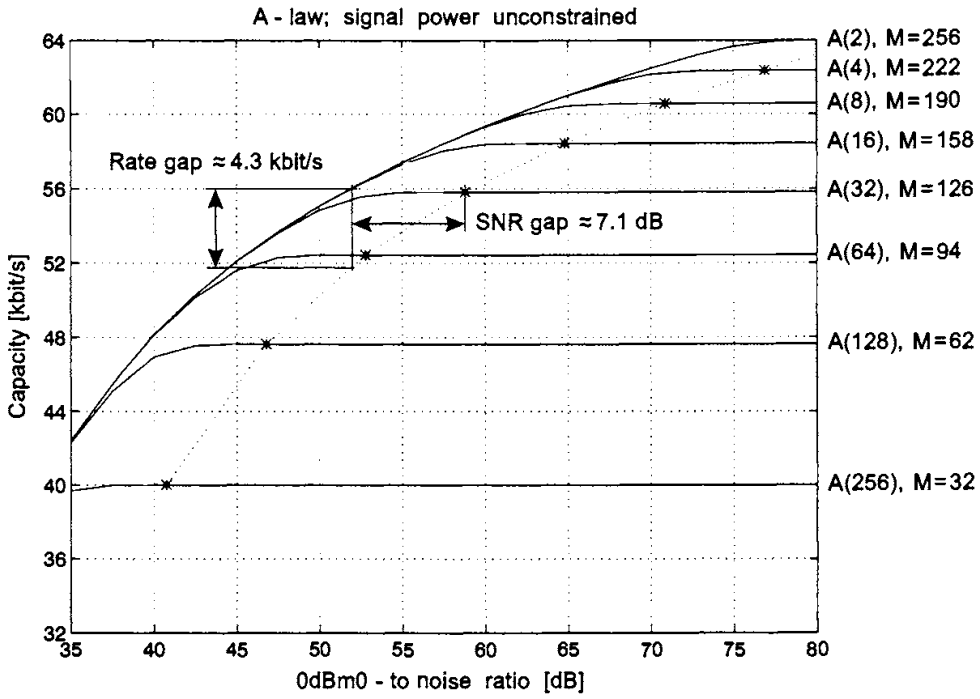


Figure 2. *Capacity of the ideal downstream channel for* $A(d_{\min})$ *with* $d_{\min} = 2, 4, \ldots, 256$

Symbol probabilities are optimized without power constraint. For uncoded modulation, asterisks (*) denote achievement of SER $= 10^{-6}$ with i.i.d. symbols.

Figure 2 shows capacity versus 0dBm0–to–noise ratio $E_{0\text{dBm0}}/\sigma_w^2$ for A-law signals with unconstrained power. Notice that the optimum symbol distribution

depends on $E_{0\text{dBm0}}/\sigma_w^2$. For lower noise, optimizing the capacity boils down to maximizing the symbol entropy. In the absence of a power constraint, an equal distribution is optimum. The performance gap between capacity-achieving modulation for 56 kbit/s and uncoded modulation with a symbol error rate of SER $= 10^{-6}$ is indicated as an SNR gap of approximately 7.1 dB for given rate, and as a rate gap of approximately 4.3 kbit/s for given ratio $E_{0\text{dBm0}}/\sigma_w^2$.

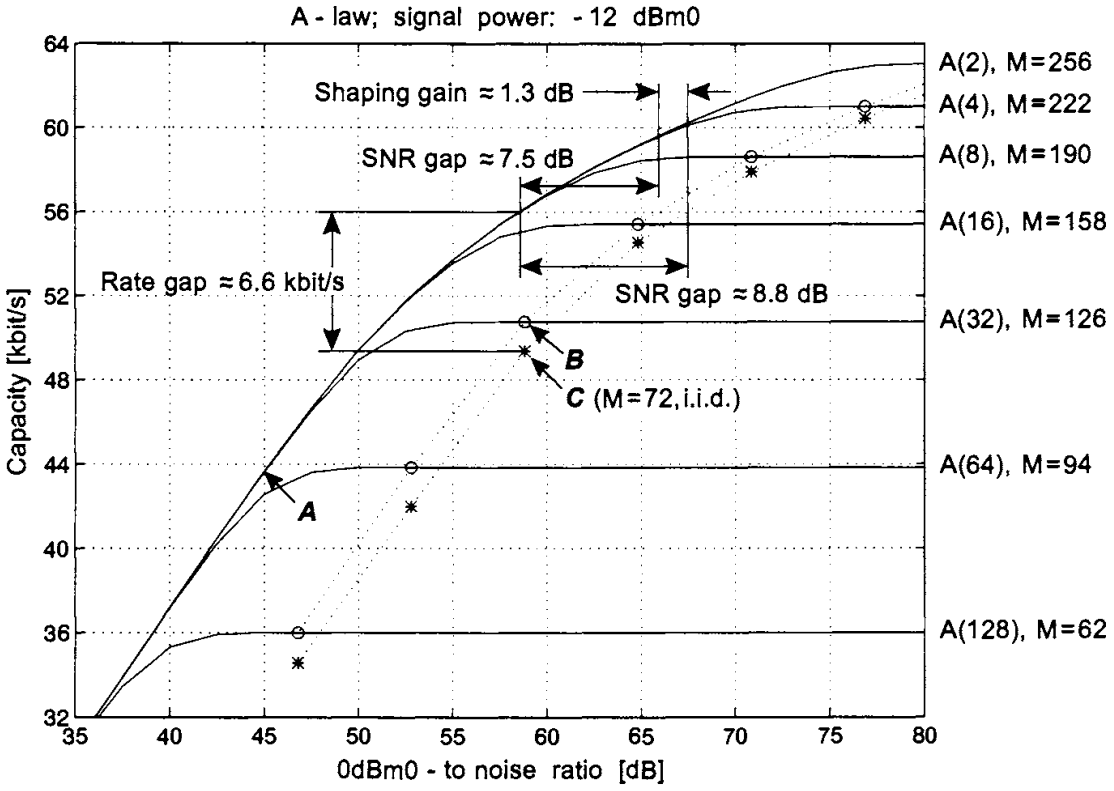


Figure 3. *Capacity of the ideal downstream channel for $A(d_{\min})$ with $d_{\min} = 2, 4, \ldots, 128$*

Symbol probabilities are optimized for $E^* = -12\,\text{dBm0}$. For uncoded modulation, achievement of SER $= 10^{-6}$ is indicated by circles (o) for optimally distributed symbols, and by asterisks (*) for i.i.d. symbols.

Figure 3 illustrates capacity versus $E_{0\text{dBm0}}/\sigma_w^2$ for A-law signals with a typical power constraint of $-12\,\text{dBm0}$. In this case the optimum symbol distribution will never be an equal distribution. A curve for uncoded modulation with equally likely symbols has therefore been added. The performance gaps relative to capacity-achieving modulation for 56 kbit/s are larger than observed in Figure 2. The SNR gap becomes approximately 7.5 dB for optimally distributed symbols and approximately 8.8 dB for i.i.d. symbols. The difference between these values indicates a shaping gain of about 1.3 dB, which is only modestly

smaller than the maximum shaping gain of 1.53 dB (namely $\pi e/6$) achievable with infinite constellations of lattice-type, or regularly spaced, signals.

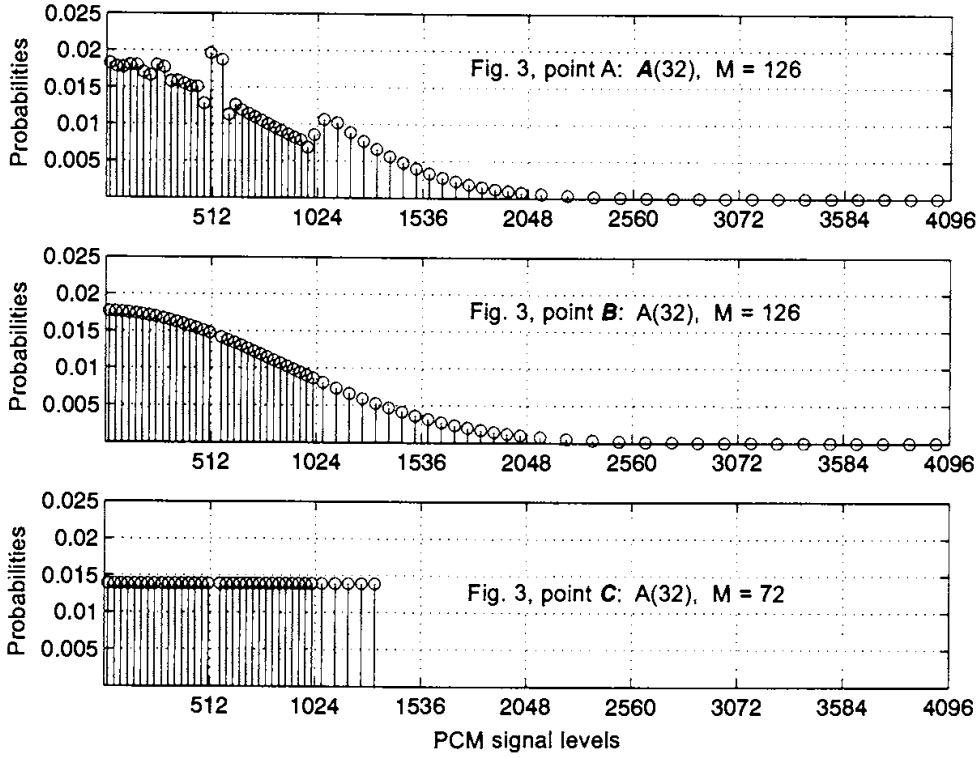


Figure 4. *Symbol distributions for positive symbols, for points A, B, and C*

Figure 4 shows the symbol distributions corresponding to the points marked A, B, and C in Figure 3. The somewhat irregular distribution of the symbol probabilities observed for point A reflects the fact that lower-valued signals in the symbol set $A(d_{\min} = 32)$ cannot always be spaced exactly by $d_{\min} = 32$. A tendency for higher probabilities can be seen, where distances between signal levels are locally larger than $d_{\min}$. The irregular probability distribution can no longer be observed for point B. At this point of lower noise, optimizing the capacity boils down to maximizing the symbol entropy subject to the power constraint. This yields the smoother Gaussian-like symbol distribution. Finally, for point C an i.i.d. distribution is found by definition.

3. Signal quantization and capacity

We show in Figures 5 and 6 that with channel Model 1 and Model 2, the same capacity is achieved in the limit for large signal constellations (high SNR). In both cases, the average signal energy is limited to E^*. We point out that the effect of the smearing/desmearing filters shown in Figure 6 could equivalently be obtained by multicarrier modulation.

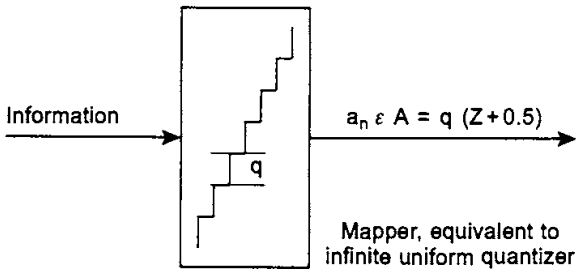


Figure 5.

Channel Model 1: *noiseless channel; quantization-noise avoidance by restricting modulation symbols to quantization levels*

a. Capacity for equiprobable symbols in a K-dimensional cube with sides $Q = N \times q$ is given by:

$$E^* = \frac{Q^2}{12} = \frac{N^2 q^2}{12}; \qquad C = \log N = \tfrac{1}{2} \log \left(\frac{12\, E^*}{q^2} \right)$$

b. Capacity for equiprobable symbols in a K-dimensional hypersphere with $K \to \infty$ is given by:

$$C = \tfrac{1}{2} \log \left(\frac{12\, E^*}{q^2} \right) + \tfrac{1}{2} \log \left(\frac{\pi e}{6} \right) \approx \tfrac{1}{2} \log \left(\frac{12\, E^*}{q^2} \right)$$

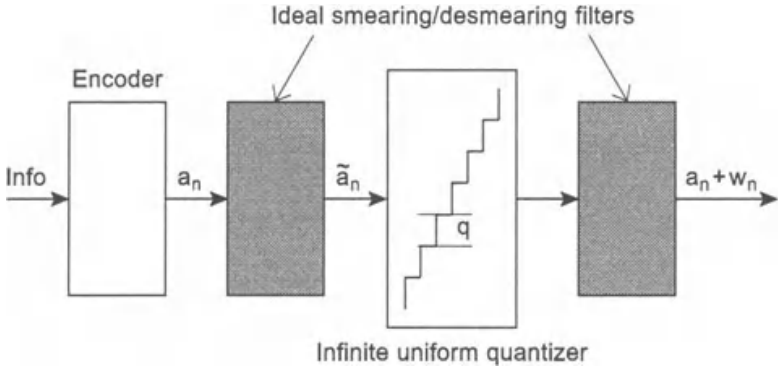


Figure 6.

Channel Model 2: *noiseless channel; modulation symbols are not restricted to quantization levels; quantization noise is converted to Gaussian noise*

c. Capacity for Gaussian channel inputs, (de-)smearing of quantization noise leads to additive Gaussian noise with variance $\sigma_w^2 = q^2/12$ is:

$$C = \tfrac{1}{2} \log \left(1 + \frac{12\, E^*}{q^2} \right) \approx \tfrac{1}{2} \log \left(\frac{12\, E^*}{q^2} \right)$$

4. Conclusions

In § 2 we have shown that the performance of future PCM voiceband modems could significantly be improved by introducing a coded-modulation method for achieving shaping and coding gains. The potential for performance improvements over uncoded modulation is higher when the power of permissible digital PCM signals is limited. Speculations concerning the significance of the observation made in § 3 about the equivalence of the two channel models in terms of channel capacity are left to the reader.

Appendix A. Computing the capacity-achieving symbol probabilities

For some initial probability vector $\boldsymbol{p}$, Lagrangian multiplier $s \geq 0$, and small quantities $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, the following algorithm is performed:

a. Compute u_i for all i (by numerical integration);

b. Compute $c_i = \exp(u_i - se_i)$ for all i, and set

$$p_i' = \frac{p_i c_i}{\sum_j p_j c_j}$$

c. Compute $E' = \sum_i p_i' e_i$; if $E'/E^* > 1 + \varepsilon_1$ or $E'/E^* < 1 - \varepsilon_1$ then adjust $s \leftarrow s\sqrt{E'/E^*}$ and goto (**b**);

d. Replace $p_i \leftarrow p_i'$ for all i; if

$$\left| \log c_{\max} - \log \sum_j p_j c_j \right| > \varepsilon_2$$

goto (**a**); else stop.

We have added step (**c**) to the original algorithm. The inclusion of this step adjusts the Lagrangian multiplier s automatically so that the desired symbol energy E^* is obtained directly, and not only as a function, for given values of s. In the special case where $s = 0$, no energy constraint is applied.

References

[1] S. ARIMOTO, An algorithm for computing the capacity of arbitrary discrete memoryless channels, *IEEE Trans. Inform. Theory*, vol. 18, pp. 14–20, 1972.

[2] Baseline text for V.pcm, ITU Study Group 16, WP 1, Temporary Document 17E, 1998.

[3] R.E. BLAHUT, Computation of channel capacity and rate-distortion functions, *IEEE Trans. Inform. Theory*, vol. 18, pp. 460–473, 1972.

[4] P.A. HUMBLET AND M.G. TROULIS, The information driveway, *IEEE Comm. Magazine*, vol. 34, pp. 64–68, 1996.

Some Wireless Networking Problems with a Theoretical Conscience

Anthony Ephremides

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF MARYLAND
COLLEGE PARK, MD 20742, USA
tony@eng.umd.edu

1. Introduction

Much of the current work on wireless networks focuses either on problems with (commercial or military) application urgency, such as cellular or ad-hoc networks, or on problems that concentrate on the physical layer, such as multiuser detection or fading channels. In this article, we examine a sampling of problems with a different character. They are pure network-layer problems in that they do not consider signal structures or formats but, rather, consider the signals as "black-box" packets, without regard to their internal structure. But they are also simplified and abstract versions of very complex real-world implementations that, hopefully, capture some of the essential conceptual aspects of wireless networking. Hence, as the title suggests, these problems have a theoretical "conscience." They are motivated either by recent attempts to exploit ways of increasing the "capacity" of multiple access channels or by, also recent, attempts to exploit the "covert" characteristics of the wireless medium.

The first problem considers the potential of antenna directivity to increase the maximum stable throughput of the collision channel. The second assesses the interplay between packet length (or bit-rate) and transmission power, again over the collision channel, and the result of that interaction on throughput and on energy savings. Finally, the third problem presents an example of the inherent covertness of the wireless medium, once again over the collision channel model, and its potential to support *subliminal* transmission.

2. Spatial directivity in the collision channel

The traditional analysis of the maximum stable throughput of the collision channel [1, 6], has considered a receiver with an omnidirectional antenna and an arbitrary number of transmitters distribed geographically (say, in two dimensions) around the receiver. If time is slotted and the users transmit indivisible

packets that fit snugly in the time slots, and if the provided feedback has the usual ternary structure (idle, success, collision), the maximum known achievable throughput, that does not cause unbounded transmission delays, is ~ 0.487.

It is natural to ask the question whether antenna directivity might increase this throughput value. In other words, if the receiver has the ability to focus on subsets of users that lie in specific sectors of the two-dimensional plane (in addition to focusing — according to the optimal first-come-first-served (FCFS) collision resolution algorithms — on subsets of users that generate their packets in specific sectors of the time-axis), then it appears plausible that throughput gains may be possible. Of course, we do not consider multiple sectorized receiver antennas that can simultaneously receive from different directions. The potential gains in that case are obvious. Instead, we simply consider a single antenna that can be pointed at will towards a chosen direction and so that the beamwidth can also be adjusted at will to a chosen value.

Specifically, consider, as in Figure 1, that the antenna is set at an angle α and that it has an ideal perfect cone pattern of angular width β.

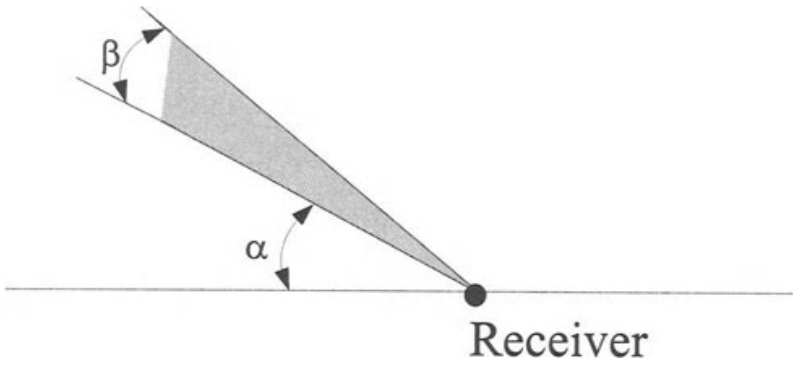


Figure 1.

Let us assume that in each slot the receiver may choose the values of α and β in addition to choosing the segment of the time axis in which arriving packets are enabled to be transmitted. A collision resolution algorithm may then be defined as a simple extension of the ones considered in [**1, 6**], so that at step k of a conflict resolution period α_k and β_k are updated as arbitrary functions of their previous values and of the received feedback. Thus, the methodology of [**6**] can be pursued in an entirely analogous way; that is, the optimal functions for the updating of α_k and β_k are to be found so as to maximize the throughput (that can be expressed in terms of those functions). The formulation fits naturally the policy iteration method of dynamic programming which yields the usual complicated equations, the numerical solution of which yields the optimum throughput value of ~ 0.487. These equations are not reproduced here for reasons of space economy, and because they can be found in [**6**]. The parallel formulation that we consider for the spatially expanded case does not change the structure of the equations and, as a result (as reported in [**8**]), the optimum throughput value remains unchanged at ~ 0.487.

This, somewhat surprising, result should actually be expected. To see this, consider the two-dimensional search space of Figure 2 in which the arriving packets are randomly scattered, as shown.

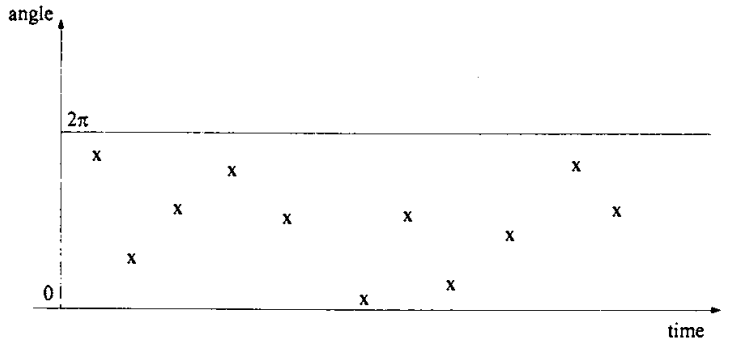


Figure 2.

The objective of any conflict resolution algorithm is to partition this space so as to isolate the packet positions (in time and in space). The traditional algorithms that correspond to omnidirectional antennas operate only in time and are thus restricted to explore partitions that involve only vertical stripes as in Figure 3.

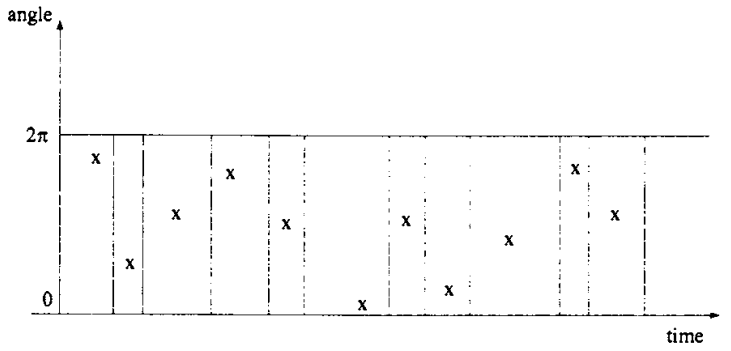


Figure 3.

In the case of a directive antenna, the algorithm has the freedom to perform this isolation by creating partitions in two dimensions, as shown in Figure 4.

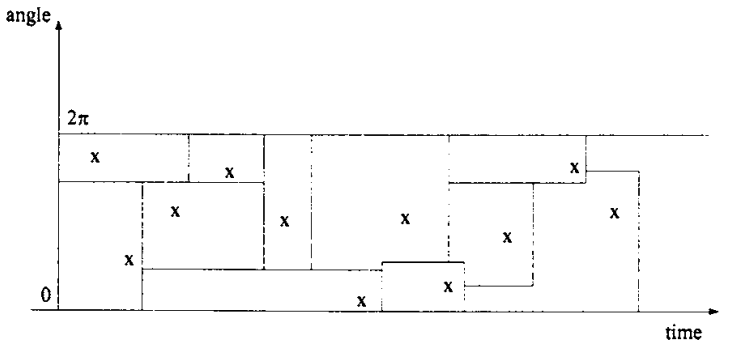


Figure 4.

It should be intuitively clear that, as long as the packet arrivals occur according to a "regular" process — that is, one in which the probability that two or more

arrivals occur in a segment of area δ is equal to $O(\delta^2)$, the number of partitioning steps needed to isolate the arrivals in the two-dimensional space (based on the ternary feedback of the collision channel) is independent of any restriction on the shape of the partitioning segments of the search. Thus the appearance of additional degrees of freedom in the search, as suggested by Figure 4, is misleading.

Although this simple result is negative, it clarifies (at the network level) the limitations of the advantage of antenna directivity. At the physical level, of course, it is another story. The differential delay at the antenna elements of the received signal can be used to separate the signals that arrive from different directions and thus the potential for improvement is very real and has been one of the bases of multiuser detection theory.

At the network level, the potential for improvement is thus confined not to the mode of searching the two-dimensional space, but, rather, to the use of multiple sectorized antennas. So, suppose that we have at our disposal M antennas at the receiver. The network layer question is then what beamwidth values $\beta_1, \beta_2, \ldots, \beta_M$ to assign to these antennas so that they span the 2π environment and so that they match the spatial distribution of the offered traffic. If λ_i is the average arrival rate of packets in the i-th sector, it is necessary that $\lambda_i < 0.487$, since the resolution of conflicts in that sector will be performed temporally. Thus, even though the total maximum stable throughput can be as high as $0.487\,M$, this value can be achieved only if the β_i's are chosen so as to satisfy the per-sector stability requirement.

How to match the sector widths to the offered traffic distribution is a question of load-balancing that can be answered by a variety of methods. Stochastic approximation methods are well-suited to the ternary-feedback collision channel model since the rate of observed collisions is a reliable estimate of the traffic intensity. Specific algorithms based on stochastic approximation are currently under development. Of interest is of course the rate of convergence to, and the stability of, the equalized sector values that result from these algorithms.

In conclusion, the usefulness of the negative result in [8] lies in shifting the efforts for capacity increase by means of directive antennas toward simultaneous detection rather than toward futile quests for "smart" search protocols.

3. Capture, throughput, and energy tradeoffs

The "capture" phenomenon in multiple access was observed very early in the history of wireless communications [7] and, gradually, it led to understanding the complexity of its causes. Capture (namely, correct reception) of one of several overlapping signals occurs either when one of the simultaneously transmitted signals arrives at the receiver with significantly higher power than the rest of the signals, or when an intelligent receiver (typically, one that is based on antenna phased arrays, or on CDMA, or both) takes advantage of signal structure differences and separates the desired signal from the rest. Thus signal capture depends on many parameters and in a very complicated way.

In network-layer studies of capture simpler models are considered; the intent is to assess the improvement in packet throughput when capture occurs. Thus it is typically assumed that capture is possible if the ratio of the desired signal's received power, or energy-per-bit, to that of the interfering ones (plus the noise) exceeds a detection threshold. In more simplified studies [**5**], it is assumed that the packet with the highest power at the receiver is correctly received (either with probability one or with a fixed probability less than one).

We consider here a model that goes a little further in drawing some signal structure features into the network layer model of capture. The motivation is the need to conserve energy in wireless communications. It is obvious that transmissions over multiple access collision channels that result in collisions waste valuable transmission energy (in addition to reducing throughput). Thus there is reason to make packets "shorter" (assuming constant offered rates of packets-per-second) in order to increase their separation and reduce thereby the likelihood of their overlap. However, if the same number of bits are packed in a shorter packet, transmission power must increase to maintain the per-bit received energy above the detectability threshold. It would appear, therefore, that all packets should be transmitted at peak-power.

The other side of the coin is that transmission at unequal power levels permits capture and thus works toward increasing throughput in a different way. So, the question is whether the gains from capture (that requires unequal power levels) outweigh the gains from packet separation (that require equal power levels). And, furthermore, there are two types of gains; gains in throughput and gains in throughput per expended energy unit.

Considering that m distinct power levels $p_1 > p_2 > \cdots > p_m$ are available and that an unspecified (therefore, infinite) number of users share the collision channel using the slotted ALOHA protocol, it has been shown in [**3**] that the achievable maximum throughput, assuming that the ALOHA protocol is stabilized via retransmission control, increases significantly relative to the case where a single power level is available. The reason is the capture phenomenon, which is assumed to occur when one of the contending packets uses a power level that is greater than that of its competitors with the proviso that the combined "other"-packet-plus-noise interference remains below the decodability threshold.

We follow a similar line of thinking except that we assume that if power level p_i is used then a corresponding packet duration or *length* L_i is used to keep E_b/N_0 constant. Thus, the number of bits per packet is assumed fixed (say, equal to M) and the total traffic rate in packets/sec at the i-th power level is also constant and equal to $\mathcal{G}_i$. At the highest power level p_1, the throughput $\mathcal{S}_1$ is equal to $\mathcal{G}_1 e^{-\mathcal{G}_1 L_1}$ if the criterion of capture is simply that the highest-powered packet wins, regardless of other-packet-interference levels. Notice that the throughput $\mathcal{S}_1$ is per absolute time unit, not per slot, since different slot lengths are used at each power level. Because the lower-power packets may collectively present unacceptably high interference for successful decoding this expression is only an upper-bound to the actual throughput. Notice, however, that if all power levels were equal (that is, if a single packet power level was used) the above expression would be exact, since in that case there would be no capture.

For lower power levels, the throughput upper bound can be tightened somewhat since for a packet at level i to be successful there must not be any packets at higher packet levels overlapping with the packet of interest. Thus we obtain

$$\mathcal{S}_i(\overline{\mathcal{G}}) \leq \mathcal{G}_i \exp\left(-\sum_{j=1}^{i} \mathcal{G}_j \max(L_i, L_j)\right)$$

where $\overline{\mathcal{G}} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m)$, and

$$\mathcal{S}(\overline{\mathcal{G}}) = \sum_{i=1}^{m} \mathcal{S}_i(\overline{\mathcal{G}}) \leq \sum_{i=1}^{m} \mathcal{G}_i \exp\left(-\sum_{j=1}^{i} \mathcal{G}_j \max(L_i, L_j)\right) \tag{1}$$

Our assumption that, for successful capture, the use of the highest power level among the contending packet is not a sufficient criterion, is complemented by the requirement that the signal-to-interference (plus noise) ratio be above a decodability threshold. For BPSK modulation and matched filter receivers this translates to the following (cf. [4]. For a packet at level p_i to be captured in the presence of another packet at level p_j, it is necessary that

$$\frac{p_i L_i}{p_j \min(L_i, L_j) + N_0 M} \geq T \tag{2}$$

where N_0 is the background noise level and T is a suitable threshold corresponding to the bit-error-rate requirement. It is implied that for a packet to be "successfully" decoded all bits (decided on a hard-decision basis) must be "successfully" decoded. If more sophisticated receiver structures, coding, and modulation schemes are to be used, the essence of the simple condition in (2) does not change, although the relationships among the relevant quantities will become undoubtedly much more complicated.

In [4], following-up on the model just described, it is shown that the the upper bound of the total throughput given by the right-hand side of (1) is maximized if $p_1 = p_2 = \cdots = p_m = p_{\max}$. But, in that case, the inequality of (1) becomes equality and thus we conclude that to maximize the actual achievable throughput $\mathcal{S}(\overline{\mathcal{G}})$, all power levels must be equal to the peak power level. That is, the gains from capture (as far as throughput is concerned) are outweighted by the gains from maximal packet separation (if the maximum power level is used, the packets are transmitted with minimum duration $L_{\min}$).

But we started out this analysis motivated by the need to reduce energy consumption. So far, we considered only effects on throughput. How does energy come in? The approach outlined above is well-suited to taking energy savings into consideration. The total average energy consumption per second — the average power consumption — is given by

$$P = \sum_{i=1}^{m} p_i \mathcal{G}_i L_i$$

For an energy-limited system, P must be bounded by a battery-dependent threshold, say $P \leq P_0$. Then, under this energy constraint, the objective is

to maximize the achievable throughput-per-energy unit. That is, we need to maximize the quantity

$$\mathcal{S}_P \stackrel{\text{def}}{=} \frac{\mathcal{S}(\overline{\mathcal{G}})}{P} \tag{3}$$

subject to $P \leq P_0$. Interestingly, an analysis that parallels the one that yielded the throughput maximization result also yields the condition that, to maximize the normalized throughput $\mathcal{S}_P$, it is again necessary to use a single power level for all packets, so that $p_1 = p_2 = \cdots = p_m$. The common value of these levels should be the lesser of the peak power value and the value that results from the constraint value P_0 in (3).

So, again, for energy-efficient maximum throughput, it is preferable to opt for packet separation in time rather than for benefits from capture.

It is interesting to note that in the analysis, there is a technical requirement that the distinct power levels (if distinct power levels are to be used) must be separated by a minimum finite amount that corresponds to the detection threshold T of (2) being greater than, or equal to, the quantity $(e-1) \cong 1.718$. Although the condition arises in a detailed technical context, it has a physical interpretation as well. The dual assumptions for successful capture, namely that the winning packet must have the highest power and, at the same time, have a signal-to-interference ratio value above a fixed threshold, start breaking down if the power levels can be arbitrarily close to each other (or, equivalently, if that threshold is arbitrarily small). They break down both mathematically and physically. Thus, the condition that $T > e - 1$, under which the quoted results can be proven, is not as restrictive as it might appear.

In conclusion, this analysis establishes a network-layer criterion for the choice of transmission power and packet length for energy conscious efficient communication. Again, the result is proven only for the case of the collision channel operating under the slotted ALOHA protocol.* However, the analysis makes a foray into the coupling of the network layer to the physical layer, for which traditional communication theory offers a rich set of analysis tools. For example, if complex signal constellations are assumed in the modulation process, the relationship of p_i to L_i for decodability becomes more intricate (rather than $p_i L_i = \text{const}$, as assumed here). Thus, the trade-off between capture and packet time separation becomes murkier. And if, furthermore, a multiuser detector is assumed (or, even if just CDMA signals are assumed), the trade-off becomes more complicated. However, its analysis might shed more light into the interplay among the communication layers (e.g., network layer and physical layer), which is a standard, albeit elusive, goal of current communication system research.

*The use of slotted ALOHA and different packet lengths imposes the further restriction the possible lengths be multiples of a common unit, which is almost equivalent to the restriction that the power levels be separated by a minimum fixed amount.

4. Subliminal collisions

For obvious reasons covert channels have been one of the centers of attention in the field of secure communications. A covert channel is a channel through which transmission of information is concealed to all but the designated receiver. Spread spectrum transmission is not covert because, even if the decoding of the transmitted information is adequately protected, the action of transmission is not. An example of covertness is the manipulation of the *timing* of acknowledgement (or other) information in multi-level security systems. Such covert channels have been studied in [**2**].

Of interest in a covert channel is of course the actual assurance and maintenance of covertness, the "capacity" of transmission (both in the Shannon-theoretic and communication-theoretic sense), its effects on the performance on the open system in which it is embedded, and, finally, the development and assessment of preventive measures against it.

The wireless medium, by its very nature (namely, its broadcast and multiaccess properties that open up its reach and accessibility), is well-suited to covert transmission. To illustrate this potential, we consider once more the collision channel and propose a covert transmission mechanism that can be concealed within a regular conflict resolution process. The proposed channel is somewhat idealized for two reasons. First, it is embedded in conflict resolution algorithms that are not known, as of yet, to be of practical significance. Second, it is sensitive to a number of key parameters of the algorithm and, as a result, would require some modification to improve its robustness. Nonetheless, it illustrates both the idea of convertness in wireless transmission, as well as practical possibilities, with an assessment of their potential and of their limitation.

So, consider the following situation. One of the participants in legitimate access to a collision channel, that use the FCFS collision resolution algorithm [**1**], intends to convey concealed information to a hidden receiver that simply monitors the collision channel. It does so as follows. By successively pretending it belongs to the set of users enabled to transmit by the algorithm in each slot, it can ensure that the number of collisions that will be observed in a conflict resolution period is at least equal to a number k of its choice, provided that at least one other legitimate user is attempting transmission. Of course, the actual number of collisions will be equal to another number c that satisfies $c \geq k$. Thus, the covert transmission is noisy.

Clearly, for this channel all the usual features of a physical channel are present. For, say, binary transmission, the covert user must select two values k_1, k_2, with $k_2 > k_1$, that represent "zero" and "one", respectively. For bit-error-rate reduction, the values of k_1, and k_2 must be chosen as far apart as possible. Thus, signal selection will affect the performance of the covert channel. However, large values of k lengthen the duration of the conflict resolution period and, hence, reduce the rate of covert transmission, since only one bit per conflict resolution period is transmitted. Thus, the trade-off involved in the choice of the k-values is analogous to the power-rate trade-off for combating noise in

ordinary communication channels. At the same time, large values of k have an adverse effect on the performance of the legitimate multiaccess transmission on the collision channel. By artificially lengthening the duration of each conflict resolution period, the covert user reduces the maximum stable throughput of the other users. In addition to this being an undesirable effect for the affected users, the reduction may lead to suspicion and, hence, detection of the covert action.

To assure that the covertness of the scheme remains unnoticed, the covert user must mimic the behavior of a legitimate user who has a packet to transmit as soon as the k artificial collisions are over. Thus, that user must pretend that a legitimate packet was generated at an arbitrary instant of his choice (within the interval enabled by the algorithm) and he must then follow the rules of the algorithm until the eventual successful transmission of that packet and the end of the conflict resolution period. If he doesn't, and if only a single legitimate user has been competing with him in that particular period, there will be no end to the conflict resolution period since at least two successful transmissions are needed for its termination. This is a small observation that shows how careful the designer of a covert protocol must be.

It should be clear that, for the quantitative analysis of the proposed channel, a crucial quantity that must be computed is the probability that c actual collisions occur in a conflict resolution period, given that k collisions are caused by the covert user. We denote this quantity by $f(c|k)$.

If α_0 is the chosen initial length of an enabled interval by the conflict resolution algorithm, and λ is the average rate of the (Poisson) collective process of packet generation, then the quantity $\mathcal{G}_j$, defined as:

$$\mathcal{G}_j \;=\; \alpha_0 \lambda 2^{-j}$$

represents the average traffic rate in the enabled fraction of that interval at the j-th stage of the algorithm. By carefully following the events and state transitions at each step of the algorithm, one can obtain the following expression for $f(c|k)$ for $k > 1$ and $c > k$:

$$f(c|k) \;=\; \frac{\mathcal{G}_{k-1}^2}{3(e^{\mathcal{G}_{k-1}} - \mathcal{G}_{k-1} - 1)} \sum_{n=0}^{\infty} \alpha_{c-k}(n)\, \mathcal{G}_{k-1}^n \qquad (4)$$

where:

$$\alpha_i(n) \;=\; \frac{1}{2^{n+2} - 1} \sum_{j=0}^{n} \frac{\alpha_{i-1}(j)}{(n-j)!}\, D_n(j)$$

with $\alpha_0(0) = 1$, $\alpha_0(j) = 0$ for $j > 0$, and $D_n(j)$ is given by:

$$D_n(j) \;=\; \begin{cases} 2 & j = n-1 \\ 1 & \text{otherwise} \end{cases}$$

Clearly, this expression can be only approximated computationally but does permit an assessment of the quality of the covert channel performance as a func-

tion of the values of its parameters. In fact, the truncation error in the summation of (4) can be upper-bounded quite tightly, so that $f(c|k)$ can be evaluated relatively accurately.

Furthermore, $f(c|k)$, $c > k$, can be shown to be a decreasing function of c for fixed k, and an increasing function of k for fixed c. This property permits an easy derivation of the maximum-likelihood decoding rule for the covert channel detector. That is, for given values of k_1, k_2 with $k_1 > k_2$), the decision rule is:

$$\text{decide } k_1, \quad \text{if } c < k_2$$
$$\text{decide } k_2, \quad \text{if } c \geq k_2$$

One needs to calculate further the probability of error (as a function of k_1 and k_2) and the average length of the conflict resolution period (again, as a function of k_1 and k_2). Thus, the information and bit-error rates of the covert channel can be calculated. Of interest, of course, is also the Shannon-theoretic capacity of the channel, but we do not address this question here.

These calculations can be straightforwardly generalized for higher-dimensional signal choices. At each channel use the signal can take any one of several values of k, say $k_1, k_2, \ldots, k_N$. Thus, the signal selection problem and the rate accuracy trade-off can become considerably more involved.

One also needs to calculate the impact of the covert use on the legitimate use of the channel. Clearly, the maximum stable throughput decreases from the value of 0.487 as a function of the k-values. A dramatic decrease will of course make the covert channel subject to detection. The objective of a good covert channel is to leave the system, within which it is embedded, intact, or affect it as little as possible.

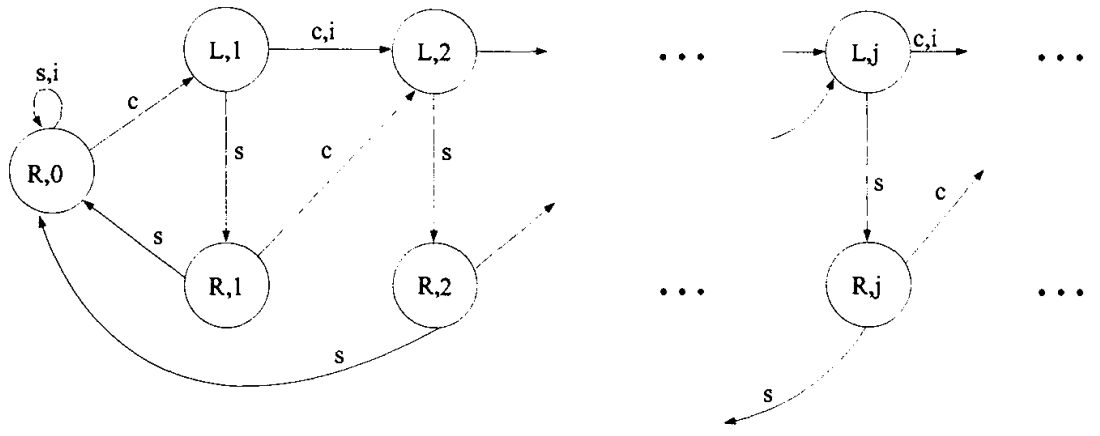


Figure 5.

The assessment of the impact on the main collision channel can be performed by following the analyses done in [6]. However, the Markov chain that described the operation of the FCFS algorithm needs to be somewhat modified. Following the notation in [1], the algorithm is in a "left" or a "right" state depending on the position it finds itself in the search tree as it keeps subdividing the enabled

interval. The Markov chain for the algorithm (without the covert user) and the associated state-transition events are shown in Figure 5. When the covert user operates this chain is modified as shown in in Figure 6.

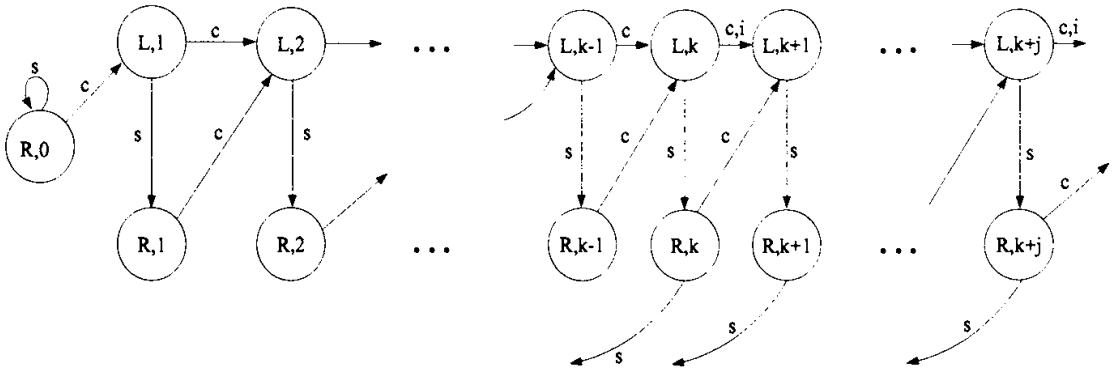


Figure 6.

In both cases the symbols s, c, and i denote the events of successful transmission, collision, or idle slot, respectively. The index $j = 0, 1, 2, \ldots$, that accompanies the L or R designation of the state represents the depth of the search tree in the execution of the algorithm.

The only difference between the dynamics of the two cases is that there can be no success while in state R, j and no idle when in state L, j for $j < k$ in the case of covert transmission. This difference is sufficiently significant to alter the performance of the algorithm but not significant enought to require a deviation from the method of analysis used for the non-covert case. This is especially so, since the state-transition probabilities are almost identical in both cases (and exactly identical for $j > k$).

Last, but not least, assessment of vulnerability to countermeasures needs to be performed. For the proposed channel, so long as the users agree to use an unaltered version of the standard FCFS conflict resolution algorithm, it appears that the covert operation is robust and fully concealed, except for the reduction in the value of the maximum stable throughput of the legitimate channel. This reduction may be debilitating, especially if the performance requirements of the covert channel, in terms of rate and accuracy, are high (resulting in large values of k).

This example of wireless covertness may be somewhat contrived but it does reveal a profile of what a covert channel is, and why the wireless environment lends itself to covert transmission. Much work still needs to be done to develop practical and realistic covert channels. And much work needs to be done to fully understand the behavior of even the simple and artificial example proposed here.

5. Conclusions

The three wireless networking problems that we described have certain characteristics in common: (i) they are all based on the collision channel, (ii) they have an intentionally reduced complexity in their definition so as to reveal the bare essentials of the particular focus that is placed upon each one of them, (iii) they are stripped of the physical layer aspects of wireless transmission so as to reveal what is important in wireless networking, and (iv) they address rather fundamental aspects of wireless multiuser operation.

Unlike much of current work on wireless communications, the sampling of these problems suggests that there is a rich repository of theoretical issues that surround wireless networking. It is this feature of our brief review that legitimizes its inclusion in this volume, which is dedicated to the work and accomplishments of Richard E. Blahut. Although they are outside the mainstream of his past work and interests, these problems share some of the principles and values that have characterized his research.

References

[1] D. BERTSEKAS AND R. GALLAGER, *Data Networks*, Englewood Cliffs: Prentice-Hall, 1992.

[2] M.H. KANG, I.S. MOSKOWITZ, AND D.C. LEE, A network pump, *IEEE Trans. Softw. Engineering*, vol. 22, pp. 329–337, May 1996.

[3] R.O. LAMAIRE, A. KRISHNA, AND M. ZORZI, Optimization of capture in multiple access radio systems with Rayleigh fading and random power levels, pp. 331–336 in *Multiaccess, Mobility, and Teletraffic for Personal Communications*, B. Jabburi, Ph. Godlewski, and X. Lagrange (Editors), Boston, MA: Kluwer Academic 1996.

[4] W. LUO AND A. EPHREMIDES, Effect of packet lengths and power levels on random access systems, in *Proc. 35-th Allerton Conference on Comm., Control, and Computing*, Monticello, IL., pp. 276–285, October 1997.

[5] J.L. MASSEY (Editor), special issue on Random-Access Communications, *IEEE Trans. Inform. Theory*, vol. 31, March 1985.

[6] J. MOSELY AND P. HUMBLET, A class of efficient contention resolution algorithms for multiple access channels, *IEEE Trans. Comm.*, vol. 33, pp. 145–151, 1985.

[7] L. ROBERTS, Aloha packet aystem with and without slots and capture, ASS Note 8, ARPA, Stanford Research Institute, June 1972.

[8] K. SAYRAFIANPOUR AND A. EPHREMIDES, Can spatial separation increase throughput of conflict resolution algorithms?, in *Proc. IEEE International Symp. Inform. Theory*, Ulm, Germany, June 1997.

Bounds on the Efficiency of Two-Stage Group Testing

Toby Berger

SCHOOL OF ELECTRICAL ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853, USA
berger@anise.ee.cornell.edu

James W. Mandell

DEPARTMENT OF NEUROSCIENCE
UNIVERSITY OF VIRGINIA MEDICAL SCHOOL
CHARLOTTESVILLE, VA 22903, USA
jwm2m@galen.med.virginia.edu

ABSTRACT. In modern biotechnology applications such as peptide and cDNA library screening and monoclonal antibody generation, significant savings in the number of screening assays frequently can be realized by employing sampling procedures based on group testing. Practical considerations often limit the number of stages of group testing that can be performed in such applications, thereby precluding the deeply nested strategies that have proved to be highly efficient in applications such as contention resolution in multiple access communication systems.

We provide exact formulas for the detection efficiency of some two-stage, three-stage and four-stage assaying strategies, including the popular rows-and-columns technique. Then we derive an upper bound on the maximum efficiency of any two-stage strategy, as opposed to classical bounds which apply only when the number of stages is unlimited. Finally, we use random test selection to show that there exist two-stage group testing protocols whose efficiencies are close to our upper bounds.

1. Introduction

Screening of peptide, cDNA, monoclonal antibody and combinatorial chemical libraries is a powerful but extremely laborious aspect of modern biotechnology. The advent of high throughput screening technologies has vastly accelerated the pace of library screening. We show that group testing provides an aid to the development of efficient algorithms for screening biological and chemical libraries that can enhance the power of both established and novel screening

Supported in part by NSF grants NCR-9216975, IRI-9005849, and IRI-9310670.

techniques. Although we embed our discussion in the context of monoclonal antibody hybridoma screening, our results also are applicable to other types of biochemical libraries including those cited above.

The traditional approach to monoclonal antibody hybridoma screening employs microtitier plates with 96 wells arranged in an 8 by 12 array into which potential antibody-producing cells are plated. By plating cells at limiting dilutions, single clones of antibody-producing cells can be obtained [**11, 10, 6, 15**]. The standard practice in most laboratories is to screen every well in which a colony has grown, despite the fact that few of the colonies produce antibodies of interest. A simple group testing procedure used by some workers to increase efficiency is to combine samples from the 8 wells of each column, giving 12 pooled samples, and likewise to combine samples from the 12 wells in each row of the array, giving 8 additional pooled samples. Assays are run on each of the 20 pooled samples. In all, then, Stage 1 consists of 8 row tests and 12 column tests for a total of 20 communal assays, also known as "group tests." Stage 2 comprises individually assaying each well located at the intersection of a row and column that both tested positive in Stage 1. We shall refer to this procedure as the *basic matrix method* (BMM). Because all 20 of the Stage 1 assays can be conducted in parallel, and likewise al the assays of Stage 2 can be performed simultaneously, we say the BMM is a two-stage method.

Two key properties of any testing procedure are its *efficiency* and its *number of stages*. Here, efficiency means the expected number of positive wells isolated per assay performed. Efficiency is the crucial parameter if laboratory man hours are the dominant consideration. However, if total elapsed time is what is most critical, then many laboratory technicians can be engaged to work in parallel, whereupon the number of stages becomes the limiting factor. The number of stages also may be limited by volatility of the material being tested. These considerations motivate our analysis in § 2 of the efficiencies of the BMM and of sc- -e new group testing algorithms that also have only a few stages, including a three-dimensional generalization in which the first-stage groups are planar sections of a cubic array. Modest increases in efficiency are realized. All the algorithms in question are statistically robust in the sense that they do not require that we know the value of the probability p that an individual active well is antibody-producing.

Some of our algorithms require that fairly intricate calculations and manipulations be performed in order to generate the groups to test in the next stage as a function of results from earlier stages. Although said intricacy may render them impractical at present, it is anticipated that computers and robots will come to perform many of the screening tasks that now are done by laboratory technicians. In that event an algorithm's intricacy will become increasingly less important and its efficiency and stage count increasingly more important.

We begin § 3 by reviewing a classical upper bound on the efficiency attainable by algorithms that have no restriction on their number of stages. Optimum nested group testing with an unlimited number of stages effectively achieves this bound, provided one is allowed to make the sequence of groups tested depend not only

on the outcomes of earlier tests but also on p. In the monoclonal antibody application, however, one rarely knows p accurately and we have seen that often there are compelling reasons to limit the number of stages severely. To our knowledge the only prior work on performance bounds for two-stage algorithms is that of Knill [**9**], who has considered both combinatorial and probabilistic two-stage group testing. In the case of Bernoulli probabilistic group testing that we consider here, Knill reports that, if there are n wells in all and the number of pools in the first stage does not exceed $\beta^2 \ln(n) \log_2 n / \ln\ln(n)$, where β is a positive constant, then the expected number of second-stage tests has to exceed $n^{1-2\beta-o(1)}$. This asymptotic result, although interesting, is of such generality that it does not even depend on the Bernoulli intensity p. Also, Macula [**13**] provides results for two-stage adaptations of nonadaptive algorithms in which there are a "fixed and relatively small number of first stage pools."

Motivated by the above considerations, we have derived an upper bound on the maximum efficiency attainable by any two-stage algorithm. At $p = 0.01$, for example, where the BMM has an efficiency of 4.55% and the classical upper bound on efficiency is 12.38% for algorithms with an unconstrained number of stages, our bound establishes that no two-stage algorithm's efficiency can exceed 9.96%. By analyzing the ensemble average efficiency of two-stage group testing when the tests constituting the first stage are selected at random, we establish a lower bound on attainable efficiency. At $p = 0.01$ this bound assures us that there exists a two-stage group test that is 7.79% efficient. We also provide specialized versions of our upper and lower bounds applicable to the important subclass of two-stage group tests in which no attempt is made to resolve positive wells.

2. Some two, three, and four-stage methods

In this section, we examine the efficiency of the basic matrix method and the sparse matrix method. We also discuss enhancements to these methods that make it possible to save some tests and thereby improve efficiency.

2.1. The basic matrix method. There usually will be some wells in which no cell colony grows in the culture. We assume henceforth that any such inactive wells are replaced by active wells from another microtitier plate before the assaying procedure begins. This both simplifies analysis and increases efficiency.

We determine the efficiency of the basic matrix method in the general case of an M-row, N-column array of active wells and then specialize to 8×12 arrays. We assume Bernoulli-p statistics; i.e., we assume that each active well has probability p of containing antibody-producing hybridomas, independently from well to well. The parameter p is unknown but usually resides in the range $0.005 \le p \le 0.2$ in cases of practical interest.

For each row index $i = 1, 2, \ldots, M$ and each column index $j = 1, 2, \ldots, N$, we define $\mathcal{R}_i$ and $\mathcal{C}_j$ as follows. We set $\mathcal{R}_i = 1$ if the i-th row's group test is positive and $\mathcal{R}_i = 0$ if the i-th row's group test is negative. Similarly, we let $\mathcal{C}_j = 1$ if

the j-th column's group test is positive and $\mathcal{C}_j = 0$ if the j-th column's group test is negative. With this notation, let

$$\mathcal{R} \stackrel{\text{def}}{=} \mathcal{R}_1 + \mathcal{R}_2 + \cdots + \mathcal{R}_M$$

denote the random number of positive row group tests. Similarly, let

$$\mathcal{C} \stackrel{\text{def}}{=} \mathcal{C}_1 + \mathcal{C}_2 + \cdots + \mathcal{C}_N$$

denote the random number of positive column group tests. Let the operator $\mathsf{E}[\cdot]$ denote statistical expectation. Then the expected total number of assays performed in the matrix method is $M + N + \mathsf{E}[\mathcal{RC}]$, there being $M + N$ tests in Stage 1, and a random number $\mathcal{RC}$ of tests in Stage 2. The expected number of positive wells isolated by the procedure is MNp, because each of the MN wells is positive with probability p and every positive well gets identified. Accordingly, the efficiency is*

$$\eta = \frac{MNp}{M + N + \mathsf{E}[\mathcal{RC}]} \tag{1}$$

Since

$$\mathcal{RC} = \sum_{i=1}^{M} \mathcal{R}_i \cdot \sum_{j=1}^{N} \mathcal{C}_j = \sum_{i=1}^{M} \sum_{j=1}^{N} \mathcal{R}_i \mathcal{C}_j$$

we have:

$$\mathsf{E}[\mathcal{RC}] = \sum_{i=1}^{M} \sum_{j=1}^{N} \mathsf{E}[\mathcal{R}_i \mathcal{C}_j] = MN\mathsf{E}[\mathcal{R}_1 \mathcal{C}_1]$$

where in the last step we have used that fact that the well at the intersection of row 1 and column 1 is *a priori* no more or less likely to get tested individually during Stage 2 than is any other well.

To calculate $\mathsf{E}[\mathcal{R}_1 \mathcal{C}_1]$, note that $\mathcal{R}_1 \mathcal{C}_1 = 0$ unless $\mathcal{R}_1 = \mathcal{C}_1 = 1$, and therefore $\mathsf{E}[\mathcal{R}_1 \mathcal{C}_1] = \Pr[\mathcal{R}_1 = \mathcal{C}_1 = 1]$. We distinguish two disjoint events that comprise the event $[\mathcal{R}_1 = \mathcal{C}_1 = 1]$. The first is when the well at the intersection of row 1 and column 1 is antibody-producing and hence tests positive. The second is when this well is negative but at least one of the other $N-1$ wells in row 1 is positive and at least one of the other $M - 1$ wells in column 1 is positive. Hence

$$\mathsf{E}[\mathcal{R}_1 \mathcal{C}_1] = p + q(1 - q^{N-1})(1 - q^{M-1}) = 1 - q^M - q^N + q^{M+N-1}$$

and

$$\eta_B = \frac{MNp}{M + N + MN(1 - q^M - q^N + q^{M+N-1})} \tag{2}$$

where $q = 1 - p$. Figure 1 shows η_B for $M = 8$ and $N = 12$ as a function of p over the range of principal interest, namely $0.005 \leq p \leq 0.2$.

*Equation (1) specifies the efficiency to be a ratio of expectations rather than the expectation of a ratio. This is because in practical applications a laboratory usually performs the matrix method for many successive $M \times N$ microtitier plates, whereupon the long-run efficiency is indeed given by the indicated ratio of expectations.

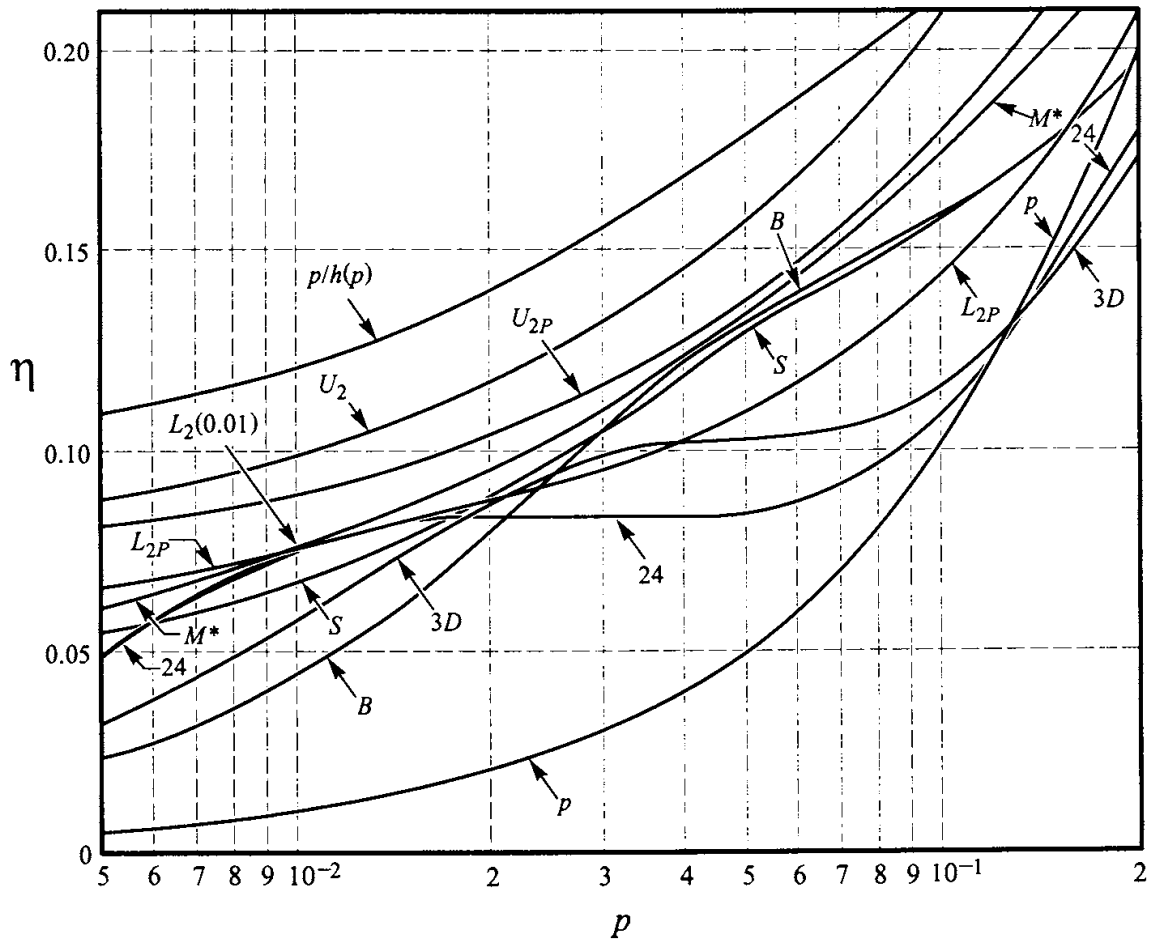


Figure 1. *Efficiencies and bounds on efficiency versus p*

$p/h(p)$: upper bound when number of stages is understricted

U_2: upper bound on thrifty two-stage algorithms

U_{2P}: upper bound on positive two-stage algorithms

$L_2(0.01)$: lower bound on thrifty two-stage algorithms, via random selection when $p = 0.01$

L_{2P}: lower bound on two-stage positive algorithms, via random selection: $(1 + e\log(q/p))^{-1}$

M^*: $M \times M$ matrix method with m optimized for each p

S: sparse 8×12 matrix method (three stages)

B: basic 8×12 matrix method

24: 24×24 matrix method

$3D$: $3D$ method with $L = M = 4$ and $N = 6$

p: individual testing of every well

2.2. The sparse matrix method. When p is small (say, 0.1 or less), which usually is the case after several levels of successive dilution in the cloning process, the BMM is highly inefficient. This is attributable to the fixed overhead associated with the $M + N$ tests always performed in Stage 1, most of which will be negative when p is small. If we are willing to increase the number of stages, we can improve the small-p efficiency. In particular, the probability that all the wells on the plate assay negatively is q^{MN}. For example $q^{MN} = 0.618$, when $p = 1 - q = 0.005$ and $MN = 96$, as in the case of an 8×12 plate. We can increase efficiency for parameter values such as these by adopting the following procedure, which we shall refer to as the *sparse matrix method* (SMM).

> **Stage 1:** *Group test the entire $M \times N$ plate. If the test is negative, proceed immediately to the next $M \times N$ plate.*
>
> **Stages 2 and 3:** *If the test of* Stage 1 *is positive, then perform the BMM on the plate.*

Although the SMM dispenses with wholly negative plates in one stage, it requires three stages for positive plates — one stage for the test of the whole plate plus two more to perform the BMM. Accordingly, the SMM is a three-stage method. In Appendix A, we derive the following expression for its efficiency:

$$\eta_S = \frac{MNp}{1 + (1 - q^{MN})(M + N) + MN(1 - q^M - q^N + q^{M+N-1})} \tag{3}$$

The efficiency of the SMM is plotted versus p in Figure 1 for $M = 8$ and $N = 12$. Comparing equations (2) and (3) reveals that the condition that must prevail in order for the SMM to be more efficient than the BMM is $q > (M + N)^{-MN}$, or equivalently

$$p < 1 - (M + N)^{-MN}$$

For the standard values $M = 8$ and $N = 12$, this calculates to $p < 0.0307$ (cf. the crossover point for η_B and η_S in Figure 1). Thus the sparse method is better than the basic method if (i) it is believed that dilution has been severe enough to reduce the probability that a randomly chosen well contains antibody-producing hybridomas to less than 3%, and (ii) it is acceptable to allot the time required to complete three stages of testing.

2.3. Thrift and enhancement. We now describe techniques for saving some tests and hence improving efficiency.

2.3.1. Thrifty BMM and thrifty SMM. Suppose that when we assay the rows we find that only one of them is positive (i.e., $\mathcal{R} = 1$). Then the wells located at the intersections of the C positive columns and this lone positive row must be positive, so it is not necessary to test them individually. That is, whenever $\mathcal{R} = 1$ we need not perform Stage 2 of the BMM. Similarly, if $C = 1$

we can dispense with Stage 2. We call the protocol that capitalizes on these savings the *thrifty BMM*. Its efficiency, derived in Appendix A, is given by:

$$\eta_{TB} = \frac{MNp}{M+N+MN\Big(1-q^M-q^N+q^{M+N-1}-pq^{MN}(q^{-M}+q^{-N}-q^{-1})\Big)}$$

Similarly, if the SMM is being used and in Stage 2 either $\mathcal{R}=1$ or $\mathcal{C}=1$, then it will not be necessary to perform Stage 3. We call the corresponding scheme the *thrifty SMM*; its efficiency, also derived in Appendix A, is given by:

$$\eta_{TS} = \frac{MNp}{1+(1-q^{MN})(M+N)+\cdots \quad \cdots+MN\Big(1-q^M-q^N+q^{M+N-1}-pq^{MN}(q^{-M}+q^{-N}-q^{-1})\Big)}$$

For $M=8$ and $N=12$, thrifty BMM and thrifty SMM provide only a miniscule improvement over BMM and SMM that disappears rapidly as p increases above 0.03. However, this improvement comes "for free" in the sense that one does not have to increase the number of stages in order to glean it. We revisit the concept of thriftiness when we derive our upper bound to the efficiency of two-stage tests in Appendix B.

2.3.2. Enhanced BMM and enhanced SMM. We can further enhance efficiency by suitably extending the concept of thriftiness in order to eliminate certain tests even when both $\mathcal{R}>1$ and $\mathcal{C}>1$. Henceforth, for $\mathcal{R}\geq 1$ (and, hence, $\mathcal{C}\geq 1$) let us call the first (i.e., lowest-indexed) positive row the *lead row* and call the first positive column the *lead column*. Note that any well in the lead row that has no other positive wells in its column must be positive. Likewise, any well in the lead column that has no other positive wells in its row must be positive. Accordingly, if in Stage 2 of the BMM we eschew testing individually any wells that belong either to the lead row or to the lead column, then in a third and final stage we will need to assay individually only those wells in the lead row (column) that have at least one other positive well in their column (row). We call the three-stage algorithm that does this the *enhanced BMM*. Analogously, *enhanced SMM* is a four-stage algorithm that eschews assaying individual wells in the lead row and lead column in Stage 3 of the SMM.

The efficiencies of enhanced BMM and enhanced SMM are derived in Appendix A, and the former is plotted as η_{EB} in Figure 1. We see that enhancement increases efficiency by a factor between 1.03 and 1.04 when $0.01\leq p\leq 0.1$. In other words enhancement eliminates between 3% and 4% of the assays that would had to have been performed were one using the BMM.

2.3.3. The 3D method. Imagine arranging the wells into an $L\times M\times N$ cubic array instead of the usual $M\times N$ planar array; in particular, if we take

$L = M = 4$ and $N = 6$, then there still will be 96 wells in all. The natural extension of the BMM to three dimensions, which we shall call the 3D *method*, is

Stage 1: *Test each of the L planar slices from front to back, the M planar slices from top to bottom, and the N planar slices from left to right.*

Stage 2: *Test individually each well located at the intersections of three planes that all tested positive in* Stage 1.

Of course, it is not really necessary to arrange the wells in a cube in order to conduct the 3D method. With 96 wells arranged in the usual 8×12 matrix, we can use the four 4×6 quadrants as the $L = 4$ front-to-back slices, the row pairs 1-and-5, 2-and-6, 3-and-7, and 4-and-8 as the $M = 4$ top-to-bottom slices, and the column pairs 1-and-7, 2-and-8, $\cdots$, and 6-and-12 as the $N = 6$ left-to-right slices. The 3D method uses a smaller number of tests in Stage 1 than does the BMM, for example, only $4 + 4 + 6 = 14$ in the instance cited as opposed to the $8 + 12 = 20$ Stage-1 tests of the BMM. Accordingly, we expect that the 3D method will outperform the BMM for small p. The efficiency of the 3D method, the derivation of which we omit, is given by:

$$\eta_{3D} = \frac{LMNp}{L+M+N+LMN\Big(1-q^{LM}-q^{LN}-q^{MN}+q^{L(M+N-1)}+\cdots \\ \cdots+q^{M(L+N-1)}+q^{N(L+M-1)}-q^{LM+LN+MN-L-M-N+1}\Big)}$$

A plot of η_{3D} for $L = M = 4$ and $N = 6$ appears in Figure 1. There we see that the 3D method outperforms the BMM for $p < 2.6\cdot10^{-2}$ but is inferior for larger values of p. Thrifty and enhanced versions of the 3D method exist, of course, but we have not studied them.

3. Bounds on efficiency

It is of interest to see how much the efficiency could be improved if we were willing to increase the number of stages substantially. Information theory allows us to upper bound the efficiency of any algorithm via the following argument. For our Bernoulli-p model* with n wells in all, the *a priori* uncertainty about which of the n wells are positive and which are not is $nh(p)$ bits, where

$$h(p) = -p\log_2 p - (1-p)\log_2(1-p)$$

is Shannon's binary entropy function. Each group test can provide at most 1 bit of information because its outcome is binary, so at least $Wh(p)$ are required to remove all the uncertainty and thereby fully resolve the situation. Since the ex-

*This is also known as a Binomial(n,p) model.

pected number of positive wells thus identified is Wp, no algorithm's efficiency can exceed $p/h(p)$. This "classical" bound on efficiency is the top curve in Figure 1. It tells us, for example, that an algorithm cannot be more than 12.38% efficient when $p = 0.01$. If one knows p and uses optimum nested group testing with a large number of stages, then it is possible to achieve an efficiency that is only a small fraction of a percent short of the classical bound [1]. The BMM's efficiency is only 4.55% at $p = 0.01$. The gap between BMM efficiency and the classical bound is attributable to two factors. One is that BMM does not assume knowledge of p and the other is that BMM is comprised of only two stages.

In Appendix B, we derive two upper bounds to the efficiency of two-stage algorithms. These also appear in Figure 1, labeled U_2 and U_{2P}. Bound U_{2P} applies to algorithms in which thriftiness in the sense of § 2 is not employed, that is, to the class of algorithms we call "positive" because they individually test in Stage 2 every well that was involved only in positive group tests in Stage 1. It tells us that no two-stage positive algorithm can have an efficiency greater than 9.06% at $p = 0.01$. Bound U_2 applies to *any* two-stage algorithm, even a thrifty one designed for a known value of p. It tells us that no two-stage algorithm's efficiency can exceed 9.96% when $p = 0.01$.

The upper bound derivation in Appendix B suggests that a good algorithm should make all the group tests in Stage 1 be the same size s, and that every well should be involved in the same number m of them. Moreover, s should be an integer near s^* and m should be an integer near $m^* = s^*\mathcal{R}^*$, where s^* and $\mathcal{R}^*$ are defined in Appendix B. One way of achieving this is to extend the 3D method described in § 2 to d-dimensional hypercubes whose faces each contain about s wells; that is, each edge of the hypercube has length $s^{1/(d-1)}$. However, such hypercube schemes have the serious weakness that they are relatively poor at resolving negatives in the sense of the following discussion.

3.1. Unresolved negatives and positives. A negative well is said to be unresolved if all the Stage-1 pools in which it is involved test positive. Any two-stage procedure required to classify every well correctly must test each of its unresolved negatives individually in Stage 2, so it is desirable to keep the number of unresolved negatives small. Since group testing is applied principally to situations in which positive wells are rare, controlling the number of unresolved negatives is a crucial consideration. A negative gets resolved if and only if at least one of the Stage-1 pools to which it belongs tests negative.

Let x denote a particular well, and let A and B denote two pools to which x belongs. If A tests positive, then all wells in A have an *a posteriori* probability of being positive that is higher than that of wells in the overall population. Accordingly, if several wells in A also belong to B, the positivity of A increases the conditional probability that B also will test positive. It follows that, if x actually is negative, it becomes less likely that x will be resolved by the combination of A and B than would be the case if x were the only well common to A and B. In the hypercube scheme, however, many pools share several wells in common. For example, in the 3D scheme the wells in the top row of the front

face also constitute the front row of the top face, so all the wells on this edge are common to the test of the front face and the test of the top face. This is conducive to leaving negatives unresolved and, hence, to inefficiency.

In the case of thrifty algorithms, there is a related concept of unresolved positives. Specifically, a thrifty algorithm avoids testing a positive well individually in Stage 2 if and only if there is a Stage-1 test that consists solely of this well and of negative wells none of which remains unresolved at the end of Stage 1. It follows that an experimental design which minimizes the expected number of unresolved negatives will tend to be effective against unresolved positives, too. Since group testing usually gets applied to situations in which the fraction of all wells that are positive is small, resolving positives cannot enhance efficiency appreciably and therefore is of secondary importance relative to resolution of negatives. Moreover, Appendix C reveals that mathematical analysis of the resolution of positives is considerably more challenging than that for negatives.

One way to make sure that pairs of pools do not overlap significantly is to choose the pools to be blocks in a design. Specifically, if we use a quasi-symmetric 2-design with $x = 0$ and $y = 1$ in the notation of [**2**, Chapter 3], then any pair of pools will contain at most one well in common, which we have seen is conducive to the resolution of negatives and, hence, also of positives. However, the number b of pools (blocks) in a quasi-symmetric design is always at least as great as the number n of wells (points); see [**2**, p. 25] for more on designs. Therefore, we would need to select a subfamily of the pools in the design to serve as our Stage-1 group tests; otherwise, it would be more efficient just to test every well individually in Stage 1. The problem of figuring out which blocks to leave out of a quasi-symmetric design, call it Q, in order for those that remain to constitute the first stage of an efficient two-stage group testing procedure is challenging. One approach we currently are investigating is to use the dual of Q, i.e., the structure whose incidence matrix is the transpose of that of Q. This approach conforms with the requirement for fewer pools than wells and also assures that no two wells are common to more than one pool, a property that should be conducive to confining the population of unresolved negatives.

Although at present we are unable to construct efficient two-stage group testing procedures explicitly, we have succeeded in using the information-theoretic technique of random selection to prove that there exist two-stage group tests with high efficiencies. Specifically, the analysis of randomly selected group tests carried out in Appendix C leads to the conclusion that there are two-stage positive group testing procedures whose efficiencies exceed the lower bound L_{2P} plotted in Figure 1 and two-stage thrifty group testing procedures whose efficiencies exceed the lower bound L_2 also plotted there. Since L_{2P} (respectively, L_2) is closer to our upper bound U_{2P} (respectively, U_2) than to the efficiencies of the widely used BMM and enhancements thereof that we treated in § 2, we conclude that these popular procedures are relatively inefficient even when a restriction to two-stage procedures is imposed. For example, at $p = 0.01$, the efficiency of the best two-stage positive group testing scheme lies between $L_{2P}(0.01) \approx 0.0728$ and $U_{2P}(0.01) \approx 0.0909$, whereas the BMM achieves only 0.0455.

Appendix A. Efficiency of variants of the matrix method

In this appendix, we derive the expressions for the efficiency of the sparse matrix method, as well as the thrifty and enhanced variants of both BMM and SMM.

A.1. Efficiency of the sparse matrix method. When the SMM is used, the expected number of assays performed per plate is

$$1 + \Pr(\mathcal{RC} > 0)\big(M + N + \mathsf{E}[\mathcal{RC}|\mathcal{RC} > 0]\big)$$

Now, $\Pr(\mathcal{RC} > 0) = 1 - q^{MN}$ and we have

$$\begin{aligned}\mathsf{E}[\mathcal{RC}|\mathcal{RC} > 0] &= \sum_{i=1}^{MN} i\,\frac{\Pr(\mathcal{RC} = i)\Pr(\mathcal{RC} > 0|\mathcal{RC} = i)}{\Pr(\mathcal{RC} > 0)} \\ &= \sum_{i=1}^{MN} i\,\frac{\Pr(\mathcal{RC} = i)\cdot 1}{\Pr(\mathcal{RC} > 0)} = \frac{\mathsf{E}[\mathcal{RC}]}{\Pr(\mathcal{RC} > 0)}\end{aligned}$$

It follows, with reference to the derivation of equation (2), that the efficiency of the SMM is given by:

$$\eta_S = \frac{MNp}{1 + (1 - q^{MN})(M + N) + MN(1 - q^M - q^N + q^{M+N-1})}$$

A.2. Efficiencies of thrifty BMM and thrifty SMM. The average number of assays avoided each time we find that $\mathcal{R} = 1$ is $\mathsf{E}[\mathcal{RC}|\mathcal{R} = 1] = \mathsf{E}[\mathcal{C}|\mathcal{R} = 1]$, so the expected number of assays saved by there being only one positive row is $\Pr[\mathcal{R} = 1]\,\mathsf{E}[\mathcal{C}|\mathcal{R} = 1]$. The first factor here is

$$\Pr[\mathcal{R} = 1] = M\Pr[\mathcal{R}_1{=}1, \mathcal{R}_2{=}0, \ldots, \mathcal{R}_M{=}0] = M(1 - q^N)q^{(M-1)N}$$

To give an expression for the second factor, let $\mathfrak{A}$ denote the event that exactly k of the N wells in the lone positive row are positive, and let $\mathfrak{B}$ be the event that at least one well in said row is positive. Then the second factor is:

$$\begin{aligned}\mathsf{E}[\mathcal{C}|\mathcal{R} = 1] &= \sum_{k=1}^{N} k\Pr[\mathcal{C} = k|\mathcal{R} = 1] = \sum_{k=1}^{N} k\Pr[\mathfrak{A}|\mathfrak{B}] \\ &= \sum_{k=1}^{N} k\,\frac{\binom{N}{k}p^k q^{N-k}}{1 - q^N} = \frac{Np}{1 - q^N}\end{aligned}$$

Thus, the expected number of assays saved by there being only one positive row is given by:

$$\frac{M(1 - q^N)q^{(M-1)N}Np}{1 - q^N} = MNpq^{(M-1)N}$$

By symmetry, the expected number of assays saved by there being only one positive column is $MNpq^{(N-1)M}$. The total expected number of tests saved by reduction is the sum of these minus a correction to account for the fact that we

have double-counted the saving of one test that occurs whenever $\mathcal{R}$ and $\mathcal{C}$ are both equal to 1. Since

$$\Pr[\mathcal{R} = \mathcal{C} = 1] \;=\; MN \Pr[\,(1,1) \text{ is the lone positive well}\,] \;=\; MNpq^{MN-1}$$

we conclude that the total expected number of tests that are avoided by being thrifty is given by:

$$MNpq^{MN}\left(q^{-M} + q^{-N} - q^{-1}\right)$$

This establishes why this is the term one subtracts in the denominator in order to convert the formula for η_B to that for η_{TB}. Similarly, η_{TS} is obtained from η_S by subtracting this same term in the denominator.

A.3. Efficiencies of enhanced BMM and enhanced SMM. The probability that row j is the lead row is $(1-q^N)q^{(j-1)N}$ for $j = 1, 2, \ldots, M$. Given that there is a lead row (i.e., given at least one well in the plate is positive), the expected number of positive wells in it is $Np/(1-q^N)$. When row j is the lead row, the probability that there are no other positive wells in the same column as a positive well in the lead row is q^{M-j}. It follows that the expected number of tests saved by never having to test wells in the lead row that are the only positive wells in their columns is

$$\sum_{j=1}^{M}(1-q^N)q^{(j-1)N}\frac{Np}{1-q^N}\,q^{M-j} \;=\; Npq^{M-N}\sum_{j=1}^{M}q^{(N-1)j}$$
$$= \; Npq^{M-1}\frac{1-q^{(N-1)M}}{1-q^{N-1}}$$

We add to this $Mpq^{N-1}(1-q^{(M-1)N})/(1-q^{M-1})$ for the corresponding expected savings for wells in the lead column that need never be tested. This sum double-counts the well at the intersection of the lead row and the lead column when that well is both the only positive one in the lead row and the only positive one in the lead column. Equivalently, (j,k) gets double-counted if and only if it is the only positive well in the first j rows and the first k columns. The probability of this is

$$p \times q^{Nj+Mk-jk-1} \;=\; \left(\frac{p}{q}\right)q^{MN-(M-j)(N-k)}$$

Thus, the expected number of double-counts is

$$\left(\frac{p}{q}\right)q^{NM}\sum_{j=1}^{M}\sum_{k=1}^{N}q^{(M-j)(N-k)} \;=\; \left(\frac{p}{q}\right)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}q^{-mn}$$

Accordingly, the expected number of tests saved by enhancing either BMM or SMM is given by:

$$Npq^{M-1}\frac{1-q^{(N-1)M}}{1-q^{N-1}} \;+\; Mpq^{N-1}\frac{1-q^{(M-1)N}}{1-q^{M-1}} \;-\; \left(\frac{p}{q}\right)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}q^{-mn}$$

The expression for the efficiency of the enhanced BMM (respectively, the enhanced SMM) is then obtained by subtracting this quantity from the denominator of the expression for η_B (respectively, η_S).

Appendix B. Upper bounds to efficiency of two-stage testing

Given W wells each of which is positive with probability p independently of all the others, our task is to determine which of them are positive and which are not using only two stages of group testing. In what follows we upper bound the efficiency with which this can be done.

With any two-stage algorithm that correctly classifies all W wells, we associate three classes of wells as follows:

Class 1: The $K_1 \geq 0$ wells that are not involved in any Stage-1 tests.
Class 2: The $K_2 \geq 0$ wells that are tested individually in Stage 1.
Class 3: The K_3 wells each of which belongs to at least one group of size ≥ 2 that is tested in Stage 1.

Of course, we must have $K_1 + K_2 + K_3 = W$. The efficiency of the algorithm is Wp divided by the expected number of tests in Stages 1 and 2, call it $\mathsf{E}[T]$, so we can upper bound efficiency by lower bounding $\mathsf{E}[T]$.

Every Class 1 well has to be tested individually* in Stage 2. Also, without loss of generality we may assume that no Class 2 well is involved in any tests in either stage other than its own individual test in Stage 1.

Let us index the K_3 wells in Class 3 by $x = 1, 2, \ldots, K_3$ and denote by $n_s(x)$ the number of Stage-1 group tests of size s in which x is one of the s group members. The total number of tests of size 2 or more performed in Stage 1 is

$$\sum_{x=1}^{K_3} \sum_{s=2}^{K_3} \frac{n_s(x)}{s}$$

where the s in the denominator accounts for the fact that each group test of size s is counted s times as we sum over x. It follows that

$$\mathsf{E}[T] \;=\; K_1 + K_2 + \sum_{x=1}^{K_3} \sum_{s=2}^{K_3} \frac{n_s(x)}{s} \;+\; \sum_{x=1}^{K_3} \Pr(x \in \mathcal{G}2)$$

where $x \in \mathcal{G}2$ is the event that one of the tests in Stage 2 is an individual test of x. Letting x^+ denote the event that well x is positive and x^- denote the event that well x is negative, we have

$$\Pr(x \in \mathcal{G}2) \;=\; p \Pr\left(x \in \mathcal{G}2 \;\middle|\; x^+\right) \;+\; q \Pr\left(x \in \mathcal{G}2 \;\middle|\; x^-\right)$$

where $q = 1 - p$ is the probability that a randomly chosen well is negative. At this juncture we distinguish two classes of two-stage algorithms, namely *positive* algorithms and *thrifty* algorithms. The distinction is illustrated by the following simple example. Suppose that in Stage 1 the pair $\{a, b\}$ and the pair $\{b, c\}$ test positive and negative, respectively. Then we will know after Stage 1 that a must be positive. A thrifty algorithm would not test a individually in Stage 2. Positive algorithms, contrastingly, impose the requirement that every Class 3 well

*Instead of being tested individually, it could be tested in a group all other members of which are known to be negative, but this does not affect our derivation.

that was involved only in positive tests in Stage 1 must be tested individually in Stage 2. Hence, a positive algorithm would waste a Stage-2 test on $\{a\}$ in the above scenario. Thrifty tests obviously are more efficient than their positive counterparts, but they may require that extensive logical operations be conducted after Stage 1 to separate the wells that need to be tested individually in Stage 2 from those which do not. In what follows we let $\hat{p}$ equal p if we are bounding the performance only of positive algorithms and equal 0 if we are bounding the performance of all two-stage algorithms including thrifty ones. Then we may write

$$\Pr(x \in \mathcal{G}2) \;\geq\; \hat{p} \;+\; q \Pr\left(x \in \mathcal{G}2 \;\middle|\; x^-\right)$$

We proceed to underbound the last term in this equation. Order the Stage 1 tests involving the Class 3 well x in some arbitrary manner, and let $T_i(x)$ denote the i-th of them. Then

$$\Pr\left(x \in \mathcal{G}2 \;\middle|\; x^-\right) \;=\; \Pr\left(T_1(x)^+ \;\middle|\; x^-\right) \Pr\left(T_2(x)^+ \;\middle|\; T_1(x)^+, x^-\right) \cdots$$
$$\cdots \Pr\left(T_i(x)^+ \;\middle|\; T_1(x)^+, \ldots, T_{i-1}(x)^+, x^-\right)$$

where the "+ " superscript on a test indicates the event that that test is positive. The key step is to note that

$$\Pr\left(T_i(x)^+ \;\middle|\; T_1(x)^+, \ldots, T_{i-1}(x)^+, x^-\right) \;\geq\; \Pr\left(T_i(x)^+ \;\middle|\; x^-\right)$$

This is because conditioning on other tests being positive when x is known to be negative increases the probability that $T_i(x)$ is positive whenever any wells other than x involved in those tests also are involved in $T_i(x)$ and does not affect this probability if none of the wells in those tests are involved in T_i. Accordingly,

$$\Pr\left(x \in \mathcal{G}2 \;\middle|\; x^-\right) \;\geq\; \prod_i \Pr\left(T_i(x)^+ \;\middle|\; x^-\right) \;=\; \prod_{s=2}^{K_3} (1 - q^{s-1})^{n_s(x)}$$

It follows that

$$\mathsf{E}[T] \;\geq\; K_1 + K_2 + \sum_{x=1}^{K_3} \left[\sum_{s=2}^{K_3} \frac{n_s(x)}{s} + \hat{p} + q \prod_{s=2}^{K_3} (1 - q^{s-1})^{n_s(x)} \right]$$

Note that this bound depends on x only through $\{n_s(x)\}$. Therefore,

$$\mathsf{E}[T] \;\geq\; K_1 + K_2 + K_3 \min_{\{n_s\}} \left[\sum_{s=2}^{K_3} \frac{n_s}{s} + \hat{p} + q \prod_{s=2}^{K_3} (1 - q^{s-1})^{n_s} \right]$$

where the minimum is over all sets of nonnegative integers $\{n_s\} = \{n_2, n_3, \ldots\}$ at least one of which is positive. Introducing the ratios $r_s = n_s/s$, we write:

$$\mathsf{E}[T] \;\geq\; K_1 + K_2 + K_3 \min_{\{r_s\}} \left[\sum_{s=2}^{K_3} r_s + \hat{p} + q \prod_{s=2}^{K_3} \left((1 - q^{s-1})^s\right)^{r_s} \right]$$

The minimum over $\{r_s\}$ may be restricted to appropriate nonnegative rational numbers, but we henceforth relax that restriction. Said relaxation can only

reduce the minimum, so it preserves the direction of inequality. Next we observe that the function $f(s) = (1-q^{s-1})^s$ has a unique minimum as a function of $s \geq 2$ for fixed q. This minimum occurs at a real number s^* that is not necessarily an integer. To find s^* we set $f'(s) = 0$. This yields

$$s^* = 1 + \frac{\log u}{\log q}$$

where u satisfies the transcendental equation

$$-u \log u + (1-u)\log(1-u) = u \log q$$

For small values of $p = 1 - q$, there are two such positive values of u and the minimum we seek corresponds to the smaller of them.

We continue our inequality chain by underbounding $f(s) = (1-q^{s-1})^s$ by $f(s^*)$ in every term in the product over s. This reduces the bound to

$$\mathsf{E}[T] \geq K_1 + K_2 + K_3 \min_{\mathcal{R}} \left\{ \mathcal{R} + \hat{p} + q f(s^*)^{\mathcal{R}} \right\}$$

where $\mathcal{R} = \sum_{s=2}^{K_3} r_s$. Differentiation with respect to $\mathcal{R}$ reveals that the minimum occurs at

$$\mathcal{R}^* = \frac{\log |q \log f(s^*)|}{|\log f(s^*)|}$$

It follows that the efficiency η_2 of any two-stage group test that resolves W wells must satisfy

$$\eta_2 \leq \frac{Wp}{\min_{0 \leq K_3 \leq W} \left\{ W - K_3 + K_3 \left[\mathcal{R}^* + \hat{p} + q(1-q^{s^*-1})^{s^* \mathcal{R}^*} \right] \right\}}$$

The minimum over K_3 is readily evaluated as follows. Define

$$z^* = \mathcal{R}^* + \hat{p} + q(1-q^{s^*-1})^{s^* \mathcal{R}^*}$$

Then the bound reads

$$\eta_2 \leq \max_{0 \leq K_3/W \leq 1} \left\{ \frac{p}{1 - \frac{K_3}{W}(1-z^*)} \right\}$$

It is clear that the maximum occurs at $K_3/W = 1$ if $z^* < 1$ and at $K_3/W = 0$ if $z^* > 1$. It turns out that $z^* < 1$ over the range $0.005 \leq p \leq 0.2$ of principal interest in practice, so the upper bounds in Figure 1 were computed according to the prescription

$$\eta_2 \leq \frac{p}{z^*}$$

designated therein as U_{2P} for positive algorithms ($\hat{p} = p$) and U_2 for thrifty algorithms ($\hat{p} = 0$). In particular, for $p = 0.01$ no positive algorithm can be more efficient than $U_{2P}(p = 0.01) = 9.06\%$. Similarly, even if one knows that $p = 0.01$ and uses a thrifty two-stage algorithm, one cannot achieve an efficiency that exceeds $U_2(p = 0.01) = 9.96\%$.

Appendix C. Lower bounds to efficiency of two-stage testing

Any specific two-stage group test's efficiency is, of course, a lower bound on the most efficient two-stage procedure's efficiency. For example, η_B and η_{3D} of § 2 are two such lower bounds. We are not satisfied with these, however, because they are too far below the upper bounds we derived in Appendix B. Although intuition and analysis provide some insight into the nature of more efficient two-stage group testing procedures, we do not at present have a quantitative description of any two-stage procedure that we think would be quasi-optimum, let alone an exact expression or even an accurate bound for its efficiency. We circumvent this difficulty by using the information-theoretic strategem of random selection. Specifically, we randomly select a family of b Stage-1 group tests over n wells by choosing i.i.d. Bernoulli-π random variables $B_{i,j}$ indexed by $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, b$, and then place well i in test j if and only if $B_{i,j} = 1$.

In § C.1 we calculate the average efficiency achieved when said randomly selected family of tests constitutes Stage 1 of a two-stage positive algorithm. There must exist a two-stage positive group testing procedure whose efficiency is at least as great as this average, so said average serves as a lower bound on the efficiency of two-stage positive algorithms. Its maximum over π and b is plotted versus p for $n = 1000$ as L_{2P} in Figure 1. It is believed that making n larger than 1000 would raise this lower bound only imperceptibly throughout the three-decade range of values of p covered by this plot.

In § C.2 we extend the analysis to cover the resolution of positives. This results in a slightly higher curve L_2, which we are assured is a lower bound on the efficiency achievable by thrifty two-stage group testing.

C.1. Derivation of L_{2P}. The probability that exactly k out the total of n wells are positive is

$$P_k = \binom{n}{k} p^k q^{n-k}$$

When there are exactly k positive wells, any particular one of the $n - k$ negatives fails to get resolved if and only if none of the b tests both contains it as an element and does not contain any of the k positive wells. It follows that the expected number of unresolved negatives, which we shall denote by $\mathsf{E}[N]$, is given by

$$\mathsf{E}[N] = \sum_{k=0}^{n} P_k \, (n-k) \left(1 - \pi \bar{\pi}^k\right)^b$$

where $\bar{\pi} = 1 - \pi$. This expression for $\mathsf{E}[N]$ was first found by Knill [9]. It turns out to be distinctly inferior for computational purposes to an alternative representation we shall now derive. Writing the binomial coefficient in the above formula for P_k in the form

$$\binom{n}{k} = \frac{n(n-1)!}{k!(n-k)(n-1-k)!}$$

and substituting into the expression for $\mathsf{E}[N]$ yields

$$\begin{aligned}\mathsf{E}[N] &= nq\sum_{k=0}^{n-1}\binom{n-1}{k}p^k q^{n-1-k}\left(1-\pi\overline{\pi}^k\right)^b \\ &= nq\sum_{k=0}^{n-1}\binom{n-1}{k}p^k q^{n-1-k}\sum_{l=0}^{b}\binom{b}{l}\left(-\pi\overline{\pi}^k\right)^l \\ &= \sum_{l=0}^{b}(-1)^l\pi^l\binom{b}{l}\sum_{k=0}^{n-1}\binom{n-1}{k}\left(p\overline{\pi}^l\right)^k q^{n-1-k}\end{aligned}$$

and finally

$$\mathsf{E}[N] = nq\sum_{l=0}^{b}(-1)^l\pi^l\binom{b}{l}\left(p\overline{\pi}^l+q\right)^{n-1}$$

This expression for $\mathsf{E}[N]$ has the advantage that its terms alternate in sign, thus speeding convergence. Also there are fewer terms, since the number b of Stage-1 tests is always smaller than the number n of wells, usually significantly so.

The efficiency of a procedure that correctly identifies each positive in a Binomial (n,p) population is the ratio of the expected number of positives, namely np, to the expected number of tests. For each way of filling in the $n \times b$ matrix in order to generate a family T of Stage-1 group tests, the expected total number of tests when a positive procedure is used is $b+np+\mathsf{E}[N|T]$. Here, $\mathsf{E}[N|T]$ denotes the conditional expectation, once a particular family T of Stage-1 pools has been selected, of the number of unresolved negatives that occur in response to which of the n wells are positive and which are negative. The ensemble average efficiency thus is

$$\bar{\eta} = \mathsf{E}_T\left[\frac{np}{b+np+\mathsf{E}[N|T]}\right]$$

where E_T denotes expectation over the random selection of T according to our Binomial (nb,π) scheme. Since $g(x)=c_1/(c_2+x)$ is a convex function of x in the range $x\in[0,+\infty)$ when $c_1\geq 0$ and $c_2\geq 0$, Jensen's inequality allows us to conclude that

$$\bar{\eta} \geq \frac{np}{b+np+\mathsf{E}_T[\mathsf{E}[N|T]]} = \frac{np}{b+np+\mathsf{E}[N]}$$

so

$$\bar{\eta} \geq \frac{p}{\frac{b}{n}+p+q\sum_{l=0}^{b}(-1)^l\pi^l\binom{b}{l}\left(p\overline{\pi}^l+q\right)^{n-1}}$$

Since there must be at least one T in the ensemble whose efficiency when positive group testing is no less than $\bar{\eta}$, the right-hand side of the previous equation is a lower bound to the efficiency attainable with two-stage positive group testing. A computer program was written to maximize this lower bound over b and π for fixed n; the resulting maximum, $L_{2P}(p)$, is plotted vs p in Figure 1 for $n=1000$. In particular, $L_{2P}(0.01)=0.0728$, which results when there are $b=103$ pools and each well is placed in each pool with probability $\pi=0.0778$.

C.2. Derivation of L_2. The middle term in the denominator of the expression for $\bar{\eta}$ at the end of § C.1, namely $+p$, is a consequence of the positivity of the testing procedure. It reflects that fact that every positive well is accorded an individual test in Stage 2. Employing a thrifty procedure would multiply this term by the probability that a positive well is unresolved at the end of Stage 1 and therefore needs to be tested individually in Stage 2. Hence, we can develop a lower bound on the efficiency of thrifty algorithms that is higher than L_{2P} by analyzing the ensemble probability that a positive well fails to get resolved.

Recall that in order for a positive well, call it x, to get resolved, it must be the lone positive in some Stage-1 test all the other members of which are negatives that get resolved. We say that k-conditioning prevails whenever there are k positive wells in all, which happens with probability $P_k = \binom{n}{k} p^k q^{n-k}$. Under k-conditioning, the probability that one of our randomly selected pools contains positive well x but does not contain any of the other $k-1$ positives is $\pi\overline{\pi}^{k-1}$. It follows that the probability that exactly l tests have x as their lone positive well when there are k positive wells in all is

$$P_{l|k} = \binom{b}{l} (\pi\overline{\pi}^{k-1})^l (1 - \pi\overline{\pi}^{k-1})^{b-l}$$

When there are l tests having x as their only positive and there are k positives in all, we shall say that (k,l)-conditioning prevails. Resolving x requires that enough Stage-1 tests are negative that it becomes possible to resolve all the negatives in some test that has x as its lone positive. Under (k,l)-conditioning, the random number M of tests that are negative has a Binomial$(b-l, \overline{\pi}^k)$ distribution. Let C denote the cardinality of the set of wells that belong to one or more of these M negative tests. We have

$$\begin{aligned} \Pr(C = c|k,l) &= \sum_{m=0}^{b-l} \Pr(C{=}c, M{=}m \mid k,l) \\ &= \sum_{m=0}^{b-l} \binom{b-l}{m} \overline{\pi}^{km} (1-\overline{\pi}^k)^{b-l-m} \Pr(C = c|k,l,m) \end{aligned}$$

Under the (k,l,m)-conditioning that appears in the last term, C has a Binomial $(n-k,\ 1-\overline{\pi}^m)$ distribution because each of the $n-k$ negative wells belongs to at least one of the m negative tests with probability $1-\overline{\pi}^m$ independently of all the other negative wells. Note that the parameters of this binomial distribution do not depend on l, in other words,

$$\Pr(C = c|k,l,m) = \Pr(C = c|k,m) = \binom{n-k}{c} (1-\overline{\pi}^m)^c \, \overline{\pi}^{m(n-k-c)}$$

If (k,l)-conditioning prevails and $C = c$, we shall say that (k,l,c)-conditioning prevails. Under (k,l,c)-conditioning, each of the l tests that has x as its lone positive has all its negatives resolved with probability $\overline{\pi}^{n-k-c}$ independently of the other $l-1$ tests. Thus the probability that x is not resolved when (k,l,c)-conditioning prevails is $(1-\overline{\pi}^{n-k-c})^l$. It follows that each positive well remains unresolved with probability $\sum_{k,l,c} P_k P_{l|k} \Pr(C = c|k,l)(1-\overline{\pi}^{n-k-c})^l$. Since the

expected number of positive wells in np, the ensemble average of the random number J of unresolved positives is

$$\mathsf{E}[J] \;=\; np \sum_{k=1}^{n} \sum_{l=0}^{b} \sum_{c=0}^{n-k} P_k P_{l|k} \Pr(C = c|k,l) \left(1 - \overline{\pi}^{n-k-c}\right)^l$$

where explicit expressions for $P_k, P_{l|k}$ and $\Pr(C = c|k,l)$ appear above. (An alternative expression for $\Pr(C = c|k,l)$, that is computationally advantageous in some contexts, is derived in § C.3.) Reasoning as in § C.1, it follows that:

$$\bar{\eta} \;\geq\; \frac{np}{b + \mathsf{E}[J] + \mathsf{E}[N]}$$

for thrifty tests. If we divide the numerator and denominator by n, as we did in § C.1, we see that the $\mathsf{E}[J]$ term indeed provides a term proportional to p, the proportionality constant being the above-derived probability that a randomly selected positive well does not get resolved. Another Matlab program was written to maximize this lower bound over b and π for fixed n; the resulting maximum L_2 is also plotted versus p in Figure 1 for $n = 1000$. In particular, $L_2(0.01) = 0.0779$, which results when there are $b = 105$ pools and each well is placed in each pool with probability $\pi = 0.0781$.

C.3. Alternative expression for $\Pr(C = c|k,l)$. From § C.2 we have:

$$\Pr(C = c|k,l) \;=\; \sum_{m=0}^{b-l} \binom{b-l}{m} \overline{\pi}^{km} \left(1 - \overline{\pi}^k\right)^{b-l-m} \binom{n-k}{c} \left(1 - \overline{\pi}^m\right)^c \overline{\pi}^{m(n-k-c)}$$

We recast this as:

$$\begin{aligned}
&\binom{n-k}{c} \sum_{m=0}^{b-l} \binom{b-l}{m} \overline{\pi}^{m(n-c)} (1 - \overline{\pi}^k)^{b-l-m} (1 - \overline{\pi}^m)^c \;= \\
&\qquad = \;\binom{n-k}{c} \sum_{m=0}^{b-l} \binom{b-l}{m} (\overline{\pi}^{n-c})^m (1 - \overline{\pi}^k)^{b-l-m} \sum_{h=0}^{c} \binom{c}{h} (-\overline{\pi})^{hm} \\
&\qquad = \;\binom{n-k}{c} \sum_{h=0}^{c} \binom{c}{h} \sum_{m=0}^{b-l} \left((-1)^h \overline{\pi}^{h+n-c} \right)^m (1 - \overline{\pi}^k)^{b-l-m}
\end{aligned}$$

Using the binomial theorem reduces this to:

$$\Pr(C = c|k,l) \;=\; \binom{n-k}{c} \sum_{h=0}^{c} \binom{c}{h} \left((-1)^h \overline{\pi}^{h+n-c} + 1 - \overline{\pi}^k \right)^{b-l}$$

This is the desired alternative expression for $\Pr(C = c|k,l)$. When $\overline{\pi}$ is close to 1, as it usually happens to be in cases of practical interest, the terms in this sum will alternate in sign. Also, the power $b - l$ to which the bracket gets raised does not vary with the summation index. However, the number of terms in this alternative expression usually is larger than that in the original expression, since most of the values of c that need to be considered are larger than b.

Acknowledgment. We are indebted to Catherine Manlove, Xiaohai Zhang, Raul Casas, Srikant Jayaraman and Anthony Macula. Catherine performed the calculations for a preliminary version of this paper. Raul wrote Matlab programs to evaluate the lower bounds L_{2P} and L_P of Appendix C; Xiaohai programmed the other formulas. Incisive comments from Xiaohai and Srikant concerning the analysis of thrifty algorithms are deeply appreciated. Anthony alerted us to the work of Knill [**9**] and kindly provided a preprint of his own paper [**13**].

References

[1] T. BERGER, N. MEHRAVARI, D. TOWSLEY, AND J.K. WOLF, Random access communication and group testing, *IEEE Trans. Commun.*, vol. 32, p. 769, 1984.

[2] P.J. CAMERON, AND J.H. VAN LINT, *Graphs, Designs and Codes*, London Mathematical Society Lecture Note Series, #43, Cambridge University Press, Cambridge, 1980.

[3] C.L. CHEN, P. STROP, M. LEBL, AND K.S. LAM, The bead-one compound combinatorial peptide library; different types of screening, *Methods Enzymol.*, vol. 267, p. 211–9, 1996.

[4] C.L. CHEN AND W.H. SWALLOW, Using group testing to estimate a proportion, and to test the binomial model, *Biometrics*, vol. 46, pp. 1035–1046, 1990.

[5] X. CHENG, B.K. KAY, AND R.L. JULIANO, Identification of a biologically significant DNA-binding peptide motif by use of random phage display library, *Gene*, vol.171, p.1-8, 1996.

[6] H.A. COLLER AND B.S. COLLER, Poisson statistical analysis of repetitive subcloning by the limiting dilution technique as a way of assessing hybridoma monoclonality, *Methods Enzymol.*, vol. 121, p. 412, 1986.

[7] R. DORFMAN, Detection of defective members of large populations, *Ann. Math. Statist.*, vol. 14, p. 436, 1943.

[8] R.A. HOUGHTEN, S.E. BLONDELLE, C.T. DOOLEY, B. DORNER, J. EICHLER, AND J.M. OSTRESH, Libraries from libraries: generation and comparison of screening profiles, *Molecular Diversity*, vol. 2, pp. 41–55, 1996.

[9] E. KNILL, Lower bounds for identifying subset members with subset queries, in *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, Ch. 40, pp. 369–377, San Francisco, CA, January 1995.

[10] J.A. KOZIOL, Evaluation of monoclonality of cell lines from sequential dilution assays, Part II, *J. Immunol. Methods*, vol. 107, p. 151, 1988.

[11] J.A. KOZIOL, C. FERRARI, AND F.V. CHISARI, Evaluation of monoclonality of cell lines from sequential dilution assays, *J. Immunol. Methods*, vol. 105, p. 139, 1987.

[12] R. LIETZKE AND K. UNSICKER, A statistical approach to determine monoclonality after limiting cell plating of a hybridoma clone, *J. Immunol. Methods*, vol. 76, p. 223, 1985.

[13] A.J. MACULA, Probabilistic nonadaptive and two-stage group testing with relatively small pools, *International Conference on Combinatorics, Information Theory and Statistics*, University of Southern Maine, Portland, ME, July 1997.

[14] M. SOBEL AND P.A. GROLL, Group testing to eliminate efficiently all defectives in a binomial sample, *Bell Syst. Tech. J.*, vol. 38, p. 1179, 1959.

[15] R. STASZEWSKI, Cloning by limiting dilution: an improved estimate that an interesting culture is monoclonal, *Yale J. Bio. and Med.*, vol. 57, p. 865, 1984.

[16] J. SUMMERTON, Sequence-specific crosslinking agents for nucleic acids: design and functional group testing, *J. Theoretical Biology*, vol. 78, pp. 61–75, 1979.

Computational Imaging

David C. Munson, Jr.

COORDINATED SCIENCE LABORATORY AND
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, IL 61801, USA
d-munson@uiuc.edu

1. Introduction

Advances in the theory of imaging systems and digital image processing, coupled with increasingly fast digital computers, have spawned a new field called computational, or digital, imaging. In this chapter we give an overview of this field and illustrate its application in microwave remote surveillance. The term computational imaging refers to systems that collect signals from the environment and then use a digital computer to compute an image from the data. Such systems may be either active or passive. Active systems emit energy and then measure the reflection from or transmission through an object or scene, whereas passive systems collect energy already extant in the environment. Typically, the data are collected from fixed spatial locations using many sensors, or from fewer sensors that are in motion. Often, the goal is to produce an image having resolution that is far higher than that dictated by the physical receiving aperture collecting the data. Table 1 lists several common types of computational imaging systems. Applications range from medical imaging to characterization of microscopic materials properties to radio astronomy.

X-ray computer tomography (CT)
Magnetic resonance imaging (MRI)
Ultrasound imaging
X-ray crystallography
Electron microscopy
Geophysical imaging
Beamforming sonar
Aperture-synthesis radio astronomy
Range-Doppler radar
Synthetic aperture radar (SAR)

Table 1. *Examples of computational imaging systems*

Imaging data can be collected from a wide range of the electromagnetic spectrum, from radio frequencies all the way through the infrared, optical, and X-ray

bands. Examples of such systems in Table 1 include range-Doppler radar [**2**], synthetic aperture radar [**2, 14**], radio astronomy [**6, 53**], X-ray tomography [**31, 37**], and X-ray crystallography [**29, 35**]. Likewise, acoustic phenomena also support computational imaging in ultrasound fetal monitors [**45**], sonar, and geophysical imaging [**12, 19, 44**].

In this chapter, we will begin by providing an overview of many of the imaging systems in Table 1. In doing so, we will highlight some of the connections between these systems. The concepts of projection (line integrals) and Fourier transform will play a prominent role. In the next section, we will then show how knowledge of more than one imaging system, together with a signal processing viewpoint, can lead to improved imaging systems. As a case in point, we will consider the problem of radar astronomy, where ideas from range-Doppler imaging, tomography, and synthetic aperture radar have been combined to produce an imaging algorithm that offers improved fidelity [**56**]. In the last section, we will close with some comments on the drive toward unification in the field of computational imaging and the role that is being played by Richard E. Blahut.

2. An overview of computational imaging

At first glance, the systems in Table 1 may appear to be unrelated. However, from a mathematical modeling point of view, they are tightly coupled. Many of the relationships occur through the concepts of projection (line integrals) and Fourier transforms in one and more dimensions. Most of the overview presented in this section was taken from [**9**].

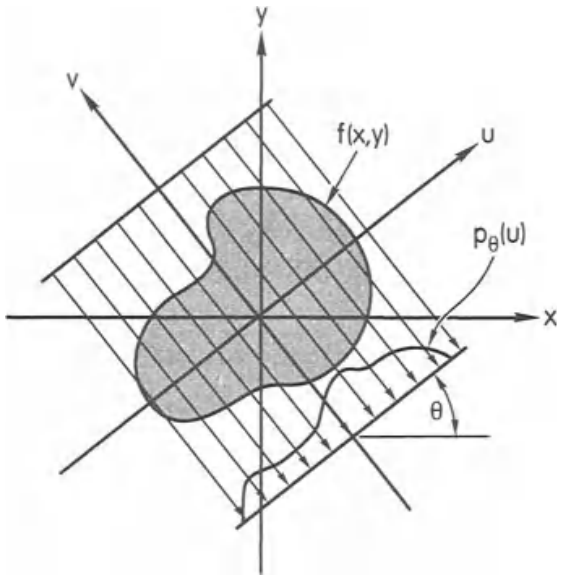


Figure 1. *Collection of a projection in tomographic imaging*

In its most basic form, a tomographic imaging system collects measurements of 1-D line integrals along parallel paths through a 2-D object, as illustrated in Figure 1. Here, let the 2-D density or reflectivity be denoted $f(x, y)$. Then the

set of line integrals collected at a given orientation θ is a 1-D function called a projection, given by:

$$p_\theta(u) = \int_{-\infty}^{\infty} f(u\cos\theta - v\sin\theta, u\sin\theta + v\cos\theta)\, dv$$

By collecting projections at different angles with respect to the fixed (x, y) coordinate system, we build up a two-dimensional function from which $f(x, y)$ is reconstructed. This function is referred to as the Radon transform [**43**]. The Projection-Slice Theorem (PST) [**3, 18**] is the fundamental result underlying the reconstruction of images from their projections. This theorem states that the Fourier transform of a 1-D projection at angle θ is equal to the 2-D Fourier transform of the image evaluated along the line passing through the origin at the same angle θ in Fourier space. Specifically, if $P_\theta(U)$ is the 1-D Fourier transform of a projection, and F is the 2-D Fourier transform of f, then

$$P_\theta(U) = F(U\cos\theta, U\sin\theta)$$

Using this result it is straightforward to compute a reconstruction of $f(x, y)$ by first Fourier transforming each projection to produce samples of the 2-D image Fourier transform, lying on the polar grid in Figure 2, where the Fourier transform of a single projection gives samples of $F(\cdot, \cdot)$ on a single radial trace in Figure 2, oriented at the same angle as the projection. These polar data then can be interpolated onto a regular Cartesian sampling grid and the image can be produced using the inverse 2-D discrete Fourier transform [**38, 51**]. This form of image reconstruction is called a direct Fourier method.

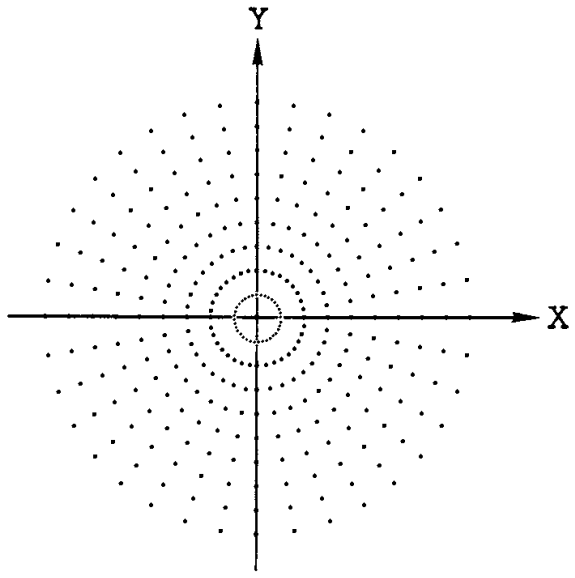


Figure 2. *Polar Fourier-domain sampling grid in parallel-beam tomography*

An alternative approach is based on analytic manipulations, showing that the image can be recovered directly from the projection data without using Fourier transforms. Each projection is filtered with a ramp filter and then back-

projected along the path over which the line integration was originally performed. Carrying out this procedure for each projection and then summing the results produces the reconstructed image [**31, 46**]. This method, typically referred to as filtered back-projection (FBP), is the basis for reconstruction in almost all commercially available computer tomography systems. In recent years it has been discovered that FPB is a fast means of implementing the direct Fourier method for the case where the polar-to-Cartesian interpolator uses a 2-D periodic-sinc kernel [**10, 48**].

Modern CT scanners collect X-ray projection data far more quickly using a fan-beam geometry where energy from a single X-ray source fans out through the object to be imaged. It is possible to sort fan-beam data into equivalent parallel projections before reconstruction. Fortuitously, however, a simple change of variable in the Radon inversion formula allows reconstruction by directly weighting, filtering, and back-projecting the fan-beam data [**25, 26, 31**].

More recently, there has been a great deal of interest in reconstruction of images from photon-limited data that can be modeled as a collection of Poisson random variables [**55**]. These data arise in nuclear medicine systems in which a patient is injected with a radio-labeled pharmaceutical. These systems detect photons produced either by gamma ray emissions, as in single photon emission computed tomography (SPECT), or by positron-electron annihilation, as in positron emission tomography (PET). Shepp and Vardi [**47**] proposed a maximum-likelihood solution for this Poisson imaging problem based on the EM algorithm. Due to the large number of unknowns (pixels) to be estimated from a finite data set, the maximum-likelihood images tend to exhibit high variance. Consequently a large number of researchers have studied methods for stabilizing the solutions using penalized-maximum-likelihood or Bayesian estimation techniques with Markov random field priors [**20, 21, 39**]. The Bayesian approach allows inclusion of prior information either in the form of a general smoothness constraint, or more specific structural information that may have been extracted from a co-registered image obtained using a different modality [**22**].

There is now increasing interest in fully 3-D CT systems in which data are collected either as cone-beam projections (X-ray CT and SPECT) or as 2-D projections of 3-D objects along large numbers of oblique angles (PET). Analytic approaches for reconstruction of images from these data remain a rich area for research [**15, 32, 42**]. Similarly, these data present significant challenges for iterative reconstruction methods [**30**]. In the next section of this chapter, where we present a new model for imaging in radar astronomy, we will have need of two different 3-D versions of the projection-slice theorem. These mathematical principles will be introduced at that time.

Magnetic resonance (MR) imaging differs from X-ray and emission CT, in the sense that the image Fourier transform or k-space is measured directly. This is achieved by using a magnetic field gradient to produce a spatial frequency encoding of the magnetic resonance signal from hydrogen nuclei in the body. Using combinations of time-varying magnetic field gradients and radio-frequency

pulses, it is possible to obtain k-space (Fourier) data lying on a wide range of sampling patterns. Early research in MR sampled Fourier space on a polar grid and reconstructed images using methods similar to those employed in CT [**36**]. More commonly, MR data are collected directly on rectangular sample patterns [**34**], although fast acquisition can be achieved using more complex patterns such as the spiral scan [**1**].

We next consider computational imaging outside of the medical realm, in X-ray crystallography, electron microscopy, seismic imaging, radio astronomy, and synthetic aperture radar. The Fourier transform ties together the imaging theory in these diverse areas. We will only briefly comment on the first three of these types of imaging and then elaborate somewhat on the latter two. In X-ray crystallography the collected data represent samples of the magnitude of the 3-D Fourier transform of the electron density. Given limited Fourier magnitude information (phase is missing), inversion procedures incorporate positivity and a priori information on crystalline structure to successfully form an image [**29, 35**]. Transmission electron microscopy is a tomographic system where an electron beam is passed through a specimen [**13**]. The specimen is rotated on a stage to collect line integrals at different angles. The angle cannot be too steep; otherwise the specimen is too thick in the beam direction. Hence, there is a missing cone of data in the Fourier domain. The so-called missing cone problem has received widespread study in both CT and electron microscopy [**16, 52**]. Seismic imaging techniques exist that use ideas from beamforming [**44**]. Tomographic imaging is also employed in seismic exploration [**17, 19**].

In radio astronomy, the objective is to make a map of the brightness, in an RF band, of a portion of the sky. In the 1950s, Ronald Bracewell pioneered projection-based radio-astronomical imaging [**6, 25**], which predated the very similar methods that were later independently developed for medical X-ray CT. In the radio-astronomical version of tomographic imaging, the projections are acquired by observing an astronomical target during occlusion by another object. For example, it is possible to image the sun by collecting measurements taken during occlusion by the moon. Suppose that the total energy in the image is measured as a function of time. Then the difference between two successive time measurements will be proportional to the integrated energy in the narrow strip on the sun (a line integral!) that is either covered or uncovered during the time period between the two measurements. In practice, it is sometimes possible to collect enough of these measurements to produce an image of very useful resolution. A second form of radio astronomy, which employs interferometry, is more common. An interferometric radio telescope points an array of antennas at a fixed region in the sky. The individual antennas may be located in the same geographic region, or they may be widely spread out – even on different continents. The Fourier-domain imaging principle employed is based on the van Cittert-Zernike theorem, which essentially states that when imaging a spatially uncorrelated source distribution, the correlation of any two antenna outputs provides a sample of the 2-D Fourier transform of the brightness of the sky in the region being imaged [**53**]. By tracking the same region of the sky for

many hours, data is collected from different vantage angles, providing Fourier data lying on an ellipse. Similarly, by performing pairwise correlations for all antennas in the array, Fourier data is obtained lying on many ellipses. The locations of the data ellipses in the Fourier plane depend on the geometry of the array. Typically the array geometry is selected so that the data ellipses will provide moderately uniform coverage of the Fourier plane. The simplest form of image reconstruction interpolates the available Fourier data to a Cartesian grid and then applies an inverse 2-D discrete Fourier transform. In practice, more sophisticated imaging algorithms, such as maximum entropy, are employed.

Synthetic aperture radar (SAR) is used to produce high-resolution microwave imagery in all-weather conditions for both surveillance and scientific purposes. The name for this type of radar stems from the fact that instead of using a large antenna array, the SAR concept is to fly a small antenna to the positions that would be occupied by the individual elements in the large array, and then to use signal processing to form a high-resolution image. The oldest form of SAR is known as strip-mapping [**2**] where the antenna remains fixed with respect to the radar platform and thereby sweeps out a strip of terrain as the platform moves. The customary image formation algorithm is correlation based. A newer and more accurate algorithm, called the omega-k or wavenumber approach, produces Fourier-domain data as an intermediate step [**7, 50**].

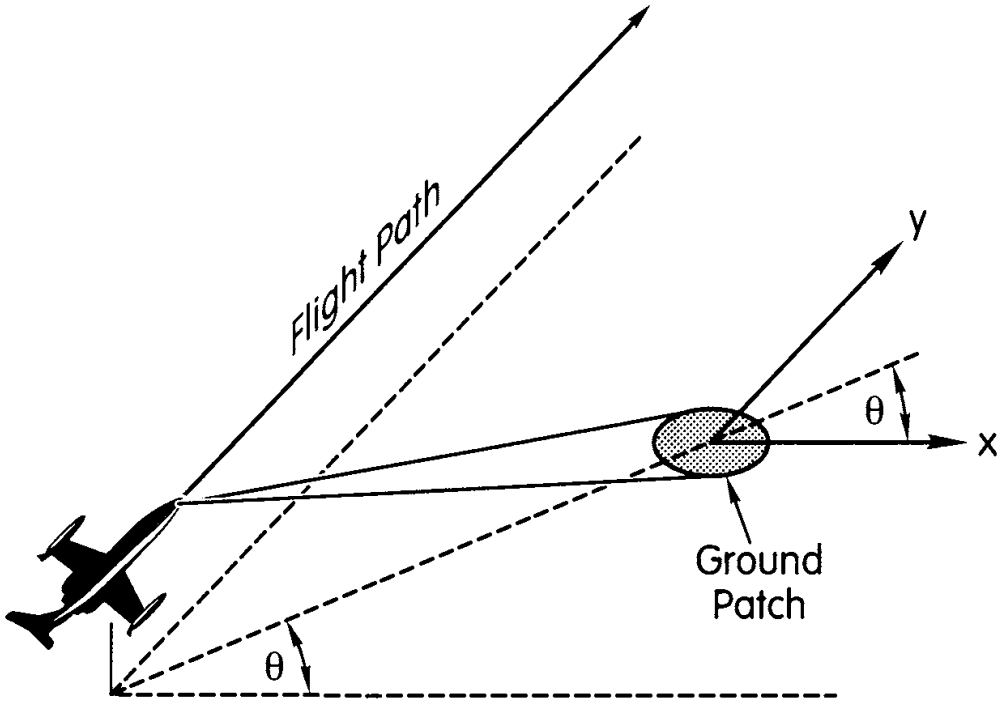


Figure 3. *Spotlight-mode SAR data collection geometry*

Jack Walker and his colleagues at ERIM invented a second type of SAR, called spotlight-mode [**54**]. In spotlight-mode SAR, illustrated in Figure 3, the radar antenna is steered to illuminate a scene from many different vantage angles.

This type of SAR was originally described in range-Doppler terms, but it was later discovered that spotlight-mode SAR can be conveniently explained in terms of the projection-slice theorem from CT [**40**]. Assuming a stepped-frequency or linear FM transmitted waveform, the signal from the quadrature demodulator in spotlight-mode SAR directly provides samples of the 1-D Fourier transform of a projection of the reflectivity, where the projection direction is orthogonal to the radar line of sight. It follows from the PST that these Fourier data represent samples of the 2-D transform F on a radial trace in 2-D Fourier space. Transmitting waveforms (collecting Fourier transform projection data) from uniformly spaced angles along the flight path provides samples of the Fourier transform of the 2-D scene reflectivity on a small section (unlike CT) of a polar grid, offset from the origin, as shown in Figure 4.

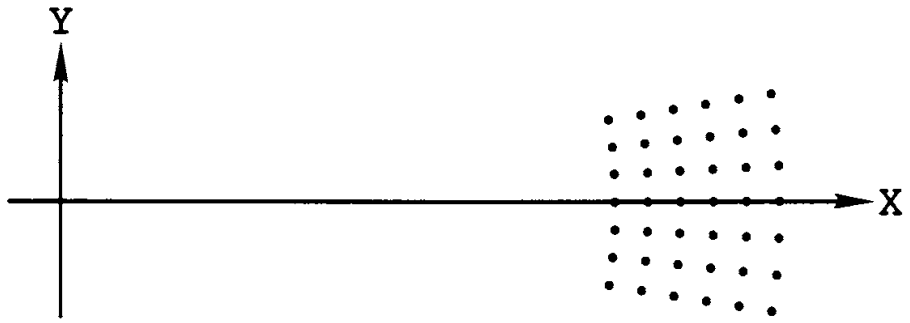


Figure 4. *Fourier data grid in spotlight-mode SAR*

The angular coverage in the Fourier domain is the same as the range of angles through which the radar views the target. The inner and outer radii of the polar sector are proportional to the lowest and highest frequencies in the transmitted waveform. The polar sector where data is available is generally small and is therefore nearly Cartesian. Thus simple interpolation suffices prior to computing an inverse discrete Fourier transform for image formation [**10**]. Although the low-frequency Fourier components are missing in the data, it is still possible to form a high-resolution image because the radar reflectivity is complex, often with fairly random phase [**41**].

Spotlight-mode SAR can also be used in an inverse mode, where the radar is stationary and the target moves. This is useful in imaging ships on the ocean surface [**57**] and also serves as an excellent processing model for radar astronomy [**56**], as we shall see in the next section. Some excellent references on SAR include [**14**] for strip mapping and [**8, 28**] for spotlight-mode.

3. Planetary radar astronomy

In the early 1960's, Paul Green [**23**] presented a complete theory for using radar to image planets and other rotating heavenly bodies. A range-Doppler imaging principle was developed, where points on the planet's surface are mapped

according to their iso-range contours (concentric rings on the planet's surface) and iso-Doppler contours (lines parallel to the axis of rotation). The processing involves a bank of matched filters that are matched to frequency-shifted (that is, Doppler-shifted) versions of the transmitted signal. Other researchers [**24**] have made this the standard processing model for radar astronomy and have imaged numerous astronomical objects, including Venus and the Moon.

The range-Doppler processing model assumes that points on the rotating planet surface do not move through resolution cells during the period of microwave illumination. While this assumption is reasonable for slowly rotating bodies, such as Venus, it is a poor assumption for imaging other objects such as asteroids or the Moon. In this section, we consider an alternative processing model based on inverse synthetic aperture radar (ISAR), which removes the restriction on motion through resolution cells. The section begins with a review of range-Doppler planetary imaging. We next discuss 3-D spotlight-mode SAR and ISAR from a tomographic perspective and show how it applies to the planetary imaging scenario. We then show a lunar image produced using this alternative processing model. § 3.1 and § 3.2 draw heavily from [**11**].

3.1. Review of range-Doppler planetary imaging. Since the surface of a planet is composed of point reflectors located at varying ranges from the radar, different portions of the surface will return energy at different delays. The loci of all points about the transmitter-receiver location having constant delay τ are concentric spheres of radius $c\tau/2$. The intersections of these spheres with the planetary surface are circles as shown in Figure 5, which are the loci of constant range as the target is viewed head-on from the radar.

The rotation of the planet will cause the various point reflectors on the planetary surface to have different line-of-sight velocities v_ℓ with respect to the radar. Therefore, a Doppler shift of $\nu = (2f_0/c)\, v_l$ will be imposed on the incident signal by each reflector, and the amount of shift will depend on the location of the reflector on the planet's surface. Here, f_0 is the transmitter frequency and c is the speed of propagation. Using the coordinate system of Figure 5, with the origin at the center of the planet, loci of points having constant Doppler shift are planes parallel to the (u, ω) plane. Along any curve of intersection between such a plane and the planetary surface, the Doppler shift is constant.

In Figure 5, contours of equal range intersect contours of equal Doppler shift at at most two points A and B as shown. To isolate and study such a pair of point reflectors, it simply remains to isolate that part of the returned signal having a given time delay τ and simultaneously a given frequency shift ν. This can be done using a bank of matched filters, with each filter tuned to a different frequency offset of the transmitted waveform. The outputs of this set of filters forms a 2-D map in range and Doppler coordinates. In practice, if the transmitted signal is a pulse train, then this same processing can be accomplished more easily using a single matched filter followed by a series of 1-D FFTs [**4**]. For unambiguous mapping it is necessary to discriminate between the two points A and B in Figure 5. This can be accomplished by illuminating the northern

and southern hemispheres separately, or by using a bi-static radar with a second receiving antenna located separately from the main transmitter/receiver.

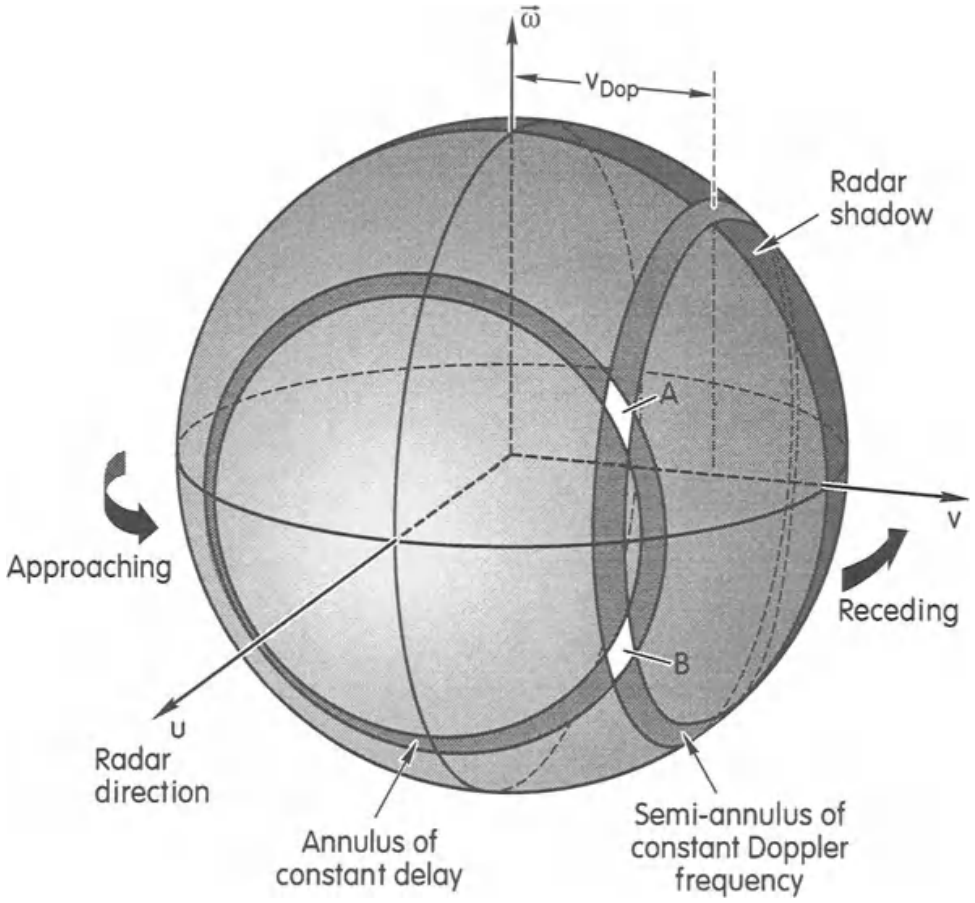


Figure 5. *Iso-range and iso-Doppler contours in planetary radar astronomy*

The operation of a bank of matched filters is conveniently characterized by introducing the notion of ambiguity function. To develop the necessary concepts, consider a target whose reflectivity density function in delay-Doppler coordinates is an impulse $\delta(\tau, \nu)$, located at the origin of the illuminated scene. The form of receiver that will develop maximum signal-to-noise ratio from such a target in additive, white noise is the matched filter. Letting $s(t)$ be the transmitted complex baseband signal and ignoring the round-trip delay, the returned echo from a point reflector at the origin is simply $q(t) = s(t)$ and the matched filter output is given by:

$$y(t) = \int_{-\infty}^{\infty} q(\beta) s^*(\beta - t)\, d\beta \tag{1}$$

The matched filter, or correlation detection operation, described above is optimum only if the reflectivity density function is an impulse located at the origin. Thus, if the impulse-like reflector is not at the origin of the scene, but instead at the point (τ_0, ν_0) in delay-Doppler coordinates, then $q(t) = s(t-\tau_0)e^{j2\pi\nu_0 t}$ and the output given by (1) is diminished. As a result, the filter impulse re-

sponse must be readjusted to the frequency offset ν_0, and its output observed τ_0 seconds later to restore the optimal matched filter condition.

Suppose the input waveform $q(t)$ in (1) is $s(t+\tau)e^{-j2\pi\nu t}$. Then the output at $t = 0$ of this matched filter, which has been set for a point reflector at $\tau = 0$ and $\nu = 0$, is given by:

$$\begin{aligned} y(0) &= \int_{-\infty}^{\infty} s(\beta+\tau)e^{-j2\pi\nu\beta}s^*(\beta)\,d\beta \\ &= e^{j2\pi\tau\nu}\int_{-\infty}^{\infty} s(\beta)s^*(\beta-\tau)e^{-j2\pi\nu\beta}\,d\beta \end{aligned}$$

If this matched-filter output is plotted as a function of τ and ν, the result is the quantity $\chi(\tau,\nu)$, called the ambiguity function. A slightly different form, that is asymmetric but essentially equivalent, is defined as follows:

$$\chi(\tau,\nu) = \int_{-\infty}^{\infty} s(\beta)s^*(\beta-\tau)e^{-j2\pi\nu\beta}d\beta \tag{2}$$

This second equivalent form of the ambiguity function will be used throughout this chapter.

A range-Doppler radar can be modeled as a device that forms a two-dimensional convolution of the reflectivity density function of an illuminated target with the ambiguity function of the radar waveform [4]. This unifying description, which will be reviewed here, is extremely powerful because it models the radar processor as a simple two-dimensional filter whose point spread function is the radar ambiguity function. Hence, the quality and resolution of the reconstructed target image are related directly to the shape of the radar ambiguity function.

For a rotating planet, the return from a point reflector with reflectivity ρ on the planetary surface is given by:

$$q(t) = \rho \cdot s\left(t - \frac{2R(t)}{c}\right)$$

where $R(t)$ is the time-dependent range of the point reflector, given by:

$$R(t) = R_0 + \dot{R}_0 t + \ddot{R}_0 \frac{t^2}{2} + \cdots$$

Usually $R(t)$ can be adequately approximated by the first several terms. If the origin of the imaging coordinate system is located at the radar, then in the simplest case of constant velocity, $R(t)$ is given by:

$$R(t) = R_0 - v_l t$$

where R_0 is the distance from the radar to the point reflector at $t = 0$ and v_l is the velocity of the reflector in the direction of the radar. The received complex baseband signal from a single point reflector is then

$$q(t) = \rho \cdot s\left(\left(1 + \frac{2v_l}{c}\right)t - \frac{2R_0}{c}\right)$$

The velocity v_l is very small compared to c, so that the dilation term $\left(1 + \frac{2v_l}{c}\right)$ in the above equation can be ignored. Thus rotation of the planet has negligible

effect on the properties of the returned echo when the incident signal is baseband. However, a high-frequency passband signal is changed quite significantly by the planet's motion; its frequency is changed. That is, after a single range delay, the transmitted complex passband signal $s(t)e^{j2\pi f_0 t}$ is received as

$$q(t) \;=\; \rho \cdot s\left(Bigl(1+\frac{2\nu_l}{c}\Big)t \;-\; 2R_0/c\right) e^{j2\pi f_0\left(1+\frac{2\nu_l}{c}\right)t} e^{-j2\pi f_0 \frac{2R_0}{c}}$$

Again, the dilation term in the baseband signal $s(t)$ is ignored, and the received signal can be expressed compactly as the complex passband signal

$$q(t) \;=\; \rho \cdot s(t-\tau_0)e^{j2\pi\nu_0 t}e^{-j2\pi\gamma}e^{j2\pi f_0 t}$$

or the complex baseband signal

$$q(t) \;=\; \rho \cdot s(t-\tau_0)e^{j2\pi\nu_0 t}e^{-j2\pi\gamma}$$

where $\tau_0 = 2R_0/c$ is the time delay at $t = 0$, $\nu_0 = (2f_0/c)v_\ell$ is a Doppler shift, and $\gamma = f_0\tau_0$ is a phase offset.

More generally, if $s(t)e^{j2\pi f_0 t}$ is the transmitted signal, then the complex baseband representation of the returned echo having delay τ and Doppler shift ν is

$$q(t) \;=\; \rho(\tau,\nu) \cdot s(t-\tau)e^{j2\pi\nu t}e^{-j2\pi\gamma'}$$

where $\gamma' = f_0\tau$ is a phase shift due to the time of propagation. It is important to note that here $\rho(\tau,\nu)$ is the sum of reflectivity densities of all point reflectors at delay τ and Doppler shift ν. For example, as Figure 5 indicates, $\rho(\tau,\nu)$ for a spherical target will have contributions from two points on the target surface for each possible (τ,ν). For a planetary surface having a complex reflectivity density of $\rho(\tau,\nu)$ in delay-Doppler coordinates, the returned complex baseband signal is given by superposition as:

$$q(t) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi\gamma'}\rho(\tau',\nu')s(t-\tau')e^{j2\pi\nu' t}d\tau' d\nu'$$

If the signal $q(t)$, given by the above equation, is the input to a filter matched to $q_0(t) = s(t-\tau)e^{j2\pi\nu t}$, then by (1) the output at $t = 0$ will be $y(0)$ given by:

$$\begin{aligned} M(\tau,\nu) \;&=\; \int_{-\infty}^{\infty} q(\beta)\left(s(\beta-\tau)e^{j2\pi\nu\beta}\right)^* d\beta \\ &=\; \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi\gamma'}\rho(\tau',\nu')s(\beta-\tau')e^{j2\pi\nu'\beta}d\tau' d\nu'\right] \\ &\qquad \cdot s^*(\beta-\tau)e^{-j2\pi\nu\beta}\, d\beta \\ &=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi\gamma'}\rho(\tau',\nu')\left[\int_{-\infty}^{\infty} s(\beta')s^*(\beta'-(\tau-\tau'))\right. \\ &\qquad \left.\cdot e^{-j2\pi(\nu-\nu')\beta'}d\beta'\right] e^{-j2\pi(\nu-\nu')\tau'}\, d\tau' d\nu' \end{aligned} \tag{3}$$

Notice that the parenthesized term is $X(\tau-\tau',\nu-v')$, from (2). In most applications $(\nu-\nu')\tau'$ is either much less than one or fairly constant over the range of (τ',ν') for which $\chi(\tau-\tau',\nu-\nu')$ has significant magnitude. In either

case, $e^{-j2\pi(\nu-\nu')\tau}$ can be treated as approximately constant and can therefore be ignored in (3). Consequently,

$$M(\tau,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi\gamma'}\rho(\tau',\nu')\chi(\tau-\tau',\nu-\nu')\,d\tau'd\nu' \qquad (4)$$

Hence, the output of the matched filter as a function of time delay and frequency offset is essentially the two-dimensional convolution of the reflectivity density function with the radar ambiguity function. Therefore, the resolution achieved with range-Doppler imaging is entirely dependent upon the shape of $\chi(\tau,\nu)$. In fact, if $\chi(\tau,\nu)=\delta(\tau,\nu)$, then $\rho(\tau,\nu)$ could be recovered exactly. On the other hand, $\chi(\tau,\nu)=\delta(\tau,\nu)$ is impossible, and range-Doppler imaging systems can only strive for impulse-like ambiguity functions [**33, 58**].

Since $\rho(\tau,\nu)$ is a function of time for a rotating target, the main approximations needed for the validity of (4) and the attendant matched–filter processing are that $\ddot{R}_0$ and other higher-order range terms can be neglected in the range function and that $\rho(\tau,\nu)$ remains essentially constant during target illumination. In practice, these assumptions are sometimes violated in radar astronomy, leading to images that are blurred. It turns out, however, that these assumptions are completely unnecessary! As we shall see, a tomographic ISAR model suggests a different image formation algorithm and completely solves the blurring problem.

3.2. 3-D spotlight-mode SAR processing model. Figure 3 earlier depicted the imaging geometry for spotlight-mode SAR where for a linear FM transmitted waveform, the data collected from each viewpoint, after quadrature demodulation, are a slice along the viewing angle of the 2-D Fourier transform of the scene. The tomographic formulation of SAR given in [**40**] considered the imaging of only planar scenes. For the radar astronomy problem, we must consider the scenario in which the radar scene is 3-D. In this section we explain, using tomographic principles, that each demodulated radar return signal from a spotlight-mode collection geometry represents a radial set of samples of the 3-D Fourier transform of the target reflectivity density function. Although 3-D SAR imaging was first studied in [**54**], the viewpoint that follows was originated by Jakowatz and Thompson [**27**].

To develop the 3-D tomographic formulation of SAR, it is first necessary to extend the projection-slice theorem from two to three dimensions. This extension produces two alternative forms, depending upon how the third dimension is used. Each version is relevant to the ensuing discussion and is described below.

Let $f(x,y,z)$ denote the 3-D complex-valued reflectivity density in spatial coordinates. Then define the 3-D Fourier transform of $f(\cdot)$ as

$$F(X,Y,Z) = \iiint f(x,y,z)e^{-j(xX+yY+zZ)}\,dxdydz$$

The two versions of the projection-slice theorem, stated below, relate certain projections of $f(\cdot)$ to corresponding slices of its Fourier transform $F(\cdot)$.

Theorem 1. *The 1-D Fourier transform of the function*

$$p_2(x) = \iint f(x,y,z)\, dydz$$

is equal to a 1-D ray of $F(X,Y,Z)$ obtained by evaluating $F(X,Y,Z)$ on the X-axis, namely:

$$P_2(X) = \int p_2(x) e^{-jxX}\, dx = F(X,0,0)$$

We refer to Theorem 1 as the linear-ray theorem, for obvious reasons. Invoking the rotational property of Fourier transforms, this theorem may be generalized to an arbitrary orientation in three-dimensional space, as shown in Figure 6.

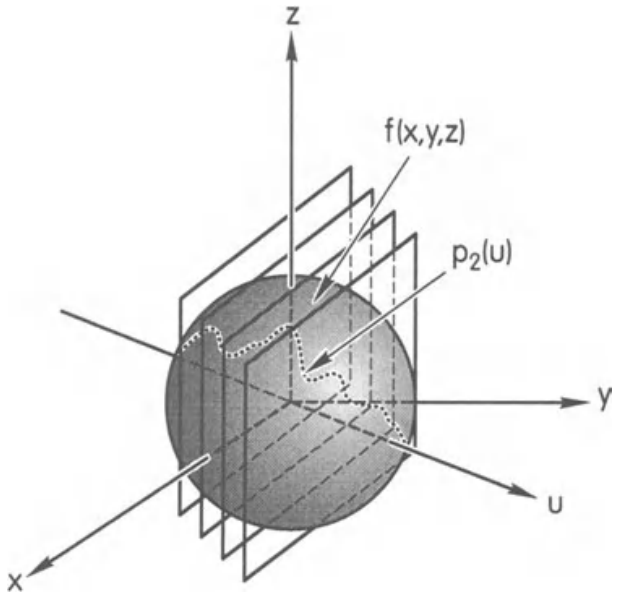


Figure 6. *Illustration of the linear-ray theorem*

Taking planar integrals orthogonal to the u-axis gives projection $p_2(u)$. The 1-D Fourier transform of $p_2(u)$ is a 1-D ray of $F(X,Y,Z)$, where the orientation of the ray in the Fourier domain is the same as the orientation of the u-axis in the spatial domain.

Theorem 2. *The 2-D Fourier transform of the function*

$$p_1(x,y) = \int f(x,y,z)\, dz \tag{5}$$

is equal to a 2-D slice of $F(X,Y,Z)$ taken through the (X,Y) plane, namely:

$$P_1(X,Y) = \iint p_1(x,y) e^{-j(xX+yY)}\, dxdy = F(X,Y,0) \tag{6}$$

Since (6) relates planar projection functions to planar Fourier transform sections, this theorem will be referred to as the planar-slice theorem. This theorem,

which can be generalized using the rotational property of Fourier transforms, shows that inverse Fourier transforming a 2-D sheet P_1 of 3-D Fourier data F will produce p_1, which is a collapsed version of $f(x, y, z)$. In particular, p_1 is a 1-D integral of f, where the direction of integration is orthogonal to the orientation of the plane of Fourier data P_1.

For 3-D SAR imaging, we will consider $f(x, y, z)$ to be equal to zero everywhere except on the outer surface of the planet. This assumption is reasonable, since for high carrier frequencies in the gigahertz range, the wavelength of the transmitted waveform is too small for the radar energy to substantially penetrate the planet surface.

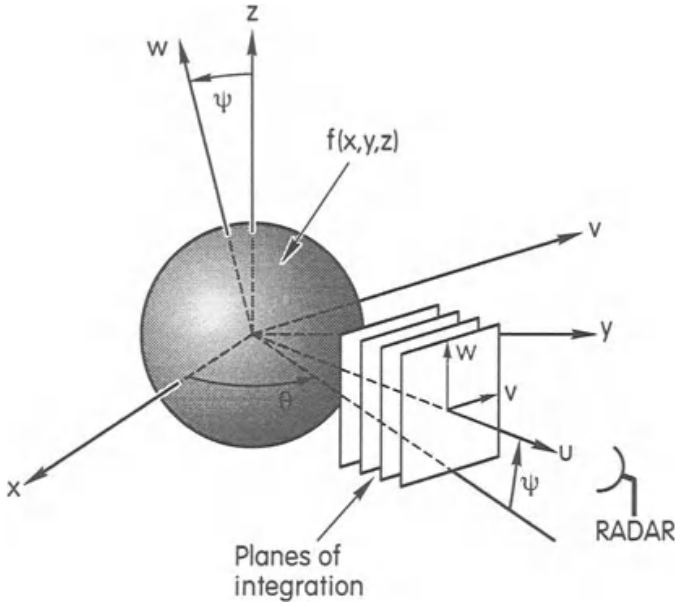


Figure 7. *Imaging geometry for three-dimensional SAR*

The collection geometry for 3-D SAR is depicted in Figure 7, where the azimuth angle θ and grazing angle ψ specify a particular orientation in the synthetic aperture from which the scene is being viewed. Following the procedure of [**40**], suppose a linear FM chirp pulse, given by

$$s(t) = \begin{cases} e^{j(\omega_0 t + \alpha t^2)} & \text{for } |t| \leq T/2 \\ 0 & \text{otherwise} \end{cases}$$

is transmitted, where 2α is the FM chirp rate, T is the pulse duration, and ω_0 is the RF carrier frequency. The returned echo from orientation (θ, ψ) is

$$r_{\theta,\psi}(t) = A \cdot \mathrm{Re}\left\{ \int_{-L}^{0} p_{\theta,\psi}(u)\, s\left(t - \frac{2(R+u)}{c} \right) du \right\} \tag{7}$$

for t on the interval:

$$-\frac{T}{2} + \frac{2R}{c} \leq t \leq \frac{T}{2} + \frac{2(R-L)}{c} \tag{8}$$

where L denotes the radius of the target and R is the distance from the antenna to the origin of the coordinate system of the target. The projection func-

tion $p_{\theta,\psi}(\cdot)$ in (7) is a set of integrals of the 3-D reflectivity density function $f(x,y,z)$ over the planar surfaces in Figure 6, and also illustrated in Figure 7, as opposed to line integrals as in Figure 1. It was shown in [**40**] that the radar return signal, after mixing with the reference chirp (delayed by $\tau_0 = 2R/c$) and low-pass filtering, is proportional to the Fourier transform of the projection function. Similarly, demodulating the returned echo in (7) gives:

$$d_{\theta,\psi}(t) \;=\; \frac{A}{2} \cdot P_{\theta,\psi}\left(\frac{2}{c}\Big(\omega_0 + 2\alpha(t-\tau_0)\Big)\right)$$

According to the linear-ray theorem, $d_{\theta,\psi}(t)$ must represent a ray at angular orientation (θ,ψ) of the 3-D Fourier transform of the unknown reflectivity $f(x,y,z)$. Since the processed return $d_{\theta,\psi}(t)$ is available only on the interval defined by (8), $P_{\theta,\psi}(X)$ is determined only for $X_1 \le X \le X_2$ with

$$\begin{aligned} X_1 &= \frac{2}{c}\Big(\omega_0 - \alpha T\Big) \\ X_2 &= \frac{2}{c}\left(\omega_0 + \alpha T - \frac{4\alpha L}{c}\right) \;\approx\; \frac{2}{c}\Big(\omega_0 + \alpha T\Big) \end{aligned}$$

At this point, tomographic principles can be exploited to understand and interpret the image that is formed from the acquired Fourier data.

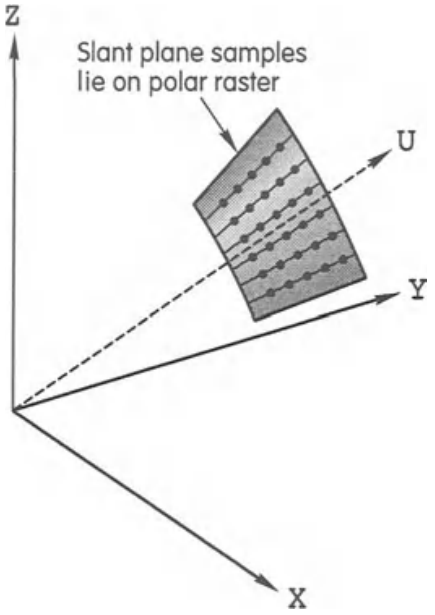


Figure 8. *Location of three-dimensional Fourier data on polar grid in the slant plane, as collected by a SAR with a linear flight path*

Consider the collection process in Figure 7 with a straight-line flight path, in which case the surface swept out in 3-D Fourier space by a sequence of demodulated returns will be a segment of a plane, referred to as the slant plane. Collecting returns in this manner provides samples of the function $F(X,Y,Z)$ on the polar grid in the annulus segment shown in Figure 8.

The usual procedure for forming a slant-plane image is to inverse Fourier transform the 2-D array of data in this plane, after interpolation from the polar grid to a Cartesian grid. This image formation procedure places no restriction on motion through resolution cells, as our coordinate system was fixed on the target.

Let us now address the question of how this 2-D slant-plane image relates to the 3-D structure of the actual scene. Recall that the planar-slice theorem links a line projection of a 3-D object to an orthogonal planar slice of its Fourier transform. Therefore, the inverse 2-D transform of the interpolated slant-plane Fourier data will produce an image that is the projection of the 3-D target in the direction normal to the slant plane in the image space. From (5), this type of projection is a 2-D function in which each value corresponds to a line integral of the target reflectivity.

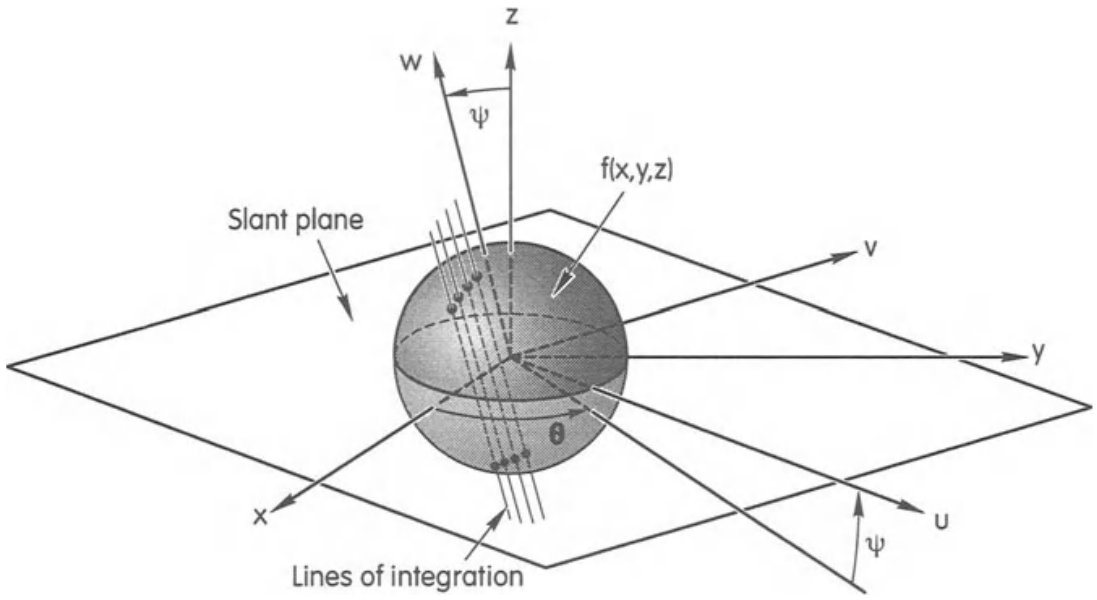


Figure 9. *Projection of a three-dimensional object onto the SAR slant plane*

Figure 9 portrays this situation. Notice that the w-axis specifies the direction of integration, while the u and v axii define the slant plane. Given the assumption made earlier that the 3-D reflectivity function is nonzero only on the target surface, line integration here is equivalent to the projection onto the slant plane of the two points on the surface (or possibly several discrete points if the surface is not spherical) where the surface is intersected by the line of integration. This is the same ambiguity between points A and B in Figure 5 that we discussed for the range-Doppler model.

3.3. Experimental results. We have applied the image formation procedure suggested by the 3-D SAR model (polar-to-Cartesian interpolation, followed by 2-D FFT) to radar data from the Moon [**56**]. Here we briefly report on a subset of these results. The lunar data set that we processed was collected at Arecibo Observatory (AO), which is located in Puerto Rico. Arecibo is one of the world's

largest filled-aperture antennas, with a 305-meter diameter and a steerable feed to steer the radar beam. Our data were collected on August 17, 1992, when the Moon was positioned high in the sky, almost directly above the main antenna. The radar illuminated only a portion of the upper hemisphere, where the impact basin Sinus Iridium is located. Generally the same side of the Moon always faces the Earth, but there is an apparent rotation of the Moon as it moves through the sky, due to the Earth's rotation. Also, the elliptical orbit of the Moon and the tilt of its rotational axis cause slight changes in the viewing angle from the Earth as a function of time.

We processed data from 16,384 pulses, sent at 11 Hz and collected over a 25-minute time interval. Because of the slow transmit-receive switch time of the main antenna, a nearby auxiliary antenna at Los Canos was used to receive the radar echo. During the data collection time, the viewing angle of Arecibo changed by only 0.1 degrees. As we shall see, however, this limited angular span for data collection is sufficient to provide for extraordinary resolution. The received echoes were demodulated, sampled, and then digitally pulse compressed. To simulate constant range from the image area, the data were phase corrected and resampled, so that the middle range bin always corresponded to the center of Sinus Iridium. We processed the digital data output from the pulse compression, and used only the middle 512 range bins. Along with the signal data, we were provided ephemeris data giving the location of AO relative to the center of Sinus Iridium at one-minute intervals. Examples of lunar images produced by conventional, and polar-format SAR processing are shown in Figure 10.

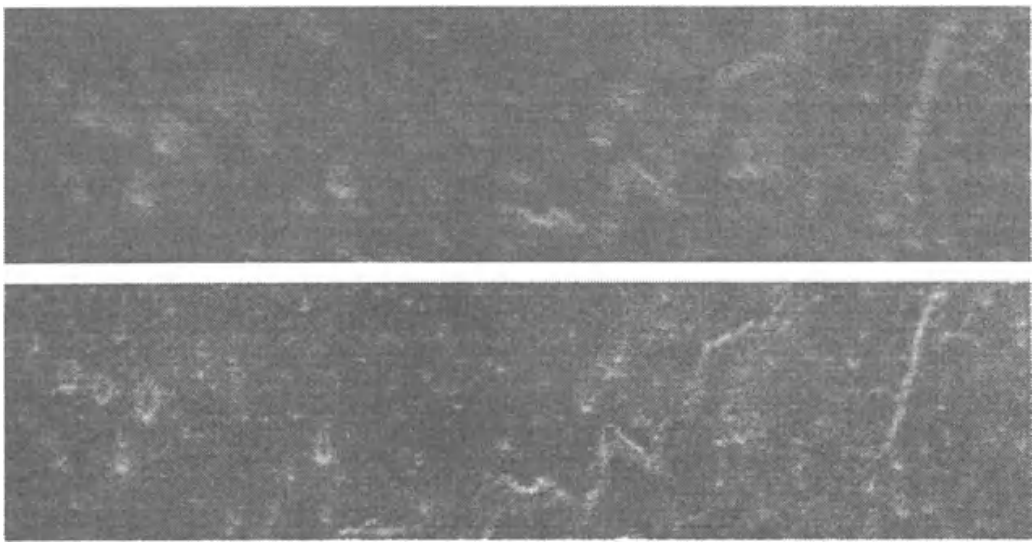


Figure 10. *Radar images of a portion of Sinus Iridium*

Top: Range-Doppler processing **Bottom:** Polar-format SAR processing

During the 25 minute data collection interval, there was enough range-cell migration and change in the apparent rotation to cause considerable smearing in the conventionally processed image. The ground resolution achieved is 17.9 m

in range, for an angle of incidence of 56.89 degrees, and about 40 m in cross-range. The image formed lies within the area illuminated by the radar, which has a 3-DB footprint on the lunar surface of 245 km in cross-range. For display purposes, only a portion of each 512×16 K image is shown, taken near the boundary of the illuminated area, where blurring is most noticeable. The image resolution was reduced by applying a 4×4 boxcar filter to each image before down-sampling by 4 in both dimensions. The resulting image displays 71.68 km in cross-range and 8.02 km in ground range; the center of this image is displaced 128 km in cross-range from the center of Sinus Iridium. On this scale, the polar-format processing results show considerable improvement over that using the conventional range-Doppler (matched filtering) processing. We emphasize that this improvement is due to a different image reconstruction algorithm, which was motivated by a (more accurate) tomographic model for the data collection. Both image reconstruction methods use the same data. The polar-format SAR algorithm requires about three times the amount of computation needed for conventional range-Doppler processing. We are hopeful that this new algorithm for image formation in radar astronomy will become standard in this field. This work illustrates the potential utility in applying ideas from one subfield of computational imaging to another.

4. Progress toward a unified field

Two groups of researchers have driven the development of the field of computational imaging. The first group is characterized by their ties to specific imaging applications and modalities. Researchers from this group (often physicists) have pioneered the development of new imaging modalities in an individual way. Each application area initially has developed its own terminology and its own approach to data modeling and inversion. In recent years, a second group of researchers has arrived, drawn primarily from the signal and image processing community. Researchers in this second group have an interest in applications, but are experts on Fourier transforms, convolution, spectral analysis, linear algebra, statistical signal processing, and the implementation of the associated numerical operations. Signal processing researchers who have worked on more than one imaging application are in a particularly good position to assist in unifying the field of computational imaging. Textbooks in this field have just recently become available [**5, 49**].

Richard E. Blahut has played a leading role in the unification of this field. His major work in the area began with his conception of and research on emitter location algorithms and systems, at what was then IBM Federal Systems in Owego, New York. The great success of this work led to his appointment as an IBM Fellow and caused him to think more deeply about remote surveillance and computational imaging. For many years* he taught a course in this area

*N. Stacy, one of R.E. Blahut's students at Cornell, was the person who first introduced us to the radar astronomy problem.

at Cornell University. He has continued his teaching activity in computational imaging since moving to the University of Illinois. His course is based on a draft manuscript for his forthcoming textbook [4] titled *Remote Surveillance.*

Signals in One Dimension
One-dimensional Fourier transform, baseband and passband signals and sampling, matched filter

Signals in Two Dimensions
Two-dimensional Fourier transform, projection-slice theorem, sampling, two-dimensional filtering, resolution

Optical Imaging Systems
Diffraction, Huygens principle, Fresnel and Fraunhofer approximations, geometric optics

Antenna Systems
Aperture and pattern, arrays, interferometry

Ambiguity Function
Theory and properties, cross-ambiguity function

Radar Imaging Systems
Imaging equation, resolution, focusing, motion compensation

Radar Search Systems
Coherent and noncoherent detection, Neyman-Pearson test, clutter, moving targets

Data Combination and Tracking
Maximum-likelihood sequential detection, assignment problem

Diffraction Imaging Systems
Three-dimensional Fourier transform, X-ray diffraction, coded aperture imaging

Image Reconstruction
Deblurring and deconvolution, phase retrieval, imaging from point events

Tomography
Back-projection, synthetic aperture radar, diffraction tomography, emission tomography

Passive Surveillance Systems
Radio astronomy, passive direction finding and emitter location

Baseband Surveillance Systems
Wideband arrays and beamformers, wideband ambiguity function, magnetic surveillance, magnetic resonance imaging

Table 2. *Table of contents for Remote Surveillance, by Richard E. Blahut*

A partial table of contents for [4] is shown in Table 2. The reader will notice a broad range of topics, which taken together, form the basis for a mathematical

understanding of the field of computational imaging. Indeed, many of these topics arise naturally in the radar astronomy problem. The publication of this text will, no doubt, represent a major contribution to the imaging community by providing the most unified treatment to date.

Acknowledgment. The author would like to credit Richard Leahy for co-authoring with him a section of [9], which formed the basis for § 2 of this chapter.

References

[1] C. AHN, J. KIM, AND Z. CHO, High-speed spiral-scan echo planar NMR imaging, *IEEE Trans. Medical Imaging*, vol. 5, pp. 2–7, 1986.

[2] D.A. AUSHERMAN, A. KOZMA, J.L. WALKER, H.M. JONES, AND E.C. POGGIO, Developments in radar imaging, *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-20, pp. 363–400, July 1984.

[3] H.H. BARRETT AND W. SWINDELL, *Radiological Imaging*, Vols. I and II, Academic Press, New York, NY, 1981.

[4] R.E. BLAHUT, *Theory of Remote Surveillance*, ECE 458 Course Notes, University of Illinois, Urbana, IL, 1997.

[5] R.N. BRACEWELL, *Two-dimensional Imaging*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[6] ———, Strip integration in radio astronomy, *Austr. J. Phys.*, vol. 9, pp. 198–217, 1965.

[7] C. CAFFORIO, C. PRATI, AND F. ROCCA, SAR data focusing using seismic migration techniques, *IEEE Trans. Aerospace Electronic Systems*, vol. 27, pp. 194–207, March 1991.

[8] W.G. CARRARA, R.S. GOODMAN, AND R.M. MAJEWSKI, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*, Artech House, Norwood, MA, 1995.

[9] R. CHELLAPPA, B. GIROD, D.C. MUNSON, JR., A.M. TEKALP, AND M. VETTERLI, (Eds.), The past, present, and future of image and multidimensional signal processing, *IEEE Signal Processing Magazine*, vol. 15, pp. 21–58, 1998.

[10] H. CHOI AND D.C. MUNSON, JR., Direct-Fourier reconstruction in tomography and synthetic aperture radar, *Int. J. Imaging Systems and Technology*, vol. 9, pp. 1–13, 1998.

[11] M.R. COBLE AND D.C. MUNSON, JR., Inverse SAR planetary imaging, *Proc. Image Processing: Theory and Applications*, San Remo, Italy, June 14-16, 1993, Elsevier Science Publishers, the Netherlands, pp. 31–39.

[12] B. CORNUELLE, ET AL., Tomographic maps of the ocean meoscale. Part I: Pure acoustics, *J. Phy. Ocean*, vol. 15, pp. 133–152, 1985.

[13] R.A. CROWTHER, D.J. DEROSIER, AND A. KLUG, The reconstruction of three-dimensional structure from projections and its applications to electron microscopy, *Proc. Roy. Soc. London (A)*, vol. 317, pp. 319–340, 1970.

[14] J.C. CURLANDER AND R.N. MCDONOUGH, *Synthetic Aperture Radar: Systems and Signal Processing*, John Wiley and Sons, New York, NY, 1991.

[15] M. DEFRISE, P. KINAHANA, D. TOWSEND, C. MICHEL, M. SIBOMANA, AND D. NEWPORT, Exact and approximate rebinning algorithms for 3D PET data, *IEEE Trans. Medical Imaging*, vol. 16, pp. 145–158, 1997.

[16] A.H. DELANEY, Efficient Edge-Preserving and Multiresolution Algorithms for Limited-Data Tomography, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1995.
[17] K. DINES AND R. LYTLE, Computerized geophysical tomography, *Proc. IEEE*, vol. 67, pp. 1065–1073, 1979.
[18] D.E. DUDGEON AND R.M. MERSEREAU, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[19] A. DZIEWONSKI AND J. WOODHOUSE, Global images of the earths interior, *Science*, vol. 236, pp. 37–48, 1987.
[20] J.A. FESSLER, Penalized weighted least-squares image reconstruction for positron emission tomography, *IEEE Trans. Medical Imaging*, vol. 13, pp. 290–300, 1994.
[21] S. GEMAN AND D.E. MCCLURE, Bayesian image analysis: An application to single photon emission tomography, *Proceedings of Statistical Computing Section of the American Statistical Association*, pp. 12–18, 1985.
[22] G. GINDI, M. LEE, A. RANGARAJAN, AND I.G. ZUBAL, Bayesian reconstruction of functional images using anatomical information as priors, *IEEE Trans. Medical Imaging*, vol. 12, pp. 670-680, 1993.
[23] P.E. GREEN, JR., Radar Astronomy Measurement Techniques, Technical Report 282, MIT Lincoln Laboratory, Lexington, MA, Dec. 1962.
[24] T. HAGFORS AND D.B. CAMPBELL, Mapping of planetary surfaces by radar, *Proc. IEEE*, vol. 61, pp. 1219–1225, September 1973.
[25] G.T. HERMAN, *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*, Academic Press, New York, NY, 1980.
[26] G.T. HERMAN, A.V. LAKSHMINARAYANAN AND A. NAPARSTEK, Convolution Reconstruction Techniques For Divergent Beams, *Comp. Biol. Med*, vol. 6, pp. 259–271, 1976.
[27] C.V. JAKOWATZ AND P.A. THOMPSON, A new look at spotlight mode synthetic aperture radar as tomography: Imaging 3-D targets, *IEEE Trans. Image Processing*, vol. 4, pp. 699–703, May 1995.
[28] C.V. JAKOWATZ, JR., ET AL., *Spotlight-Mode Synthetic Aperture Radar, A Signal Processing Approach*, Kluwer Academic Publishers, Boston, MA, 1996.
[29] L.H. JENSEN AND G.H. STOUT, *X-ray Structure Determination*, John Wiley and Sons, New York, NY, 1989.
[30] C. JOHNSON, Y. YAN, R. CARSON, R. MARTINO, AND M. DAUBE-WITHERSPOON A system for the 3D reconstruction of retracted-speta PET data using the EM algorithm, *IEEE Trans. Nucl. Sci.*, vol. 42, pp. 1223-1227, 1995.
[31] A. KAK AND M. SLANEY, *Principles of Computerized Tomographic Imaging*, IEEE Press, New York, NY, 1988.
[32] P.E. KINAHAN AND J.G. ROGERS, Analytic 3D image reconstruction using all detected events, *IEEE Trans. Nucl. Sci.*, vol. 36, pp. 964–968, 1989.
[33] J.R. KLAUDER, The design of radar signals having both high range resolution and high velocity resolution, *Bell System Tech. Journal*, vol. 39, pp. 809–820, 1960.
[34] A. KUMAR, D. WELTI, AND R. ERNST, NMR Fourier Zuematography, *J. Magnetic Resonance*, vol. 18, pp. 69–83, 1975.
[35] M.F.C. LADD AND R.A. PALMER, *Structure Determination by X-ray Crystallography*, Plenum Press, New York, NY, 1985.
[36] P. LAUTERBUR, Image formation by induced local interactions: examples employing nuclear magnetic resonance, *Nature*, vol. 242, pp. 190-191, 1973.
[37] A. MACOVSKI, *Medical Imaging Systems*, Prentice Hall, Englewood Cliffs, NJ, 1983.
[38] R. MERSEREAU AND A. OPPENHEIM, Digital reconstruction of multi-dimensional signals from their projections, *Proc. IEEE*, vol. 62, pp. 1319–1338, 1974.
[39] E.U. MUMCUOGLU, R. LEAHY, S.R. CHERRY, AND Z. ZHOU, Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images, *IEEE Trans. Medical Imag.*, vol. 13, pp. 687–701, December 1994.
[40] D.C. MUNSON, JR., J.D. O'BRIEN, AND W.K. JENKINS, A tomographic formulation of spotlight-mode synthetic aperture radar, *Proc. IEEE*, vol. 71, pp. 917–925, August 1983.

[41] D.C. MUNSON, JR. AND J.L.C. SANZ, Image reconstruction from frequency offset Fourier data, *Proc. IEEE*, vol. 72, pp. 661–669, June 1984.
[42] J. OLLINGER AND J.A. FESSLER, Positron-emission tomography, *IEEE Signal Processing Magazine*, vol. 14, pp. 43–55, 1997.
[43] J. RADON, On the determination of functions from their integral values along certain manifolds, first published 1917 in German, translated and re-printed in *IEEE Trans. Medical Imag.*, vol. 5., pp. 170–176, 1986.
[44] E.A. ROBINSON AND S. TREITEL, *Geophysical Signal Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[45] R.E. SABBAGHA, *Diagnostic Ultrasound Applied to Obstetrics and Gynecology*, 3rd edition, J.B. Lippincott, Philadelphia, PA, 1994.
[46] L.A. SHEPP AND B.F. LOGAN, The Fourier reconstruction of a head section, *IEEE Trans. Nucl. Sci.*, vol. 21, pp. 21–33, 1974.
[47] L.A. SHEPP AND Y. VARDI, Maximum-likelihood reconstruction for emission tomography, *IEEE Trans. Medical Imaging*, vol. 1., pp. 113–122, October 1982.
[48] M. SOUMEKH, Band-limited reconstruction from unevenly spaced sampled data, *IEEE Trans. Acoustics, Speech, Signal Proc.*, vol. ASSP-36, pp. 110–122, January 1988.
[49] ———*Fourier Array Imaging*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[50] ———, A system model and inversion for synthetic aperture radar imaging, *IEEE Trans. Image Processing*, vol. 1, pp. 64–76, January 1992.
[51] H. STARK ET AL., An investigation of computerized tomography by direct Fourier inversion and optimum interpolation, *IEEE Trans. Biomed. Eng.*, vol. 28, pp. 496–505, 1981.
[52] H. STARK (EDITOR), *Image Recovery: Theory and Application*, Academic Press, Orlando, FL, 1987.
[53] A.R. THOMPSON, J.M. MORAN, AND G.W. SWENSON, JR., *Interferometry and Synthesis in Radio Astronomy*, John Wiley and Sons, New York, NY, 1986.
[54] J.L. WALKER, Range-Doppler imaging of rotating objects, *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-16, pp. 23–52, January 1980.
[55] S. WEBB (Editor), *The Physics of Medical Imaging*, Institute of Physics, London, 1988.
[56] J.L.H. WEBB, D.C. MUNSON, JR., AND N.J.S. STACY, High-resolution planetary imaging via spotlight-mode synthetic aperture radar, to appear *IEEE Trans. Image Processing*, vol. 7, Nov. 1998.
[57] D.R. WEHNER, *High Resolution Radar*, Artech House, Norwood, MA, 1994.
[58] C.H. WILCOX, The Synthesis Problem for Radar Ambiguity Functions, University of Wisconsin Math. Res. Center, Tech. Report 157, April 1960.

Index