EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Coding theory, information theory and cryptology : proceedings of the EIDMA winter meeting, Veldhoven, December 19-21, 1994

Citation for published version (APA):
Tilborg, van, H. C. A., & Willems, F. M. J. (Eds.) (1994). *Coding theory, information theory and cryptology : proceedings of the EIDMA winter meeting, Veldhoven, December 19-21, 1994*. Euler Institute of Discrete Mathematics and its Applications.

Document status and date:
Published: 01/01/1994

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 16. Nov. 2023

PROCEEDINGS OF THE

EIDMA Winter Meeting on Coding Theory, Information Theory and Cryptology

Henk C.A. van Tilborg and
Frans M.J. Willems (Eds.)

Veldhoven, December 19-21, 1994

Euler Institute of Discrete Mathematics
and its Applications

Eindhoven University of Technology

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Proceedings

Proceedings of the EIDMA winter meeting on coding theory,
information theory and cryptology, Veldhoven, December 19-21, 1994
/ Henk C.A. van Tilborg and Frans M.J. Willems (eds.). -
Eindhoven : Euler Institute of Discrete Mathematics and
its Applications, Eindhoven University of Technology. -
Ill., fig., tab.
With ref.
ISBN 90-75332-01-7
NUGI 811/832
Subject headings: coding theory / information theory / cryptology.

Contents

Source Coding

Special Channels

Combined Codes

Coding and Decoding

Cryptography

Algebraic Geometry Codes

Preface

By organizing the 1994 Winter Meeting on Coding Theory, Information Theory and Cryptology, the Euler Institute of Discrete Mathematics and its Applications continues a tradition that was started by Prof. Han Vinck who arranged winter meetings in 1991 and 1993 in Essen, Germany.

Again the primary intention of the meeting is to give young researchers the opportunity to present their results in front of an audience of scientists from various countries. In addition to this, senior scientists are encouraged to present survey papers or to point at new directions in research. Last but not least, this meeting should provide an atmosphere that allows the participants to communicate with each other in an informal way. We hope that the third winter meeting will be just as successful as the previous meetings both from a scientific and personal perspective.

At this point we would like to thank Mrs. Henny Houben who assisted in the organization and Phons Bloemen for his help in preparing these proceedings. The STIMULANS support for EIDMA from NWO made it possible to arrange this meeting.

Henk van Tilborg and Frans Willems,
Meeting organizers
December, 1994.

List Decoding of Convolutional Codes — A Tutorial

Rolf Johannesson

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden. email: rolf@dit.lth.se

Abstract — **In this tutorial, list (convolutional) decoding is considered. It is shown that the error performance depends on the early part of the distance profile and on the number of survivors kept, and not on the free distance or on the details of the code generators. Particularly, the encoder may be feed-forward systematic without loss. Furthermore, it is shown that this kind of encoder solves the correct path loss problem. Other kinds do not. Therefore only systematic encoders should be used with list decoders[1].**

I. Introduction

In Viterbi decoding we first choose a suitable code and then design the decoder in order to "squeeze all juice" out of the chosen code. In sequential decoding we choose a code whose encoder memory is long enough to warrant essentially error free decoding.

In *list decoding* (the M-algorithm) we first limit the resources of the *decoder*, then we choose an encoding matrix with a state space that is larger than the *decoder* state space. Thus, assuming the same decoder complexity, we use a more powerful code with list decoding than with Viterbi decoding.

List decoding is a non-backtracking breadth-first search of the code tree. At each depth only the L most promising subpaths are extended, not all, as is the case with Viterbi decoding. These subpaths form a *list* of size L. All subpaths on the list are of the same length and finding the L best extensions reduces to choose the L extensions with the largest values of the cumulative Viterbi metric.

II. The Correct Path Loss Problem and the Systematic vs. Nonsystematic Convolutional Codes Question

Since only the L best extensions are kept it can happen that the correct path is lost. This is a very severe event that causes many bit errors. If the decoder cannot recover a lost correct path it is of course a "catastrophe", i.e., a situation similar to the catastrophic error propagation that can occur when a catastrophic encoding matrix is used to encode the information sequence. The list decoder's ability to recover a lost correct path depends heavily on the type of *encoder* that is used.

A systematic encoder supports a spontaneous recovery. We will illustrate this by comparing the bit error probability for list decoders with various list sizes when they are used to decode sequences received over a BSC and encoded with both systematic and nonsystematic encoders that are equivalent over the first memory length. Both encoders have the same distance profile. The free distance of the systematic encoder is by far the least, yet its bit error probability is more than ten times better! The only advantage of the nonsystematic encoder is its larger free distance. Yet this extra distance has almost no effect on neither the burst nor the bit error probability. Nor does it change the list size L needed to correct e errors, as long as e falls within the powers of the systematic encoder.

In conclusion, using feed-forward systematic convolutional encoders essentially solves the correct path loss problem with list decoders. Since both systematic and nonsystematic encoders have the same error rate in the absence of correct path loss, systematic encoders are clearly superiour to nonsystematic ones.

III. The List Minimum Weight

Consider a list decoder with a fixed list size L. For every depth $t = 0, 1, \dots$ and every received sequence $\underline{r}_{[0,t]} \in \mathbb{F}_2^{(1+t)c}$ let $\rho_L(\underline{r}_{[0,t]})$ denote the largest radius of a sphere with center $\underline{r}_{[0,t]}$ such that the number of codewords in the sphere is less than or equal to L. The smallest such radius is of particular significance:

For a list decoder with a given list size L the *list minimum weight* $w_{\min}$ is

$$w_{\min} = \min_{t} \min_{\underline{r}_{[0,t]}} \rho_L(\underline{r}_{[0,t]}),$$

where $\underline{r}_{[0,t]}$ is the initial part of the received sequence $\underline{r}$.

Given a list decoder of list size L and a received sequence with at most $w_{\min}$ errors. Then the correct path will not be forced outside the L survivors.

Unfortunately, $w_{\min}$ is hard to estimate. This leads us to restrict the minimization to those received sequences that are codewords:

For a given list size L the *list weight* w_{list} of the convolutional code C is

$$w_{\text{list}} = \min_{t} \min_{\underline{v}_{[0,t]}} \rho_L(\underline{v}_{[0,t]}),$$

where $\underline{v}_{[0,t]}$ is the initial part of the codeword $\underline{v} \in C$.

The list minimum weight $w_{\min}$ is upper and lower bounded by w_{list} according to

$$\left\lfloor \frac{1}{2} w_{\text{list}} \right\rfloor \leq w_{\min} \leq w_{\text{list}}.$$

Given a list decoder of list size L and a received sequence with at most $\lfloor \frac{1}{2} w_{\text{list}} \rfloor$ errors. Then the correct path will not be forced outside the L survivors. If the number of errors exceeds $\lfloor \frac{1}{2} w_{\text{list}} \rfloor$, then it depends on the code C and on the received sequence $\underline{r}$ whether the correct path is forced outside the list.

By bounding the list minimum weight we can show that the required list size grows exponentially with the number of errors to be corrected!

References

[1] Osthoff, H., Anderson, J.B., Johannesson, R., and Lin, C.-F.: "Systematic Feed-Forward Convolutional Encoders Are as Good as Other Encoders with an M-algoritm Decoder". In preparation.

[1] This work was supported in part by the Swedish Research Council for Engineering Sciences under Grants 92-661 and 94-83.

Proof of the Completeness of Bi-infinite Convolutional Codes

Emma Wittenmark, Zhe-xian Wan

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden. email: emma@dit.lth.se

Abstract — **A conventional convolutional code defined with one-sided infinite input sequences is known to be incomplete by Willems' definition of completeness [1]. However, a convolutional code C defined with bi-infinite input sequences is complete. This can be shown with the help of symbolic dynamics. This paper presents a self-contained proof of the completeness of such convolutional codes.**

The paper also includes a definition of a state realization of convolutional codes defined in this way[1].

I. Notation

Let C be a rate $R = k/n$ convolutional code and let $G(D)$ be a polynomial encoding matrix of C. Then $G(D)$ is $k \times n$ of rank k and C is the set

$$C = \{x(D)G(D) | x(D) \in \mathbb{F}_2{}^k(((D)))\} \tag{1}$$

where $\mathbb{F}_2(((D)))$ is the set of bi-infinite sequences over the signal alphabet $\mathbb{F}_2$ and $\mathbb{F}_2{}^k(((D)))$ is the k-dimensional row vector space over $\mathbb{F}_2(((D)))$.

II. Completeness

In [1], Willems defines a code C (or a system) to be complete if any sequence $\mathbf{w} \in \prod_{t \in \mathbb{Z}} \mathbb{F}_2{}^n$ such that $\mathbf{w}|_I \in C|_I$ for any finite interval I of $\mathbb{Z}$ implies that $\mathbf{w} \in C$. This means that a code (system) is complete if it is locally defined. Convolutional codes, defined conventionally with input sequences of Laurent-type, are known to be incomplete. However, by extending the definition of convolutional codes to be the set of bi-infinite sequences we have,

Theorem 1 *The bi-infinite convolutional code C is complete.*

Proof: Assume $c(D) = \sum_{i=-\infty}^{i=\infty} c_i D^i$ any word such that for any finite interval I, $\exists y_I(D) \in C$ such that $y_i = c_i, \quad \forall i \in I$. In particular, we have for $I = \{0\}$, $\exists x^{(0)}(D) = \sum_{i=-\infty}^{i=\infty} x_i^{(0)} D^i$ such that

$$c_0 = x_{-m}^{(0)} G_m + x_{-m+1}^{(0)} G_{m-1} + \ldots + x_0^{(0)} G_0. \tag{2}$$

For $I = \{-n, \ldots, n\}$, $\exists x^{(n)}(D)$ such that

$$\begin{aligned} c_{-n} &= x_{-m-n}^{(n)} G_m + x_{-m-n+1}^{(n)} G_{m-1} + \ldots + x_{-n}^{(n)} G_0 \\ &\vdots \\ c_n &= x_{-m+n}^{(n)} G_m + x_{-m+n+1}^{(n)} G_{m-1} + \ldots + x_n^{(n)} G_0 \end{aligned}$$

etc.

There exists an infinite set S_0 of nonnegative integers such that all $(x_{-m}^{(n)}, x_{-m+1}^{(n)}, \ldots, x_0^{(n)})$ equal the same vector $\forall n \in S_0$. Also, there exists an infinite subset $S_1 \subset S_0$ such that all $(x_{-m-1}^{(n)}, x_{-m}^{(n)}, \ldots, x_0^{(n)}, x_1^{(n)})$ equal the same vector $\forall n \in S_1$. We then have a sequence of nested infinite sets $S_0 \supset S_1 \supset S_2 \supset \ldots$ such that for each $l = 0, 1, 2, \ldots$ all $(x_{-m-l}^{(n)}, \ldots, x_l^{(n)})$ equal the same vector $\forall n \in S_l$.

Now, define $x' = \ldots x'_{-m} x'_{-m+1} \ldots x'_0 x'_1 \ldots$ by

$$(x'_{-m-l}, \ldots, x'_l) = (x_{-m-l}^{(n)}, \ldots, x_l^{(n)}) \quad \forall n \in S_l. \tag{3}$$

x' is well-defined since we have a nested sequence of subsets $S_l \supset S_{l+1}$.

Given any $l \geq 0$, $\exists n \geq l$ and $n \in S_k$ for some k such that

$$\begin{aligned} c_{-n} &= x_{-m-n}^{(n)} G_m + x_{-m-n+1}^{(n)} G_{m-1} + \ldots + x_{-n}^{(n)} G_0 \\ &\vdots \\ c_{-l} &= x_{-m-l}^{(n)} G_m + x_{-m-l+1}^{(n)} G_{m-1} + \ldots + x_{-l}^{(n)} G_0 \\ &\vdots \\ c_n &= x_{-m+n}^{(n)} G_m + x_{-m+n+1}^{(n)} G_{m-1} + \ldots + x_n^{(n)} G_0. \end{aligned}$$

We have,

$$\begin{aligned} c_{-l} &= x_{-m-l}^{(n)} G_m + x_{-m-l+1}^{(n)} G_{m-1} + \ldots + x_{-l}^{(n)} G_0 \\ &= x'_{-m-l} G_m + \ldots + x'_{-l} G_0 \quad \forall n \in S_l. \end{aligned} \tag{4}$$

Thus, $c(D) = (\sum_{i=-\infty}^{i=\infty} x_i D^i)(G_0 + G_1 D + \ldots + G_m D^m)$ and is then a codeword in the code C, and the proof is complete. □

III. State Realization

For any input sequence $x(D)$, define an abstract state of the code C relative to the encoding matrix at time t to be the output from the encoding matrix truncated to start at time t due to an input sequence truncated to end at time $t-1$. Denote the abstract state by $s_t(D)$. Also, let $c_t \in \mathbb{F}_2{}^n$ be the t-th coefficient of a codeword $c(D) = x(D)G(D) \in C$ and $B_{G(D)}$ to be the set

$$B_{G(D)} = \{((c_t, s_t(D)), t \in \mathbb{Z}) | \mathbf{c} \in C\}. \tag{5}$$

Then $B_{G(D)}$ is a state realization of the code C. For the definition of state realization, see [2]. Moreover, when $G(D)$ is minimal-basic, $B_{G(D)}$ is the minimal state realization of C.

References

[1] Jan C. Willems, "Models for Dynamics", in *Dynamics Reported*, vol. 2, John Wiley and Sons, 1989.

[2] G. David Forney Jr, Mitchell D. Trott, "The Dynamics of Group Codes: State Spaces, Trellis Diagrams and Canonical Encoders", *IEEE Trans. on Information Theory*, 39, 1993.

[1] This research was supported in part by the Swedish Research Council for Engineering Sciences under Grants 92-661 and 94-77.

An Alternate Metric for Sequential Decoding

Gerhard Krämer and Dirk J. Tempel
Institute for Signal and Information Processing
Swiss Federal Institute of Technology Zürich
CH-8092 Zürich, Switzerland

The Fano metric is almost universally applied for the sequential decoding of codes whose state transitions in time can be described by a tree. The justification for using this metric was given in [1] where it was shown that the Fano metric extends the most likely path for a model which specified that one knew nothing about the unexplored part of the tree. However, in [2] an alternate metric was presented, and in [3] it was found that this metric achieves the same error probability as the Fano metric with less searching for the sequential decoding of the Golay code.

It is shown that the alternate metric in [2, 3] can be derived using the same model as in [1] by maximizing the joint probability of the received signal, the considered path *and* the tail, in contrast to the Fano metric which maximizes the probability of the received signal and the considered path only. However, simulation results for punctured codes show that the alternate metric performs much *worse* than the Fano metric for codes of rate 1/2 to 7/8, and performs just as well for higher rate codes. Thus, it seems that this metric is of limited usefulness for the sequential decoding of convolutional codes unless they have a very high rate.

References

[1] J.L. Massey, "Variable-Length Codes and the Fano Metric," *IEEE Trans. Inform. Theory,* vol. IT-18, Jan. 1972, pp. 196-198.

[2] Z. Xie, C.K. Rushforth, and R.T. Short, "Multiuser signal detection using sequential decoding," *IEEE Trans. Commun.*, vol. COM-38, May 1990, pp. 578-583.

[3] D.J. Tempel, "Sequential Decoding of Linear Block Codes," M.Sc. Thesis, Dept. of Elec. and Comp. Eng., University of Manitoba, March 1993.

A Comparision of Different Metrices for GMD Decoding

Rainer Lucas

Communication Engineering Department, University of Ulm, Germany

It is well known that soft decisions on a AWGN channel gives better performance of channel coding. Maximum-Likelihood-Decoding (MLD) is the best method of minimizing the bit error probability but its complexity grows exponentially with the code length n. One decoding method which trade a slight degradation in performance for reducing the complexity is *Generalized Minimum Distance (GMD) Decoding* proposed by G.D. Forney.

This paper deals with the principle problems of GMD decoding.

The idea of GMD decoding is as follows. An algebraic error and erasure decoding algorithm Ω works correctly if $2t + s < d_{\min}$ holds (t: number of errors, s: number of erasures, $d_{\min}$: minimum distance of the considered code). By means of this algorithm Ω, try to find a codeword using the following steps:

- erase the s symbols of the received sequence (hard decision) which are less reliable.
- try to find a codeword by means of the algorithm Ω which holds

  $$d_H(\underline{h}, \underline{\tilde{c}}) < \frac{1}{2}(d_{\min} - s)$$

  where the Hamming distance of the $n - s$ non erased symbols is considered.

The result of this recursion is a set of codewords F_a that may be empty in the case of a decoding failure.

For any kind of list decoding there are two questions of interest:

1. Does the codeword $\underline{\tilde{c}}$ found by the algorithm Ω belong to the list F_a if a certain condition Λ between $\underline{\tilde{c}}$ and the received sequence $\underline{y}$ is fullfilled ?
2. Is it possible to choose the condtion Λ in such a manner that there is not more than a single codeword in the list F_a ?

This paper gives and compares known results to these questions. In Detail the acceptance criteria of Forney, Dumer, Enns (Taipale/Purley) Kabatyanskii and the recent result by Kaneko, Nishijima et al. are explained. In order to compare their decoding domains theoretically they will be represented in a uniform manner. Furthermore some simulation results will be presented.

European Transmission Standards for Digital TV Broadcasting

Paul G.M. de Bot

Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
(Fax: +31 40 742630; Email: debot@prl.philips.nl)

I. Introduction

Recently, practical systems for video source coding have been developed in the framework of the ISO/MPEG project. This effort has lead to a growing interest for introduction in Europe of digital broadcasting services in the near future. With this respect, we should distinguish between satellite direct-to-home distribution, cable network distribution and terrestrial distribution. Since these distribution media each have different channel characteristics and require different receiver equipment, different transmission mechanisms have to be designed, each optimized for a specific medium. All of these mechanisms enable the transport 24-40 Mbit/s in a single channel. Such a transport stream is sufficiently large to contain a number (4-8) of normal standard definition TV programs.

II. Satellite Distribution

Early this year, a draft European standard has been fixed, describing a transmission mechanism for TV broadcasting via satellite [2]. Satellite transmission is characterized by low available transmitter power, relatively high channel bandwidth (33-40 MHz), highly nonlinear transmitter amplification and a transmission channel which approaches the Additive White Gaussian Noise (AWGN) channel. For these reasons, QPSK modulation is chosen with powerful concatenated error correction coding consisting of a $\nu = 6$ convolutional code, interleaving and a [204,188,17] Reed-Solomon code. The $R = 1/2$ convolutional mother-code can be punctured to obtain rates of 2/3, 3/4, 5/6 and 7/8. The used interleaving is convolutional byte interleaving of depth $I = 12$. This depth is chosen such that at the receiver side a burst error at the output of the Viterbi decoder is sufficiently distributed over different Reed-Solomon codewords. To reduce the peak-to-average ratio of the transmitted signal, and to ease the synchronization in a receiver, Nyquist filtering with a relatively large roll-off of 35% is used. The Nyquist filter is equally split in a transmitter part an a receiver part.

III. Cable Distribution

For cable TV networks, another transmission standard is drafted this year [3]. The cable channel is characterized by a high signal-to-noise ratio, a strong bandwidth limitation (8 MHz), and short reflections due to impedance mismatches in the network. These constraints have lead to the choice of 64-QAM modulation, interleaving in combination with a single Reed-Solomon code. For compatibility reasons, the interleaving and Reed-Solomon coding are chosen the same as for the satellite system. To maximize the channel efficiency, Nyquist filtering with a roll-off as low as 15% is used, divided over a transmitter and a receiver part. To compensate for the channel reflections, channel equalization is needed in the receiver.

IV. Terrestrial Distribution

The terrestrial channel is for sure the worst and most difficult of the three channels discussed. For this reason, no draft standard has yet been fixed in Europe. Discussions all focus on the use of Orthogonal Frequency Division Multiplexing (OFDM), in contrast to the single carrier systems for satellite and cable [1]. OFDM uses up to 8192 narrowband carriers, which can be modulated each with for instance 64-QAM. This technique makes the reception less sensitive to the strong multipath nature of the channel. The error correction should be as strong as possible, to maximally extend the coverage area of the transmitter stations. Preferably, a concatenated coding scheme as for satellite should be used.

References

[1] P.G.M. de Bot, B. Le Floch, V. Mignone, and H.D. Schütte. An overview of the modulation and channel coding schemes developed for digital terrestrial television broadcasting within the dTTb project. In *Proc. Int. Broadcasting Convention*, pages 569–576, Amsterdam, The Netherlands, September 1994.

[2] ETSI. Channel coding and modulation for 11-12 GHz satellite receivers, April 1994. prETS 300xxx/6.

[3] ETSI. Framing structure, channel coding and modulation for CATV cable and smatv distribution, May 1994. prETS 300xxx/7.

Decoding of Concatenation of Outer Convolutional Code with Inner Orthogonal Code

Thomas Frey

Communication Engineering Department, University of Ulm, Germany

Channel estimation in the uplink of a CDMA system is a difficult problem. One possibility to avoid this problem is the use of incoherently detectable modulation schemes, eg. M-ary orthogonal modulation. In order to reduce the bit error rate to an acceptable level, in addition channel coding has to be employed. This paper deals with possible demodulation and decoding schemes.

An example for such an existing system may be the uplink of the Qualcomm system which uses the set of Walsh functions with dimension 64 as the orthogonal signaling system and a convolutional code of rate 1/3 and constraint length 9 as channel code. In terms of coding theory this system can be described as a concatenation of an inner orthogonal code with an outer convolutional code, whereby the inner orthogonal Walsh Hadamard code is a subset of a (biorthogonal) Reed-Muller code of first order. First order Reed-Muller codes are low-rate, which provide an inherent spreading and make them suitable for spread spectrum applications.

This paper investigates several possibilities of decoding this concatenated scheme. It is assumed that both inner and outer code are soft maximum-likelihood decoded, eg. by a fast Hadamard Transform or by a Viterbi decoder respectively. A comparision is done by using two types of reliability information between inner and outer code, a symbolwise (Walsh symbol with 6 bits) and a bitwise, whereby the bitwise performs slidely better. A major improvement is achived by maximum-likelihood decoding of the whole code, which can be performed by a Viterbi decoder using a new trellis with Walsh symbols as metric.

Using Codes to Design CDMA Signal Sets

Beat Keusch
Signal and Information Processing Laboratory
Swiss Federal Institute of Technology
CH–8092 Zürich
email: keusch@isi.ee.ethz.ch

Most approaches to efficient coding for the CDMA channel are simply too complex to implement. This complexity stems from the necessity to perform joint decoding of all users which requires a decoding algorithm that is able to track all possible states of all encoders.

An alternative approach to coding that aims to achieve reasonable efficiency while avoiding the impractical complexity of optimum decoding is based on a 'modulation engineering' point of view:

First, we employ a partial demodulator scheme in order to segregate the users into small, roughly independent groups. Coding for one user group subsequently ignores the specific codes for all other groups and thus reduces to coding for the noisy (binary) adder channel.

As the second step in our approach, we devise good, small signal sets or 'codes' for each group in such a way that the resulting 'virtual channels' for each user exhibit substantial independence that allows independent single-user decoding with little loss of optimality: The modulation system should be designed to create a 'good' channel for coding whereby the capacity of this discrete channel consisting of the modulators, the noisy real adder channel and the demodulator is only slightly decreased and the users are separated to some extend. We investigate some information-theoretic aspects of this signal set design problem. Some examples are used to show that small, uniquely decodable or larger, 'almost' uniquely decodable multi-user block codes create a 'virtual channel' and should be considered not as codes but rather as multi-user signal sets for use with a coding alphabet of the same size.

Modified Delay-Locked-Loop Structures for PN-Code Tracking

Andreas Wilde
German Aerospace Research Establishment (DLR), NE-NT, D-82234 Wessling, Germany

Abstract — **In direct sequence (DS) spread spectrum systems the pseudo-noise (PN) code tracking with a delay-locked loop (DLL) is commonly used. Some modifications to the conventional DLL structure are described and their performance is evaluated.**

I. Introduction

In spread spectrum systems PN-code tracking is a crucial performance aspect. The DLL is an appropriate device to guarantee fine synchronization. The code phase estimate is produced by comparing the received signal with both the early and late replicas of the locally generated PN reference sequence. The code phase timing error drives the clock of a PN-code generator to adjust the code phase timing. This is a closed-loop tracking control system. The conventional DLL has been studied in the literature in detail [1]. Extended tracking range DLLs have been described in [2] and [3]. Please note, that in this paper we always refer to coherent DLLs using maximal-length sequences as the spreading codes. However, the modifications can also be applied to noncoherent tracking loops.

II. Conventional DLL

The received signal is multiplied by the early and late replicas of the local PN-code. The spacing between the early and late replicas is $\Delta\ T_c$. The parameter Δ is the total normalized time difference between the early and late discriminator branches. The two branches are low pass filtered (LPF) to perform the autocorrelation over the PN-code and then they are subtracted. The result is the error signal $e(t)$. The loop filter generates the input signal for the voltage controlled oscillator (VCO) steering the local PN-code generator. The delay-lock discriminator dc output D_Δ is plotted as a function of the timing error ε in the so called S-curve. The S-curve describes the structural characteristic of the DLL.

III. Modified Coherent DLL

An improvement to the conventional loop can be achieved by reducing the noise power in the loop. This can be done by selecting one of the two correlation branches in the DLL instead of taking both [4]. This will change the detector S-curve only slightly for a Δ spacing of 2. The difference is due to the out-of-lock correlation of the inactive branch which is very small for long sequences having good autocorrelation properties. An important aspect for the realization of the modified DLL is that the expectation over the correlation should be performed before the selection device. This means that the LPF must be before the subtraction and can not be integrated in the loop filter as for the conventional DLL.

IV. Extended Tracking Range DLL

Extended tracking range DLL's have been proposed in [2] and [3]. The principle is to extend the overall tracking range by using more than two correlators to produce the loop error signal. However, each additional correlator increases the noise power in the loop. To reduce the noise power in the extended tracking range loop, a similar approach to the one used in section III. The input signal is despread with the time-shifted replicas of the code and low pass filtered to perform the autocorrelation over the PN-code. The autocorrelation values of the different branches are then processed in a select/combine block. The algorithm selects the two strongest branches and combines them with the correct sign. The result is the loop error signal which is fed back to control the local PN-code generator. The S-curve for this modified extended tracking range DLL is almost identical to the one of the conventional extended tracking range DLL. The number of correlators can easily be increased without influencing the jitter performance of the loop. The number of correlators is only limited by the processing overhead.

V. Comparison of Loop Performance

The tracking jitter for the various loop structures has been calculated by linear analysis. The linear analysis holds for small tracking errors and white Gaussian input noise. This means that the detector output is always kept in the linear tracking range. The linear tracking range is defined to be that region where D_Δ depends linearly on the tracking error ε. The overall tracking range is the region where the detector produces an usable control signal D_Δ to drive the VCO through the loop filter. Outside the overall tracking range the loop must be considered out-of-lock and a re-acquisition should be started. The modification of the coherent 2Δ-DLL gives a reduction in tracking jitter of 3 dB compared to the conventional 2Δ-DLL. The modified extended tracking range DLL enlarges the overall tracking range arbitrarily while conserving the jitter performance of the conventional DLL.

VI. Conclusion

Modifications to the conventional DLL structure can improve the performance of the DLL significantly. This is achieved by reducing the noise power in the tracking loop. Further investigations to apply the principle of limiting the noise power in the loop will be carried out.

References

[1] R.E. Ziemer and R.L. Peterson, "Digital Communications and Spread Spectrum Systems", Macmillan, NY 1985.

[2] W.M. Bowles, "GPS Code Tracking and Acquisition Using Extended-Range Detection", NTC Record, Houston 1980.

[3] K. Wakabayashi, M. Nakagawa et al., "Tracking Performance of Improved Delay-Locked Loop", NTC Record, Houston 1980.

[4] A. Wilde, "Modified Coherent PN-Code Tracking Delay-Locked Loop", PIMRC'94, Den Haag.

On Binary DC-free Parity-Check Codes

Volker Braun

Institute for Experimental Mathematics, Ellernstr. 29, 45326 Essen, Germany; e-mail: volker@exp-math.uni-essen.de

Abstract — **A rate 8/10 binary DC-free Parity-Check (PC) code is presented. We determine its soft decision error rate performance and present examples of feasible code rates of other DC-free PC codes.**

I. Preliminaries

Binary codes with a spectral null at the zero frequency, i.e., DC-free codes, have been widely applied [1-4]. We define the *Running Digital Sum (RDS)* of the encoded sequence $\{x_i\} = \{\dots, x_{-1}, x_0, \dots, x_i, \dots\}$, $x_i \in \{-1, 1\}$, as

$$z_i = \sum_{j=-\infty}^{i} x_j = z_{i-1} + x_i.$$

RDS-constrained codes are characterized by the property that their RDS remains within a bounded range, i.e., the RDS of the encoded sequence takes a finite number of values. This number is called the *Digital Sum Variation (DSV)* and is denoted by N. RDS-constrained codes are DC-free codes, i.e., the encoded sequences have high-pass characteristics with a spectral null at the zero frequency. A tutorial description of RDS-constrained codes including several examples can be found in [1]. We consider codes which map m source bits to n channel symbols. We define the *code rate* $R = m/n$ and the *rate efficiency* $\eta = R/C(N)$, where $C(N)$ denotes the *noiseless capacity* of a sequence occupying N sum states [1]. We confine ourselves to encoders having 2 states. From [1], we know that such encoders lead to relatively high rate efficiencies. A rate 8/10 RDS-constrained code based on a 2-state encoder, for example, achieves the maximum rate efficiency possible for this rate ($\eta \approx 94.2\%$).

The construction of RDS-constrained codes implies a *free Hamming distance* of 2. In order to exploit this free distance, several authors [2-4] considered the application of rather complex sequence estimation algorithms. We propose RDS-constrained codes with *minimum Hamming distance* 2 which hence allow the application of the *Soft Decision Parity-Check (SDPC)* (or 'Wagner') decoding algorithm [5]. The computational complexity of this algorithm is known to be very low and it is independent of the DSV of the code. Further, error propagation is limited to m decoded source bits.

II. Rate 8/10 Code

The minimum DSV of a rate 8/10 DC-free Parity-Check (PC) code equals $N = 9$, thus we call it '$N9$ code'. The $N9$ code is a subset of all paths through the *RDS trellis* [1] in Fig. 1. Since all codewords have odd *weight*, the minimum Hamming distance is 2. The maximum number of consecutive like symbols in the channel string can be limited to 5.

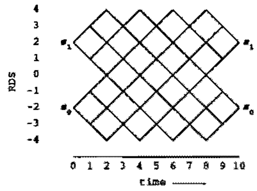


Figure 1: RDS trellis of a rate 8/10 DC-free PC code

The error rates for the $N9$ code in the presence of additive white Gaussian noise have been determined by means of computer simulations in the case of Maximum Likelihood Sequence Estimation (MLSE) or SDPC decoding. For symbol error rates in the order of 10^{-5}, MLSE leads to a gain of about 2.8 dB, and SDPC decoding to a gain of about 2.5 dB versus threshold detection level. Since the $N9$ code is a subset of the complete set of 2^9 odd-weight patterns of length 10, the SDPC decoding algorithm results in a scheme which is not optimal in the maximum-likelihood sense.

III. Generalization

We consider DC-free PC codes having arbitrary (even) codeword length n and DSV N. Our codes are based on 2-state RDS trellises whose principal states (s_0 and s_1) are associated with RDS values of -2 or 2 (see Fig. 1). We truncate the number of RDS trellis paths to the nearest power of 2 in order to determine the code rates feasible. Examples of feasible code rates are given in Table 1.

Table 1: Examples of feasible code rates

N	codeword length n						
	8	10	12	14	16	18	20
7	6/8	7/10	9/12	11/14	13/16	14/18	16/20
8	6/8	7/10	9/12	11/14	13/16	15/18	16/20
9	6/8	8/10	10/12	12/14	13/16	15/18	17/20
10	6/8	8/10	10/12	12/14	13/16	15/18	17/20
11	6/8	8/10	10/12	12/14	14/16	16/18	17/20

Finally, we mention several DC-free PC codes having relatively high rate efficiencies: for $N = 7$, we find a rate 6/8 code with $\eta \approx 84.7\%$, and a rate 13/16 code with $\eta \approx 91.7\%$. For codeword lengths in the range $10 \leq n \leq 14$, the choice $N = 9$ leads to relatively high rate efficiencies: codes having rates 8/10, 10/12, and 12/14 achieve $\eta \approx 86.2\%$, 89.8%, and 92.4%.

References

[1] K.A. Schouhamer Immink, *Coding Techniques for Digital Recorders*, Prentice Hall International (UK) Ltd, 1991.

[2] R. Wood, "Viterbi Reception of Miller-Squared Code on a Tape Channel," *Proc. of the Fourth Intern. Conf. on Video and Data Recording*, pp. 333-343, Southampton, April 1982.

[3] H. Thapar, J. Rae, C. Shung, R. Karabed, and P. Siegel, "On the Performance of a Rate 8/10 Matched Spectral Null Code for Class-4 Partial Response," *IEEE Trans. Magn.*, vol. MAG-28, no. 5, pp. 2883-2888, September 1992.

[4] V. Braun, K.A.S. Immink, M.A. Ribeiro, and G.v.d. Enden, "Sequence Estimation Algorithms for the Digital Compact Cassette (DCC)," *Proc. of the 1994 Intern. Symposium on Information Theory*, p. 210, Trondheim, June 1994.

[5] J.K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 1, pp. 76-80, January 1978.

New Optimum Distance Profile Trellis Codes

Per Ljungberg

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden. email: perl@dit.lth.se

Abstract — **New trellis codes over various lattice partitions having optimum distance profile (ODP) and encoders with large constraint lengths are constructed. They are attractive to use in combination with sequential decoding algorithms since their ODP property ensures good computational performance[1].**

I. Introduction

Trellis coded modulation (TCM) can achieve significant coding gain over uncoded transmission without any bandwidth expansion. For error rates of the order of 10^{-6}, the gap between the Shannon limit and uncoded high-rate QAM-signaling is approximately 9 dB, being the maximum possible coding gain for any coded modulation scheme operating in this region. A perhaps more realistic performance limit is the computational cut-off rate, R_0, beyond which the computational distribution for sequential decoding becomes unbounded. The possible coding gain under the R_0-criterium is 7.5 dB. It can be separated in two parts, fundamental coding gain and shaping gain [1]. The maximum values of these gains are approximately 6 dB and 1.5 dB, respectively.

The signal constellation can be viewed as a finite set of points from an infinite $2N$-dimensional lattice Λ. A sublattice Λ' of Λ induces a partition Λ/Λ' of Λ into $|\Lambda/\Lambda'|$ cosets of Λ'. In each time interval the output of a rate $R = \frac{k}{k+1}$ convolutional encoder is used to select one of the 2^{k+1} cosets. The uncoded bits then select one of the points in the specified coset. The fundamental coding gain is determined by the convolutional encoder and the lattice partition, whereas the shaping gain depends on the bounding region of the constellation.

Current implementations of TCM-systems all use the Viterbi algorithm (VA) for decoding the trellis code. The decoding effort of the VA is proportional to the number of states in the trellis, 2^ν, where ν is the overall constraint length of the convolutional encoder. These systems are thus restricted to have a relatively small number of states and can therefore not achieve the previously mentioned 6 dB. An example is the new modem standard V.34 where three different codes are available. The most complex encoder has 64 states and a coding gain of 4.6 dB.

The aim of this work is to increase the coding gain by increasing the number of states in the encoder. The decoding is then performed by a sequential decoder since its decoding effort is essentially independent of the number of states.

A major drawback of sequential decoders is that the number of computations is a random variable, thus complicating real-time implementations of systems using such algorithms. It is a well-known fact that the code should have an optimum distance profile (ODP) in order to minimize the average number of computations. As a first step we here report ODP trellis codes over various lattice partitions. Similar constructions for 8-PSK and 16-QAM can be found in [2].

II. Code search

It is convenient to search for good $R = \frac{k}{k+1}$ encoders on a systematic feedback form. The corresponding generator matrix is

$$G(D) = \left(\ I_k \quad | \quad G^{k-1}(D)/G^0(D)\ \right)$$

where

$$G^i(D) = g_0^i + g_1^i D + \cdots + g_\nu^i D^\nu$$

are polynomials in the delay operator D. The search was then performed as follows:

Assume that the set of ODP-encoders of constraint length ν is known. Form the 2^k possible extensions of every encoder on the list and calculate their distance profiles. Retain the encoders with the best distance profile, these form the set of ODP-encoders of constraint length $\nu + 1$.

To be able to make an accurate estimate of the error performance of a code, the number of paths at distances d_{free}^2 up to $d_{free}^2 + i$ should be computed for some small i. Since the number of encoders to be investigated is large, it is important for the algorithm to be efficient.

The FAST algorithm [3] is considered to be an efficient algorithm for computing the spectral components of a convolutional code. The extension to trellis codes includes a transformation of the systematic feedback encoders to equivalent non-systematic feedforward encoders, an operation of low complexity. The above transformation is not allowed if the bit error probability is considered since it changes the mapping from information sequence to codeword.

For each lattice partition the results are presented in a table containing the generator polynomials for one encoder for each constraint length. This encoder is the one having the best error performance of the encoders in the complete ODP-set.

References

[1] G. D. Forney, "Coset Codes - Part I: Introduction and geometrical classification", *IEEE Trans. Inform. Theory*, **IT-34**, pp. 1123-1151, September 1988.

[2] Fu-Quan Wang, "Efficient Sequential Decoding of Trellis Codes", Ph. D. Thesis, University of Notre Dame, December 1992.

[3] M. Cedervall and R. Johannesson, "A Fast Algorithm for Computing Distance Spectrum of Convolutional Codes", *IEEE Trans. Inform. Theory*, **IT-35**, pp. 1146-1159, November 1989.

[1] This research was supported in part by the Swedish Research Council for Engineering Sciences under grants 92-661 and 94-83.

On the Use of Concatenated Codes for 8-PSK Multilevel Coding

Joakim Persson

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden. email: mauritz@dit.lth.se

Abstract — **Simulation results for concatenated outer Reed-Solomon and inner Convolutional Codes used in multilevel schemes are presented. Different high rate inner convolutional codes were considered, viz., punctured codes and partial unit memory (PUM) codes. Best results were obtained for PUM codes, since they have a better extended row distance profile. The effect of channel and block interleaving was also studied, and iterative decoding was tried[1].**

I. Introduction

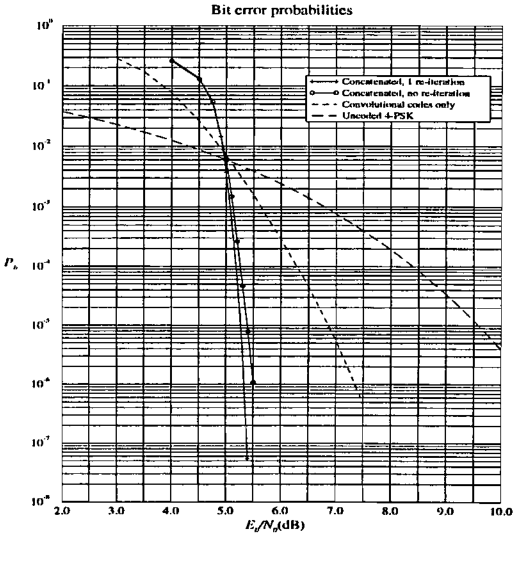
A multilevel code uses some signal set S_0 which is a finite subset of a lattice or a set of points with some group structure. This set is partitioned in a k-level partitioning chain, $S_0/S_1/\dots/S_k$. Each partition at level i, S_{i-1}/S_i, is determined by a component code C_i at this level. Using a multilevel approach when constructing codes makes it possible to achieve very large asymptotic coding gain in a systematic way. The codes also possess structural properties which are advantegous. Unfortunately, the decoding must by necessity be carried out in a way which is not maximum likelihood, otherwise the computational effort becomes far to large even for small systems (i.e., systems with not very complex component codes). The computational complexity of the preferred multistaged decoding procedure ([1]) is proportional to the sum of the complexity of each component code, but it suffers from error propagation. In order to minimize the errors at each level, a concatenated scheme with outer Reed-Solomon and inner convolutional codes was considered. The errors of the inner convolutional decoders are gathered in bursts, and the idea is that the inherent burst error correcting capability of the outer RS code will correct these errors.

II. The used system

Our system transmits signals over the AWGN channel. The used signal constellation is 8-PSK. This implies three levels in the system. Since the partition chain is 8-PSK/4-PSK/2-PSK/1-PSK, the minimum squared Euclidean distance among the signal points in the subsets at the different levels increases for each partition. Therefore the encoder of level 1 must be protected by a more powerful code than that of level 2, *et cetera*, i.e., $R_1 < R_2 < R_3$ (R_i is the rate of level i). The simulations showed that there were no need for a concatenated component code at level 3, only a convolutional code was used. In order to retain as high overall rate as possible, the rate of the inner code of level two must be quite large.

III. Results

For the inner code of level 2, at first a high rate punctured convolutional code was chosen due to the simple implementation of the decoder. Simulations then showed that bit error rate (BER) performance of this level bounded the overall code BER. This is caused by the bad extended row distance profile of punctured codes, i.e., error vectors **e** of small weight is enough to result in quite long bursts. As an alternative, a PUM code was tested. There exist decoding procedures for these codes [2] that are not more complex than decoding of punctured codes. The simulations showed a small improvement with this system. Introducing block interleaving between inner and outer codes resulted in approximately 0.9 dB gain in SNR. One idea to decrease error propagation is to interleave the transmitted symbols from the different levels. This channel interleaving would split a burst from a high level into several shorter, such that subsequent decoders see a channel which is bursty for several short periods rather than one long period. Comparing simulations of this system with simulations of a theoretical system without any error propagation at all (a genie between every level), showed a difference of less than 0.05 dB already at a BER of 10^{-4}. This implies that there is no need for channel interleaving as a way of decreasing error propagation. Simulations with channel interleaving supported this assumption. Finally we studied the influence of iterative decoding at the different levels. The hard output of the RS decoders complicates such schemes. There is no immediate way of extracting error probability of individual bits, needed for MAP-decoding, to be used with the hard estimates transmitted to subsequent decoding stages. It turned out that only level 1 benefitted from iterative decoding. Since the rate of this level is low, the total BER was not changed more than a few tenth of a dB.



References

[1] H. Imai, S. Hirakawa, "A new multilevel coding method using error correcting codes", *IEEE Trans. on Inform. Theory*, Vol. IT-23, pp. 371-376, May 1977.

[2] V. V. Zyablov, V. R. Sidorenko "Soft-decision Maximum-Likelihood Decoding of Partial-Unit-Memory Codes", *Prob. Peredachi Inform.* (English transl.), Vol. 28, No. 1, pp. 22-27, Jan.-March 1992.

[1] This work was supported in part by the Swedish Research Council for Engineering Sciences under Grants 92-661 and 94-83.

On Viterbi decoding of High Rate Convolutional Codes on Partial Response Channels

M. Fjelltveit and Øyvind Ytrehus

UCSD, Dept. of CMRR, 9500 Gilman Drive, La Jolla, CA 92093-0401, USA, email: fjellt@ucsd.edu
University of Bergen, Dept. of Informatics, HiB, N-5020 Bergen, Norway, email: oyvind@ii.uib.no

Abstract — **A new decoding technique is applied to a class of PUM codes on a 1-D PRC. Every bundle of parallel branches occurs more than once in the trellises representing these codes. This property is exploited to decrease the decoding complexity**[1]**.**

I. Introduction

The decoding of high rate convolutional codes on a 1-D Partial Response Channel (PRC) using a new proposed decoding technique presented in [1], is considered. The complexity of the decoding is characterized by the number of operations needed per decoded information bit, and by the size of the path memory. The path memory size is reduced by using Partial Unit Memory (PUM) codes. By exploiting the linearity of the parallel transitions in the trellises representing such codes, the number of decoding operations are reduced.

II. New proposed decoding technique from [1]

Consider the decoder trellis of an $[n,k,d]$ PUM code. Because $k > \nu$, there are parallel branches between pairs of states in the trellis which correspond to cosets of a block code with length n and dimension $k - \nu$. The block code is defined by the labels on the branches starting and ending in the zero state. The ACS step of the Viterbi algorithm is split into two steps:

1. Decode all parallel branches by a local Viterbi decoding. Identify in each step the surviving branch.
2. Decode the reduced trellis consisting of the surviving branches from step 1.

The block code and the cosets can be represented by a trellis, as shown for an [8,2,5] block code in Figure 2, and decoded by the Viterbi algorithm. Let $\{\delta_1, \ldots, \delta_{k'}\}$ denote the levels where this trellis merge. The number of operations in step 1 is $\#comparisons = \sum_{i=1}^{k'} 2^{\delta_i - i}$ and $\#additions = 2^{n-k'+1} - 2 + 2 \cdot \#comparisons$ where $k' = k - \nu$. It has been shown in [1] that this bound is attained if the block code satisfies the chain condition.

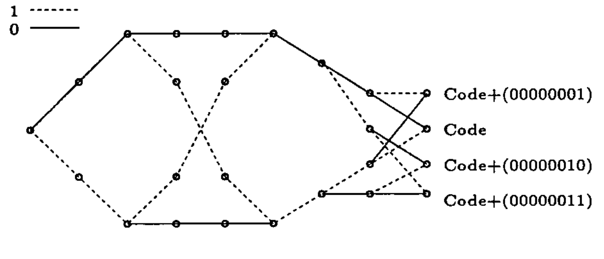


Figure 2: Trellis of an [8,2,5] block code and three cosets.

III. The new technique applied to PRCs

In [2] a system for a PRC is described. The binary information is encoded by an error-correcting code; a coset is used to generate the input to a precoder whose output is passed over a 1-D channel. The channel has input from {0,1} and output from $\{0, \pm 1\}$. The information of the PRC states is included into the decoder trellis by duplicating the states and giving them polarity. This reduces the number of parallel branches between pairs of states in the trellis.

The trellis in Figure 3 represents the block code and the cosets determined by the trellis of a $[5,3,d_{free}^2 = 6]$ code, $\nu = 1$, on a PRC. Notice that the block code and all the cosets start in positive states. The trellis for the cosets starting in negative states, has identical structure but with reversed signs.

The number of operations needed to decode this block code and the cosets, is 2·(4 comp + 26 add). The total number of operations per decoded bit for the PUM code is $(30 + 28)/2 \approx 30$ operations. A comparable punctured code uses 56 operations per decoded bit. The parity check matrices of the codes determined in [2] has a structure that do not allow every coset of the block code to be present in the convolutional code trellis. Every coset therefore occurs more than once in the trellis, which is the reason why the new technique for these codes on PRCs is an attractive alternative to punctured solutions.

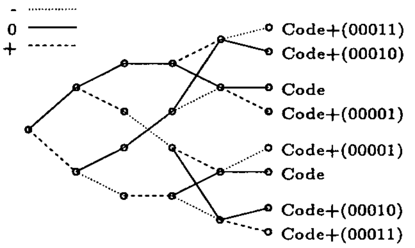


Figure 3: Trellis of a [5,2,6] block code and cosets.

References

[1] M. Fjelltveit and Ø. Ytrehus, "Reduced path memory Viterbi decoding of high rate convolutinal codes," presented at *1994 IEEE International Symposium on Information Theory*, submitted to *IEEE Trans. on Inform. Theory*, Aug. 1994.

[2] K. J. Hole and Ø. Ytrehus, "Improved coding techniques for precoded partial-respons channels,", *IEEE Trans. on Inform. Theory*, vol. 40, pp. 482-493, March 1994.

[1] This work was supported by the Norwegian Research Council (NFR).

Convolutional Codes for Precoded Partial-Response Channels: A Review

Kjell Jørgen Hole and Øyvind Ytrehus

University of Bergen, Department of Informatics, HiB, N-5020 Bergen, Norway.

E-mail: Kjell.Hole@ii.uib.no, Oyvind.Ytrehus@ii.uib.no. Partially supported by the Norwegian Research Council (NFR).

Abstract — **Cosets of convolutional codes may be used to generate zero-run length limited trellis codes for a precoded 1-D partial-response channel. Results on the zero-run length and free Euclidean distance of the trellis codes are reviewed.**

I. Cosets of Convolutional Codes

A rate $R = k/n$ convolutional code, C, may be defined by an $n-k$ by n polynomial parity-check matrix $\mathbf{H}(D)$. The i-th *input constraint length*, $\nu(i)$, of $\mathbf{H}(D)$ is equal to the maximum degree of the polynomials in the i-th row. The *(overall) constraint length* is given by $\nu = \sum \nu(i)$. As an example, a rate $R = 3/5$ convolutional code is defined by the parity-check matrix

$$\mathbf{H}(D) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ D & D+1 & 0 & 1 & D \end{pmatrix}.$$

Here $\nu(1) = 0$, $\nu(2) = 1$ and $\nu = 1$. We say that the parity-check matrix is *ordered* since $\nu(i) \leq \nu(i+1), i = 1, \ldots, n-k-1$. There are infinitely many parity-check matrices for a given convolutional code. We assume that a parity-check matrix with minimal constraint length ν is used. A *coset*, $C+\mathbf{a}$, of a convolutional code C is obtained by adding a fixed sequence, $\mathbf{a} \notin C$, to each codeword in C. The *maximum zero-run length*, L, of a coset $C+\mathbf{a}$ is the maximum number of consecutive zeroes in any sequence contained in $C+\mathbf{a}$.

II. Convolutionally Coded 1-D Channel

A coset of a convolutional code $C+\mathbf{a}$ may be used for error control and symbol synchronization in a precoded partial-response channel with transfer polynomial 1-D [1]. A binary sequence in $C+\mathbf{a}$ is sent through a *channel precoder* of characteristic $1/(1 \oplus D)$ and subsequently through the 1-D partial-response channel. The precoder essentially inverts the channel transfer function. Thus "0"s in the precoder input correspond to "0"s in the channel output, while "1"s in the precoder input correspond to "± 1"s in the channel output, where the signs alternate.

The set of noiseless ternary ($\{-1, 0, 1\}$) sequences generated by a coset of a convolutional code, the precoder, and the 1-D channel constitutes a *non-linear* trellis code. The maximum zero-run length of this trellis code is determined by the maximum zero-run length of the coset of the convolutional code. A good trellis code must have: (i) high rate R ($> 1/2$), (ii) short maximum zero-run length L, and (iii) large free squared Euclidean distance, d^2_{free}.

III. Bounds on Zero-Run Length

Theorem 1 *[2] Let C be any rate $R = k/n$ convolutional code with ordered parity-check matrix $\mathbf{H}(D)$. Then the maximum zero-run length $L \geq n\nu(1)$ for any coset $C+\mathbf{a}$.*

Corollary 1 *Let C be a rate $R = (n-1)/n$ convolutional code whose parity-check matrix has constraint length ν. Then $L \geq n\nu$ for any coset $C+\mathbf{a}$.*

From Corollary 1, any $R = (n-1)/n$ coset with large constraint length and/or rate has large maximum zero-run length.

Theorem 2 *[2] Let $\mathbf{H}(D)$ be an ordered parity-check matrix defining a rate $R = k/n$ convolutional code C. There exists a coset $C+\mathbf{a}$ with $L \leq n\nu(1) + 2n - 2 - l - t$, where $0 \leq l, t < n$ are determined from $\mathbf{H}(D)$.*

Corollary 2 *Let $\mathbf{H}(D)$ be an ordered parity-check matrix with $\nu(1) = 0$ defining a rate $R = k/n$ convolutional code C for $k \leq n-2$. Then there exists a coset $C+\mathbf{a}$ with $L \leq 2n-2-l-t$.*

Since the upper bound in Corollary 2 is independent of ν, there exist $R = k/n$, $k \leq n-2$, cosets with short maximum zero-run length for any constraint length.

IV. Good Trellis Codes for the 1-D Channel

Consider a convolutional code C with ordered parity-check matrix $\mathbf{H}(D)$ and free Hamming distance d_H. Theoretical results [1] and computer searches strongly indicate that if $\nu(1) > 0$ then $d^2_{free} = 2\lceil d_H/2 \rceil$ for any coset of C, else there may exist a coset with larger d^2_{free}.

Definition *Let $\mathcal{A}^*$ be the class of 2 by n parity-check matrices with $\nu(1) = 0$ and $\nu(2) = \nu > 0$.*

Let C be a rate $R = (n-2)/n$ code with parity-check matrix in $\mathcal{A}^*$. Then there exists a coset of C that generates a trellis code with $L \leq 2n - 2 - l - t$ and $d^2_{free} \geq 2\lceil d_H/2 \rceil$. A search technique for determining trellis codes with $d^2_{free} > 2\lceil d_H/2 \rceil$ is described in [1], [3]. The parameters of some good trellis codes defined by parity-check matrices in $\mathcal{A}^*$ are listed in the Table. More codes may be found in [1], [3].

TABLE : Good trellis codes for the 1-D channel.

R	states	d^2_{free}	L
3/5	16	8	5
3/5	64	10	5
4/6	8	6	7
4/6	32	8	6
5/7	16	6	7
7/9	8	4	8

References

[1] K. J. Hole and Ø. Ytrehus, "Improved coding techniques for precoded partial-response channels," *IEEE Trans. Info. Theory*, vol. 40, pp. 482-493, March 1994.

[2] K. J. Hole, "Cosets of convolutional codes for symbol synchronization and error control," Dep. of Informatics, Univ. of Bergen, Tech. Rep. no. 64, June 1992.

[3] K. J. Hole and Ø. Ytrehus, "Trellis codes for precoded 1-D partial-response channels: further improved search techniques," in *Proc. ISITA '94*, Sydney, Australia, 20-24 Nov. 1994.

Probabilistic Dependence and Information Theory

James L. Massey

Signal & Info. Proc. Lab., Swiss Federal Inst. Tech., CH-8092 Zurich, Switzerland

It is argued that mutual information is essentially a measure of probabilistic dependence and that information theory provides a convenient calculus for reasoning about probabilistic dependence. Because $I(X;Y) \geq 0$ with equality if and only if the random variables X and Y are independent, it follows that the determination of whether X and Y are independent reduces to computing the single real number $I(X;Y)$. Moreover, the vanishing of $I(X;Y)$ can alternatively be taken as the definition of probabilistic independence. Similarly, the vanishing of the conditional mutual information $I(X;Y/Z)$ can be taken as the definition of the independence of X and Y when conditioned on (knowledge of) Z. Independence and conditional independence are in general unrelated properties of random variables; X and Y can be independent but not independent when conditioned on Z and, conversely, X and Y can be dependent but independent when conditioned on Z. Conditional independence is shown to play an important role in the calculus of probabilistic dependence. It is shown that a *Markov chain* can be defined as a sequence $X_1, X_2, ...X_n$ of random variables such that X_i and X_k are independent when conditioned on X_j for all $1 \leq i < j < k \leq n$. An immediate consequence of the symmetry $I(X;Y/Z) = I(Y;X/Z)$ is that the reversed sequence $X_n, X_{n-1}, ...X_1$ is also a Markov chain. Similarly, it is shown that a *sufficient statistic* can be defined in the manner that Z is a sufficient statistic for any decision about, or estimate of, X from the pair (Y, Z) just when X and Y are independent when conditioned on Z. This interpretation provides insight into (generalizations of) the *theorem of irrelevance* and the *theorem of the magic genie*, which are familiar to all readers of the classic textbook of Wozencraft and Jacobs [1]. The real utility of information theory for analyzing probabilistic dependence becomes evident when considering networks of information sources, channels, encoders and decoders. Precise definitions of all these devices are given together with the rules for their interconnection in neworks. Principles for deducing probabilistic dependencies, or the lack thereof, in such networks are formulated. The distinction between causal dependence and probabilistic dependence is seen to be crucial to this formulation. The practical utility of the above concepts is illustrated by several examples chosen from cryptology.

References

[1] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: Wiley,1965.

Practical Save-up Strategies

Phons Bloemen

Information and Communication Theory Group, Eindhoven University of Technology

Abstract — **In 1982, Schalkwijk designed coding strategies for two-way channels by subdivision of a unit square. This method was used to construct discrete coding strategies and save-up strategies for the BMC. This paper concerns the implementation of these strategies.**

I. Introduction to coding strategies

Schalkwijk [2] first introduced the method of progressively subdividing a unit square to present a coding strategy for the BMC that outperforms Shannon's inner bound region. The binary multiplying channel (BMC), is a TWC which has two binary inputs X_1, X_2 and a common binary output $Y = X_1 \cdot X_2$. Schalkwijk defined his strategy on the 1×1 unit square, representing the messages of each terminal by intervals on $[0, 1)$.

Discrete coding strategies on $M \times M$ squares for the BMC, where the messages are taken from a finite set of M messages were constructed using the same method. A discrete coding strategy subdivides the $M \times M$ square up to basic 1×1 squares: all information contained in a message is transmitted.

Improvements to the rate of discrete coding strategies were obtained by *save-up* strategies in which the $M \times M$ square is subdivided into rectangular areas. Not all information contained in a message is transmitted anymore: a remainder is *saved up*. Tables of various coding strategies are found in [1].

II. Discrete strategies

Discrete strategies can be easily implemented using a finite state machine at each terminal $i, i = 1, 2$, to perform the encoding/decoding. The input message m_i and the channel output symbols $y_1, y_2, \ldots, y_k$ are used as input symbols for the finite state machine. It generates the next channel input symbol $x_{i,k+1}$, and finally the received message m_{3-i}.

In the figure, the leftmost picture shows the channel output symbol sequences of a 4×4 discrete coding strategy. Not all possible channel output symbol sequences are used; in general, about $M^2/2$ different sequences show up in a $M \times M$ strategy. The decoder must have knowledge about at least one of the transmitted messages, to distinguish between message pairs. Cryptographic application of this feature was discussed in [3].

III. Save-up strategies

Save-up strategies subdivide the $M \times M$ square up to rectangular sets of message pairs. In a rectangular resolution product, the input messages of the two terminals are statistically independent. Depending on the size of the rectangle a message pair is situated in, some information bits still must be received, and some information bits still must be sent. These untransmitted information bits are *saved up*. When both terminals have saved up $\log M$ bits of information, they encode them in a new message, and transmit it using the $M \times M$ strategy. The middle picture shows the 4×4 *save-up* strategy. *Save-up* occurs in the two 2×1 areas in the upper right/lower left corner, and in the lower right 3×3 square.

When implementing *save-up* strategies in the way described above, practical problems arise. Memory is needed to keep track of the save-up information, and to buffer incomplete messages. Also, the *save-up* process introduces *decoding delay*: it may take several uses of the strategy before the complete message is received. Third, buffer overflow may occur if save-up takes place in only one terminal. Finally, the encoding of the saved-up pieces of information may not be performed without *encoding losses*: in the 4×4 strategy, it is possible that a single bit (one of the '00' pairs is selected) or a trit (log 3 bits) is saved up. Conversion of trits to bits by a code with a finite block length is not possible without losses.

IV. Two-power strategies

To address the problems described above, a modification of the save-up strategy is used. In the so-called *2-power* strategy, only those $N_1 \times N_2$ areas are *saved-up* where N_1 and N_2 are powers of 2. This means that saved-up information can be easily converted into bits. The table lists rates $R'(M \times M)$, in bit per direction per transmission, of some *2-power* strategies. The right part of the figure shows the 4×4 *2-power* strategy.

Using a strategy where M is a power of 2 simplifies encoder/decoder design drastically. The strategy encoder takes $\log M$ bits from the input message stream to encode the first message for the strategy. When save-up occurs, some bits are left untransmitted. These bits, and enough new bits from the input message stream together are used to encode the next message for the strategy. Less memory is needed to administer the save-up process, no decoding delay or encoding loss occurs. Compared to the rate of a *save-up* strategy, a small price must be paid.

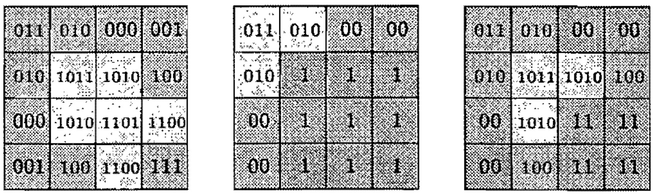
References

[1] Hendrik B. Meeuwissen and Alphons H.A. Bloemen. Practical two-way communication coding strategies. In *Proc. Joint Swedish-Russian International Workshop on Information Theory*, volume 6, pages 92–96, Mölle, Sweden, 1993.

[2] J. Pieter M. Schalkwijk. The binary multiplying channel - a coding scheme that operates beyond the Shannon inner bound. *IEEE Transactions on Information Theory*, IT-28(1):107–110, Jan 1982.

[3] Bernard J.M Smeets. *Topics in Cryptography*. PhD thesis, Dep of Computer Engineering, University of Lund, Sweden, Sep 1985. Teknisk-Licentiat-Thesis.

M	$\#'(M \times M)$	Dpt	$T(M \times M)$	$R'(M \times M)$
2	7	2	8.00	0.5714286
4	43	3	52.00	0.6046512
8	254	5	312.00	0.6141732
16	1417	7	1756.00	0.6196189
32	5149	9	6416.00	0.6230336



011	010	000	001
010	1011	1010	100
000	1010	1101	1100
001	100	1100	111

Discrete R = 0.59259

011	010	00	00
010	1	1	1
00	1	1	1
00	1	1	1

Saveup R = 0.60521

011	010	00	00
010	1011	1010	100
00	1010	11	11
00	100	11	11

2-power R = 0.60465

Further Results on the Non-Cooperative Binary Adder Channel

Suzanne Hjelm

Department of Electrical Engineering, Linköping University, S-581 83 Linköping

Abstract — **We present a table of new ideal linear anti-jamming codes for the non-cooperative binary adder channel. For given parameters the codes are optimal in the sense that they have the lowest possible erasure probability. We also present upper and lower bounds on the erasure probability for any given linear anti-jamming code.**

I. Introduction

We study a channel where binary information is transmitted. On this channel there is intentional interference caused by a jammer. The jammer is supposed to be intelligent. This means that within the frame of certain constraints he adjustes his activities so as to cause the worst possible disturbance on the legal transmission. The limitations on the jammer are given by the constraint that also the jammer is supposed to transmit binary messages.

The channel model used is the binary adder channel. Given binary inputs the output is given as the Euclidean sum, see [1].

II. Code Construction

Let $\mathcal{M}_z$ be a given set. A code consists of a family $\{\mathcal{C}(z); z \in \mathcal{M}_z\}$ of codes which the legal user can alternate between. For each transmitted message one code is pointed out by a key $z \in \mathcal{M}_z$ which is known by both the encoder and the decoder but unknown to the jammer.

Let $F \triangleq \{0,1\}$ and let $\oplus$ denote the usual binary addition. Let $\mathcal{C}_0$ be a linear code in F^n and let $\mathcal{M}_z$ be an arbitrary set in F^n. We define subcodes $\mathcal{C}(z)$ as

$$\mathcal{C}(z) \triangleq \mathcal{C}_0 \oplus z; z \in \mathcal{M}_z.$$

The resulting total code is a linear anti-jamming code if

$$\mathcal{C} \triangleq \bigcup_{z \in \mathcal{M}_z} \mathcal{C}(z)$$

is a linear code, see [1].

III. The erasure probability

Assuming equally probable subcodes the erasure probability is given by

$$\Gamma(s_0) \triangleq \max_{s \in F^n} \Gamma(s) \triangleq \max_{s \in F^n} \frac{1}{\mid \mathcal{C} \mid} \sum_{x \in \mathcal{C}} \gamma(x \oplus s)$$

where $\gamma(\cdot)$ indicates whether a codeword can be decoded with or without ambiguity given the jamming vector s, see [1]. According to the same reference it is enough to search through all $s \in \epsilon$ where ϵ is the set of all binary polynomials, of degree less than or equal to the degree of $g(x)$, represented as sequences of length n. We propose the following two bounds:

Proposition 1 *Let $w(x \oplus s)$ denote the Hamming weight of $x \oplus s$ and let d_{min} be the minimum distance within each subcode. An upper bound for $\Gamma(s_0)$ is given by*

$$\max_{s \in F^n} \Gamma(s) \leq \max_{s \in \epsilon} \mid \{x \in \mathcal{C} : w(x \oplus s) \geq d_{min}\} \mid \cdot \frac{1}{\mid \mathcal{C} \mid}.$$

Proposition 2 *Let $w(x)$ denote the Hamming weight of x and let k_0 be the dimension of $\mathcal{C}_0$. Then a lower bound for $\Gamma(s_0)$ is given by the maximum of*

$$\max_{s \in F^n} \Gamma(s) \geq \max_{s \in \epsilon \setminus 0^n} (\mid \{x \in \mathcal{C} : w(x \oplus s) \geq n - k_0 + 1\} \mid) \cdot \frac{1}{\mid \mathcal{C} \mid}$$

and

$$\max_{s \in F^n} \Gamma(s) \geq (\mid \{x \in \mathcal{C} \setminus \mathcal{C}_0 : w(x) \geq n - k_0 + 1\} \mid + M_x - 1) \cdot \frac{1}{\mid \mathcal{C} \mid}.$$

IV. The ideal codes

Let $1 \oplus x^n = a(x)b(x)g(x)$ and let $g_o(x) = a(x)g(x)$. Then the code $\mathcal{C}$ is generated by $g(x)$ and the subcode $\mathcal{C}_0$ of $\mathcal{C}$ generated by $g_0(x)$. We present a list of ideal binary cyclic anti-jamming codes of lengths between 15 and 23 with an information-rate less than the capacity. For describing the codes we use the notation from [1] specifying $a(x), b(x)$ and $g(x)$. The polynomials are represented by the integers A, B and G obtained by changing $\oplus$ to $+$ and inserting $x = 2$. In this case we have the information-rate R_x

$$R_x = \frac{1}{n} \deg b(x)$$

and the key-rate R_z as

$$R_z = \frac{1}{n} \deg a(x).$$

The erasure probability of an ideal code satisfies

$$P_E = \frac{M_x - 1}{M_x \cdot M_z}$$

where $M_x = \mid \mathcal{C}_0 \mid$ and $M_z = \mid \mathcal{M}_z \mid$, see [1].

A	B	G	R_x	R_z	$-\log P_E$
11,13	7	3,87,117	2/21	6/21	1.931
13,87	7	3,11,117	2/21	9/21	2.834
7,87	13	3,11,117	3/21	8/21	2.466
13	7	3,11,87,117	2/21	3/21	1.028
7	13	3,11,87,117	3/21	2/21	$6.601 \cdot 10^{-1}$
87	7,	3,11,13,117	2/21	6/21	1.931
87	13	3,7,11,117	3/21	6/21	1.864
87	11	3,7,13,117	3/21	6/21	1.864
7	3,13	11,87,117	4/21	2/21	$6.301 \cdot 10^{-1}$
7	19	3,25,31	4/15	2/15	$6.301 \cdot 10^{-1}$
7	31	3,19,25	4/15	2/15	$6.301 \cdot 10^{-1}$
7	3,19	25,31	5/15	2/15	$6.158 \cdot 10^{-1}$

Acknowledgements

The author would like to thank Prof. Thomas Ericson and Prof. Stefan Dodunekov for encouragement and helpful comments.

References

[1] T. Ericson, "The Noncooperative Binary Adder Channel" *Transactions on Information Theory*, vol. 32, pp. 365–374, 1986.

Communication Complexity of the Hamming Distance

Ulrich Tamm

Department of Mathematics, University of Bielefeld, P.O. Box 100131, 33501 Bielefeld, Germany

The communication complexity $C(f)$ is the number of bits that have to be exchanged between two persons P_1 and P_2 in order to enable them to evaluate a function $f(x,y)$ when initially each person knows only one of the arguments.

A lower bound is obtained via the rank of the matrices $M_k(f) = (a_{xy})_{x,y}$ defined by

$$a_{xy} = \begin{cases} 1 \text{ if } f(x,y) = k \\ 0 \text{ if } f(x,y) \neq k \end{cases},$$

namely

$$C(f) \geq \lceil \log_2 \sum_k rank M_k(f) \rceil. \tag{1}$$

Often this lower bound is close to the upper bound obtained from the protocol in which P_1 transmits all the bits of x enabling P_2 to determine $f(x,y)$ which he returns. For the Hamming distance d_n (over an alphabet of size q this yields the upper bound $C(d_n) \leq \lceil n \cdot log_2(q) \rceil + \lceil log_2(n+1) \rceil$.

The communication complexity of the Hamming distance was first considered by El Gamal and Pang [2]. They determined $C(d_n)$ up to one bit if $q = 2$. This result was later extended by Ahlswede [1] to alphabet sizes $q = 4, 5$. So for $q = 2, 4, 5$ and all $n \geq 1$

$$\mid C(d_n) - \lceil n \cdot \log_2(q) \rceil - \lceil \log_2(n+1) \rceil \mid \leq 1. \tag{2}$$

In order to prove (2) a lower bound using constant distance code pairs was applied. In [4] the rank lower bound (1) was used to prove that (2) holds for all q and the special parameters $n = p^m - 1, m \geq 1$, where p is a prime factor of q. The matrices $\{M_k(d_n)\}_{k=0}^n$ of the Hamming distance just form the Hamming association scheme. The eigenvalues of $M_k(d_n)$ are the Krawtchouk polynomials $K_k(x,q,n) = \sum_{j=0}^n \binom{x}{j}\binom{n-x}{k-j}(-1)^j(q-1)^{k-j}$ evaluated at the integers $x = 0, \ldots, n$. If all these eigenvalues are different from 0, i. e., the Krawtchouk polynomials do not have integral zeroes, then by (1) the statement (2) is immediate. The proof in [4] makes use of number theoretic arguments. Another approach was done in [3] using the observation that two consecutive integral zeroes of $K_k(x,2,n)$, $k \neq \frac{n}{2}$ have difference greater than 2. This allows to give a new proof of statement (2) for alphabet size $q = 2$ by application of the rank lower bound.

In [1] also the communication complexity of the Hamming distance (modulo 2) was exactly determined for alphabet sizes $q = 2, 4$. With the rank lower bound (1), in [4] this result was extended to all positive integers q making use of the fact that for $z = 0, 1$

$$\sum_{k \equiv z (\text{mod } 2)} K_k(i,q,n) \tag{3}$$

$$= \begin{cases} \frac{1}{2}(q^n + (-1)^z(2-q)^n) & i = 0 \\ (-1)^z 2^{n-w-1}(2-q)^w & i = 1, \ldots, n \end{cases} \tag{4}$$

where w is the number of 1's in the binary representation of i. From this follows that

$$C(d_n(mod2)) = \begin{cases} 2 \text{ for } q = 2 \\ \lceil n \cdot \log_2(q) \rceil + 1 \text{ for } q \geq 3 \end{cases} \tag{5}$$

Further, in [4] the communication complexity of the Hamming distance (modulo 3) was determined up to one bit exploiting recursion formulas for the Krawtchouk polynomials.

References

[1] Ahlswede, R., *On code pairs with specified Hamming distances*, in "Colloq. Math. Soc. Janos Bolyai", Vol. 52, pp. 9 - 47, North Holland, Amsterdam, 1989.

[2] El Gamal, A. and Pang, K. F., *Communication complexity of computing the Hamming distance*, SIAM J. Comput., Vol. 15, No. 4, pp. 932 - 947, 1986.

[3] Spieker, B., *Deterministic communication complexity of the Hamming distance*, Memorandum No. 1026, Faculty of Applied Mathematics, University of Twente, Enschede.

[4] Tamm, U., *Communication complexity of sum-type functions invariant under translation*, to appear in Information and Computation.

Constrained Sequences and Optimization

Thijs Veugen

Eindhoven University of Technology, Group on Information and Communication Theory, PO box 513, 5600 MB Eindhoven, The Netherlands

Abstract — **An optimization problem for constrained m-ary sequences is considered. It is shown that the solution to this generally difficult problem is easily found when a fractional generating polynomial for the constrained sequences has been derived.**

I. Notation

We consider m-ary sequences of arbitrary length that satisfy some constraints. Let v_i be a non-negative integer, $(i = 0 \dots m-1)$. Denote the number of constrained sequences consisting of precisely v_i symbols i $(0 \le i < m)$ by $M(v_0, v_1, \dots, v_{m-1})$. Let $Q = \{\underline{q} = (q_0, q_1, \dots, q_{m-1}) \mid q_i \ge 0, (i = 0 \dots m-1)$, and $\sum_i q_i = 1\}$ be the set of symbol distributions. Let l be a non-negative integer. Let $\underline{q} \in Q$. The notation $[\underline{q}l]$ is used to denote the vector $(l_0, l_1, \dots l_{m-1})$ such that l_i is the closest integer to $q_i \cdot l$, $(i = 0 \dots m-1)$.

II. Assumptions

We assume that the constraints on the m-ary sequences are such that

1. If $(v_0, v_1, \dots, v_{m-1})$ and $(w_0, w_1, \dots, w_{m-1})$ are vectors of non-negative integers such that $v_i \le w_i$ $(i = 0 \dots m-1)$, then $M(v_0, v_1, \dots, v_{m-1}) \le M(w_0, w_1, \dots, w_{m-1})$.
2. The function P defined by
$$P(\underline{q}) = \limsup_{l \to \infty} \frac{\log_m M([\underline{q}l])}{l}$$
is continuous on Q.

The function P can be interpreted as the average amount of information contained in a symbol of a constrained sequence.

III. Optimization

Let $C_i > 1$ be a constant $(i = 0 \dots m-1)$. The constant C_i can be interpreted as the cost of using symbol i. Our goal is to maximize the function R defined by

$$R(\underline{q}) = \frac{P(\underline{q})}{\sum_{0 \le i < m} q_i C_i}$$

over all $\underline{q} \in Q$. Since P is continuous, the maximum is attainable. The function R can be interpreted as the average amount of information per cost unit.

IV. Solution

In general it is hard to obtain an explicit expression for $P(\underline{q})$ in terms of $\underline{q}$, so straightforward maximization is not easy. Sometimes it is possible, by using recurrence relations, to derive the generating polynomial p defined by $p(x_0, x_1, \dots, x_{m-1}) = \sum_{v_0 \ge 0, \dots v_{m-1} \ge 0} M(v_0, \dots, v_{m-1}) x_0^{v_0} \dots x_{m-1}^{v_{m-1}}$ as a fraction of a nominator polynomial p_n, and a denominator polynomial p_d. Then the maximization problem can be solved by the following theorem.

Theorem 1 *Let $R \ge 0$.*

1. *If $R < \max_{\underline{q} \in Q} R(\underline{q})$, then $p(m^{-E_0 R}, \dots, m^{-E_{m-1} R}) = \infty$*
2. *If $p(m^{-E_0 R}, \dots, m^{-E_{m-1} R}) = \infty$, then there exists a $\underline{q} \in Q$ such that $R(\underline{q}) = R$*

The maximization problem is solved by numerically computing the solution $R \ge 0$ of $p_d(m^{-E_0 R}, \dots, m^{-E_{m-1} R}) = 0$ (provided $p_n(m^{-E_0 R}, \dots, m^{-E_{m-1} R}) \ne 0$).

V. Example

Suppose that there are no constraints imposed on the sequences. Then

$$M(v_0, v_1, \dots, v_{m-1}) = \binom{v_0 + v_1 + \dots + v_{m-1}}{v_0, v_1, \dots, v_{m-1}}$$

The function P is equal to the m-ary entropy function H. The generating polynomial p is equal to $1/(1 - (x_0 + x_1 + \dots + x_{m-1}))$. From Theorem 1 follows that the maximum value of $R(\underline{q})$ over all $\underline{q} \in Q$ is the solution $R > 0$ of $m^{-E_0 R} + m^{-E_1 R} + \dots + m^{-E_{m-1} R} = 1$.

This result can also be obtained by a Lagrange optimization of function R: $\mathcal{L}(\underline{q}, \lambda) = \lambda \cdot (\sum_{0 \le i < m} q_i - 1) + H(\underline{q}) / \sum_{0 \le i < m} q_i C_i$. The equation $\frac{\delta \mathcal{L}}{\delta q_i} = 0$ $(i \in \{0, \dots, m-1\})$ leads to $-\log_m q_i = 1 + E_i R(\underline{q}) - \lambda \sum_{0 \le j < m} q_j E_j$. By multiplying this equation with q_i and adding over all i, $0 \le i < m$, we obtain $\lambda = 1 / \sum_{0 \le j < m} q_j E_j$. Therefore the maximum value of $R(\underline{q})$ is the solution R of $m^{-E_0 R} + m^{-E_1 R} + \dots + m^{-E_{m-1} R} = 1$, obtained by $q_i = m^{-E_i R}$.

The above example shows that even in the case of unconstrained sequences, where the function P can be determined, the application of Theorem 1 saves some calculations. On the other hand, when one is interested in the optimal values of q_i, an immediate solution can not be obtained from Theorem 1.

VI. Application

The results described in this paper are used to show that multiple repetition strategies can achieve capacity [1]. The capacity achieving symbol distributions turn out to be easily computable.

References

[1] Thijs Veugen. Capacity achieving strategies for discrete memoryless channels with feedback. In *1994 IEEE International Symposium on Information Theory*, page 466, June 1994.

Multiterminal Source Coding Challenges

Toby Berger

School of Electrical Engineering, Cornell University, Ithaca, NY 14853 USA

Abstract — **Several fundamental problems in multiterminal source coding are identified and discussed. Among them are the CEO Problem (Berger-Zhang), for which we provide some exact asymptotics and the multiple descriptions problem for which we provide improved bounds. We also examine universal extensions of these and other multiterminal source coding problems.**

I. Introduction

Multiterminal source coding offers many analytical challenges. The CEO Problem, the multiple descriptions problem, the Slepian-Wolf problem with distortions and extensions of these and other problems to universal lossy coding, especially of the incremental parsing variety, are areas of active exploration. We shall provide an overview of recent progress and remaining challenges in this dynamic research area.

II. The CEO Problem

Here is a new problem in multiterminal source coding. A firm's Chief Executive Officer (CEO) is interested in the data sequence $\{X(t)\}_{t=1}^{\infty}$ which cannot be observed directly. The CEO deploys a team of L agents who observe independently corrupted versions of $\{X(t)\}_{t=1}^{\infty}$. Either because the CEO is extremely busy or because the agents must remain clandestine, the combined data rate at which the agents may communicate information about their observations to the CEO is limited to, say, R bps. Suppose that the agents are not permitted to convene, Agent i having to send data based solely on his own noisy observations, $\{Y_i(t)\}$. We show that then there does not exist a finite value of R for which even infinitely many agents can make D arbitrarily small. Furthermore, in this isolated-agents case we determine the asymptotic behavior of the minimal error frequency in the limit as L and then R tend to infinity.

III. The Multiple Descriptions Problem

Multiple description source coding concerns situations in which the transmission of the source information is distributed over two data streams at rates R_1 and R_2, respectively. When both data streams are received, the decoder uses the combined data at rate $R_1 + R_2$ to reconstruct the source information with average distortion d_0. If a communication breakdown prevents one of the data streams from reaching the receiver, the decoder has to base its reconstruction solely on the available data at rate either R_1 or R_2. This results in a higher distortion of either d_1 or d_2, respectively. The region $\mathcal{R}$ of all quintuples $(R_1, R_2, d_0, d_1, d_2)$ has been determined in the so-called 'no excess rate' case defined by imposing the requirement $R_1 + R_2 = R(d_0)$, where $R(\cdot)$ is the rate-distortion function of the source. The case with excess rate in which $R_1 + R_2 > R(d_0)$ is permitted seems difficult. In the special case of the excess rate problem in which it is required that $R_t = R(d_t), t = 1, 2.$ we obtain lower and upper bounds on d_0 separated by only a tiny gap when evaluated for a binary equiprobable source and the Hamming distortion measure.

IV. Slepian-Wolf with Distortion

This by now classical problems concerns two correlated sources, ""X_k"" and ""Y_k"" observed at separated terminals. We seek the region comprised of those pairs (R_x, R_y) of encoding rates that suffice to permit a recipient of both encoder outputs to recover the components of the source with respective distortions D_x and D_y. When $D_x = D_y = 0$ we have the Slepian-Wolf problem, when either $D_x = 0$ or $D_y = 0$, we have the Wyner-Ziv problem. The case in which both $D_x > 0$ and $D_y > 0$ remains open. We discuss some ideas for improving bounds on the rate region, especially for small D_x and D_y.

V. Universal Extensions

All the above problems assume a priori parametric knowledge of the joint distribution of all the source data. Since such knowledge usually is unavailable in practice, it is of considerable interest to appemtp to extend the above results to universal contexts in which only the source alphabet and (perhaps) the distortion measure are known, but not the source statistics. In particular, we seek lossy extensions of lossless incremental parsing algorithms of the Lempel-Ziv variety to multiterminal situations; the challenges here are many and the results to date are scant.

Elias Omega Code and Log-Star Function

R. Ahlswede, T. S. Han and K. Kobayashi
Bielefeld Universität, Fakultät Mathematik, POB 100131, 33501 Bielefeld, Germany
The University of Electro–Communications, Graduate School of Information Systems and,
Department of Computer Science and Information Mathematics, Chofugaoka 1-5-1, Chofu, Tokyo, 182, JAPAN

Abstract — **In this talk we consider the asymptotically optimal universal prefix code on the set of positive integers $\mathcal{N}^+$, especially concentrate on the Elias omega code, and if time permits, a code induced by Bentley-Yao unbounded search tree.**

1. Introduction and Notations

The efficient representation of numbers is important in the computer science, and can be used for the data compression. In order to study the prefix code on $\mathcal{N}^+$, we first introduce notations for representing binary sequences (if necessary, we extend the notations to r-ary sequence in trivial manner). We denote the standard binary expression of positive integer $j \in \mathcal{N}^+$ as $(j)_2$, the most significant bit(MSB) of which is 1. For example, $(13)_2 = 1101$. Next we express the floor function of log by

$$\lambda_2(j) = \lfloor \log_2 j \rfloor. \qquad (1)$$

Moreover, λ_2^k is the k-hold composition of function λ_2.

2. Elias omega code

Elias[1] introduced a universal code $\omega : \mathcal{N}^+ \rightarrow \{0,1\}^*$, called the ω-code, described by

$$\omega(j) = \begin{cases} 0 & \text{for } j = 1 \\ (\lambda_2^{k-1}(j))_2 \cdots (\lambda_2(j))_2 (j)_2 0 & \text{for } j \geq 2 \end{cases} \qquad (2)$$

where $k = k(j)$ is the positive integer satisfying $\lambda_2^k(j) = 1$ (which exists for any $j \geq 2$). Then the codeword length of this prefix code ω is given by

$$c_E(j) = |\omega(j)| = \sum_{i \geq 1 : \lambda_2^i(j) \geq 0} (\lambda_2^i(j) + 1) \quad (j = 1, 2, \ldots). \qquad (3)$$

3. Bounds for the codeword length function of code ω

In order to introduce the bound for $c_E(j)$, we define the **log–star** function $\log_2^*(x)$ for $x \geq 1$ as

$$\log_2^*(x) \equiv \log_2(x) + \log_2 \log_2(x) + \cdots + \log_2^{w^*(x)}(x) \qquad (4)$$

where $\log_2^w(x)$ is the k-hold composition of the function $\log_2(x)$, and $w^*(x)$ is the largest positive integer satisfying $\log_2^w(x) \geq 0$. Therefore, $w^*(x) = 1, \log_2^*(x) = 0$ for $x = 1$.

Then we established upper and lower bounds for the length function $c_E(j)$.

□ **Theorem 1** *For any real $x \geq 1$,*

$$\log_2^*(x) < c_E(x) \leq \log_2^*(x) + w^*(x). \qquad (5)$$

Here we have extended the domain of function $c_E(\cdot)$ to the set of real numbers through the extension of λ_2. Through a simple consideration, we can check that the upper bound is attained at the points $j_m = \exp_2^m(1)$ $(m = 0, 1, \ldots)$, where $\exp_2(x) = 2^x$ and $\exp_2^k(x)$ is the k-hold composition of function $\exp_2(\cdot)$. Moreover, the lower bound is also attained at the same points in the meaning of

$$\lim_{x \uparrow j_m} c_E(x) = \log_2^*(j_m). \qquad (6)$$

Therefore, the two bounds are best possible as far as we restrict the bounding functions to such smooth functions.

Furthermore, we remark that the unbounded search tree on $\mathcal{N}^+$ induced by the Elias omega code has a more beautiful recursive structure than Bentley-Yao search tree[2].

4. Modified log–star function

Due to the finiteness of the sum of $2^{-\log_2^*(j)}$, that is,

$$\sum_{j=1}^{\infty} 2^{-\log_2^*(j)} < +\infty, \qquad (7)$$

we can construct a prefix Shannon code with the length function satisfying

$$c_0(j) = \lceil c^* + \log_2^*(j) \rceil \qquad (8)$$

for a normalizing constant $c^* = 1.5185\ldots$. This code has better performance than Elias omega code in larger integers. Then, is this the best prefix code on $\mathcal{N}^+$? Next lemma gives an answer to this question. Before describing the lemma, we define the **modified log–star** function by

$$\log_{r,\alpha}^*(x) = \log_r^*(x) - \alpha w_r^*(x) \quad (x \geq 1) \qquad (9)$$

for integer $r \geq 2$ and real number α.

□ **Lemma 1** *For integer $r \geq 2$, set $\alpha_r^* = \log_r(\log_r e)$.*

1) If $\alpha < \alpha_r^*$, then

$$\sum_{j=1}^{\infty} r^{-\log_{r,\alpha}^*(j)} < +\infty, \qquad (10)$$

2) If $\alpha \geq \alpha_r^*$, then

$$\sum_{j=1}^{\infty} r^{-\log_{r,\alpha}^*(j)} = +\infty. \qquad (11)$$

We will discuss on the consequences from the lemma, and the topics about another kind of asymptotically optimal universal prefix codes.

References

[1] Elias, P. , "Universal codword sets and representation of the integers," *IEEE Trans. on Information Theory*, vol.IT-21, pp.194–203, 1975.

[2] Bentley, J. L. and Yao, A. C. , "An almost optimal algorithm for unbounded searching," *Information Processing Letters*, vol.5, no.3, pp.82–87, 1976.

Sporadic Information Sources

Urs Loher
Signal and Information Processing Laboratory
Swiss Federal Institute of Technology
CH-8092 Zürich, Switzerland

Message arrivals encountered in digital transmission over most real communication channels are not independent but appear in clusters. Sources forming bursty (or clustered) message arrivals are said to exhibit memory, i.e., statistical dependence in the occurrence of message symbols, and thus cannot be adequately represented by a classical memoryless symmetric source. A model of a bursty K−ary source using a Markov chain with two states "quiet" (or "idle") and "busy" (sometimes also called "active") is proposed. In the "quiet" state, the source transmits no (message) information, while in the "active" state, the source acts as a $(K-1)$−ary discrete memoryless source (DMS). The clustered arrivals of bits can be interpreted as a characteristic of the source *or* of the channel (DMS concatenated with a two-state Markov channel). Basic limitations on the amount of protocol information that must be transmitted over a link in a communication network to keep track of intramessage information (e.g., message lengths) are considered. Different strategies are developed to reduce this information drastically. Certain generalizations of the concept of sporadic sources are devised for some related applications.

Asymptotically Optimal Constructions for Covering Codes

René Struik

Eindhoven University of Technology, Department of Mathematics and Computing Science, P.O. Box 513, 5600 MB
Eindhoven, the Netherlands, e-mail: dwrs@win.tue.nl

1. Introduction

We will show how a generalization of the direct sum construction can be used to construct some exceptionally good covering codes, i.e. codes that have few codewords, given their length and covering radius.

2. Notation

An $(n, M, d)r$ code denotes a (n, M, d) code with covering radius r. Let $\mathcal{C}$ be a binary code of length n and let $t \geq 0$. The density $\mu(\mathcal{C}, t)$ of $\mathcal{C}$ is defined as the average number of codewords that is at distance at most t from a word in the vector space $I\!F_2^n$, i.e. $\mu(\mathcal{C}, t) = 2^{-n}|\mathcal{C}| \cdot \sum_{i=0}^{t} \binom{n}{i}$.

3. The Direct Sum Construction; Generalizations

One of the easiest ways to combine two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ is simply to take their direct sum $\mathcal{D} := \mathcal{C}_1 \times \mathcal{C}_2$. This construction, though simple, generally yields codes with a poor minimum distance and a poor covering radius. Sometimes, however, one can show that code $\mathcal{D}$ contains a proper subcode with better distance properties than $\mathcal{D}$ itself and (almost) the same covering radius as code $\mathcal{D}$. These proper subcodes can be obtained via the following generalization of the direct sum construction:

Definition 1 *[1] Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be the union of the k subcodes $\mathcal{C}_1^{(1)}, \ldots, \mathcal{C}_1^{(k)}$, resp. $\mathcal{C}_2^{(1)}, \ldots, \mathcal{C}_2^{(k)}$. The blockwise direct sum (BDS) of codes $\mathcal{C}_1$ and $\mathcal{C}_2$ w.r.t. these subcodes is the code $\mathcal{D} := \cup\{\mathcal{C}_1^{(i)} \times \mathcal{C}_2^{(i)} \mid 1 \leq i \leq k\}$.*

The next theorem gives a bound on the minimum distance of the blockwise direct sum of two codes.

Theorem 2 *[1] Suppose that $\mathcal{C}_1$ has distance d_1 and that all its disjoint subcodes have distances at least d_{11}. Furthermore, suppose that the respective distances for code $\mathcal{C}_2$ and its disjoint subcodes are d_2 and d_{22}. Then the BDS of these two codes has distance $d \geq \min\{d_{11}, d_{22}, d_1 + d_2\}$.*

It is possible to give a bound on the covering radius of the blockwise direct sum of two codes. This bound depends on a notion, called the k-norm.

Definition 3 *Let $\mathcal{C}$ be the union of k subcodes $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(k)}$. The k-norm N of code $\mathcal{C}$ w.r.t. subcodes $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(k)}$ is the maximum value of $\min_i d(\mathbf{x}, \mathcal{C}^{(i)}) + \max_j d(\mathbf{x}, \mathcal{C}^{(j)})$ over all $\mathbf{x} \in I\!F_2^n$.*

Theorem 4 *[2] If $\mathcal{C}_1 \subset I\!F_2^{n_1}$ has k-norm N_1 w.r.t. subcodes $\mathcal{C}_1^{(1)}, \ldots, \mathcal{C}_1^{(k)}$ and if $\mathcal{C}_2 \subset I\!F_2^{n_2}$ has k-norm N_2 w.r.t. $\mathcal{C}_2^{(1)}, \ldots, \mathcal{C}_2^{(k)}$, then their BDS has covering radius $r \leq \lfloor (N_1 + N_2)/2 \rfloor$.*

4. Some Examples

Bounds for the minimum distance and covering radius of codes constructed via the BDS-construction follow by a straightforward application of Theorem 2 and Theorem 4, once the distances and k-norm are known. In general, determining the k-norm of a code is very hard. Below, we mention one of the results that can be obtained, when the BDS-construction is applied to codes for which determining the k-norm was a feasible task. For details and other results we refer to [3].

Up to now, the best known linear codes with covering radius two and odd codimension were those constructed by Gabidulin et al. [4]. These codes have parameters $[n, n-(2m-1), d]2$, where

$$n = \begin{cases} \frac{5}{4}2^m - 1 & \text{if } d = 3 \text{ or } (d = 4 \text{ and } m = 2), \\ \frac{3}{2}2^m - 3 & \text{if } d = 4 \text{ and } m \geq 3, \\ \frac{23}{16}2^m - 3 & \text{if } d = 4 \text{ and } m \geq 5. \end{cases}$$

These codes have density approximately $1\frac{17}{32}$ if $d = 3$, and $2\frac{17}{256}$ if $d = 4$. Using the BDS-construction, one can do even better: one can construct codes with covering radius two that asymptotically have density 1, i.e. codes which are asymptotically optimal!

Let $m \geq 4$ be even. Then there are systematic codes $\mathcal{D}_{2m-1}^{(d)}$ with parameters $(n, n-(2m-1), d)2$, where

$$n = \begin{cases} 2^m + \frac{5}{4}\sqrt{2^m} - 2 & \text{if } d = 3 \text{ and } m \geq 4 \text{ is even}, \\ 2^m + \frac{23}{16}\sqrt{2^m} - 4 & \text{if } d = 4 \text{ and } m \geq 10 \text{ is even}. \end{cases}$$

In both cases, we find that the density of this class of codes satisfies $\mu(\mathcal{D}_{2m-1}^{(d)}, 2) \to 1$, if $m \to \infty$. This means that we constructed a sequence of asymptotically optimal codes with covering radius two!

References

[1] N.J.A. Sloane, S.M. Reddy, C-L. Chen, "New Binary Codes," *IEEE Trans. Inform. Theory*, Vol. IT-18, pp. 503-510, July 1972.

[2] I.S. Honkala, "On (k, t)-Subnormal Covering Codes," *IEEE Trans. Inform. Theory*, Vol. IT-37, pp. 1203-1206, July 1991.

[3] R. Struik, "Covering Codes," Ph.D. Dissertation, Eindhoven University of Technology, the Netherlands, 1994.

[4] E.M. Gabidulin, A.A. Davydov, L.M. Tombak, "Linear Codes with Covering Radius 2 and Other New Covering Codes," *IEEE Trans. Inform. Theory*, Vol. IT-37, pp. 219-224, January 1991.

The Context Tree Maximizing Method : Some Ideas

Paul A.J. Volf and Frans M.J. Willems

Information and Communication Theory Group, Eindhoven University of Technology

Abstract — **The context tree weighting algorithm was introduced at the 1993 ISIT. Here we are concerned with the context tree maximizing algorithm. We discuss two modifications of this algorithm.**

1. Introduction

In this paper we assume that the source has a tree stucture. With the context (here we use the most recent symbols from the source sequence) one selects one of the leaves. Symbols following this context are assumed to be independent. The structure of the tree is called the *model* of the source. A full tree with depth D and with symbol counts in its nodes and leaves is called a *context tree*. In [2] an one-pass algorithm, the *context tree weighting* algorithm was introduced. This method uses such a tree.

It has been proved for the individual redundancy ρ of a source sequence x_1^T, with respect to a binary source with model $\mathcal{S}$ and with parametervector $\Theta_\mathcal{S}$ that:

$$\rho(x_1^T|x_{1-D}^0, \mathcal{S}, \Theta_\mathcal{S}) < (2|\mathcal{S}| - 1) + (\frac{|\mathcal{S}|}{2}\log\frac{T}{|\mathcal{S}|} + |\mathcal{S}|) + 2.$$

This holds for every model $\mathcal{S}$ and every parametervector $\Theta_\mathcal{S}$. The *context tree maximizing* algorithm (see also [1]), a two-pass algorithm, fulfills the same upperbound, but at the same time, it will give a slightly longer codeword. During the first pass the counts in the tree will be updated. After the first pass the two-pass algorithm will determine the "best" model, and in the second pass it uses this model to compress the sequence. Two-pass algorithms can have distinct advantages. Most important is that their complexity is considerably less than the complexity of the weighting algorithm.

2. The context maximizing algorithm

Just like the weighting algorithm, this algorithm uses the Krichevsky-Trofimov estimator for encoding memoryless sequences. This results in the following block probability for a sequence with a zeros and b ones (if $a > 0$ and $b > 0$) :

$$P_e(a,b) = \frac{\frac{1}{2}\cdot\frac{3}{2}\cdot\ldots\cdot(a-\frac{1}{2})\cdot\frac{1}{2}\cdot\ldots\cdot(b-\frac{1}{2})}{1\cdot 2\cdot\ldots\cdot(a+b)}.$$

In every node of the context tree we compute the maximized probability according to the following formula. With D we denote the maximum level of the tree, and $l(s)$ is the length of the context in node s. Then we define

$$P_m^s = \begin{cases} P_e(a_s, b_s) & \text{if } l(s) = D, \\ \frac{1}{2}\max(P_e(a_s,b_s), P_m^{0s}P_m^{1s}) & \text{if } l(s) < D. \end{cases}$$

One can find the model by walking depth-first through the tree. If the product of the maximized probabilities of the children is larger than the P_e in this node then s must be an internal node of the model, else s is a leaf. The maximizing algorithm will find a model which minimizes the *description length* (MDL). The description length is the sum of the cost needed to describe the model and the cost of describing the data with this model.

3. Restricted number of leaves: the yoyo method

The maximizing algorithm can be constrained by a maximum number of leaves. This limits the complexity of the algorithm. The maximizing algorithm must now find the best model with not more than say C leaves. We walk through the context tree again in a depth-first search way. In every node we compute a list which contains for all $c = 1, C$ the maximized probability achievable with not more than c leaves. If for a node, the maximized probability is reached with c_m nodes, then this list need only contain the entries $1, c_m$. In each node the list can be computed by combining the estimated probability in that node with the lists from its two children.

For every total number of leaves one looks for the distribution of leaves over its two children that results in the highest product of the maximized probabilities. Finally one finds a list in the root with for every number of leaves up to M, the corresponding maximized probability.

To determine the list in the root one needs at most $D+1$ open lists. Once one knows the appropriate total number of leaves, one knows which distribution of the number of leaves over each child resulted in this "optimal" solution. In this way the problem is reduced to two trees of depth $D-1$. If one applies this technique recursively, we will find the best constrained model.

4. Model description on the fly

We could send the entire model description first, followed by the code for the data. To specify the model we need $2|\mathcal{S}| - 1$ bit then. But this can be done in a smarter way. We will send description of parts of the model to the receiver, only if they are needed. The decoder walks through the context tree as far the current model allows. If the current context passes an endpoint (leaf) of the current model, which is not known to be a leaf or internal node of the MDL model yet, and this current context differs from the previous contexts that have passed this endpoint, then the decoder needs more information about the model. We must first tell him that the endpoint is a leaf or not. If not we should give him the same information about the next node on the context path, etc. This process ends when the current context diverges from the previous ones. The diverging node must be included.

In total the encoder has to describe all internal nodes of the found model, plus all leaves (not at the maximum depth) which are followed by different context sequences.

With this technique we gain compared to the first two-pass algorithm. But the model costs in the weighting algorithm are similar. The maximizing algorithms can be modified such that the best "on the fly models" will be found.

References

[1] P. Volf and F. Willems. Context maximizing: Finding MDL decision trees. In *15th Symp. Inform. Theory Benelux*, pp. 192-200, Louvain-la-Neuve, Belgium, May 1994.

[2] F. Willems, Y. Shtarkov, and Tj. Tjalkens. Context tree weighting: A sequential universal source coding procedure for FSMX sources. In *IEEE ISIT*, page 59, San Antonio, Texas, Jan 1993.

Linear Codes for Error Detection on the Local Binomial Channel

Torleiv Kløve
Department of Informatics, University of Bergen, HIB, N-5020 Bergen, Norway

Abstract — **The worst-case probability of undetected error for a linear $[n,k;q]$ code used on a local binomial channel is studied. For the two most important cases it is determined in terms of the weight hierarchy of the code. The worst-case probability of undetected error for simplex codes is determined explicitly. A conjecture about Hamming codes is given.**

The *local binomial channel* was defined implicitly by Korzhik and Fink and explicitly by Korzhik and Dzubanov. It is a channel which is a q-ary symmetric channel for each transmitted symbol, but the symbol error probability may vary from one transmitted symbol to the next.

Let $P_{\text{ue}}(C,\bar{p}) = P_{\text{ue}}(C,p_1,p_2,\ldots,p_n)$ denote the probability of undetected error when a codeword from a linear $[n,k;q]$ code C is transmitted over a local binomial channel with symbol error probability p_i for i'th transmitted symbol. It is easy to see that

$$P_{\text{ue}}(C,\bar{p}) = \sum_{\substack{\bar{c}\in C\\ \bar{c}\neq\bar{0}}} \prod_{i=1}^{n} \left(\frac{p_i}{q-1}\right)^{w(c_i)} (1-p_i)^{1-w(c_i)}.$$

Let the *worst-case error probability* be defined by

$$P_{\text{wc}}(C,v) = \max\Big\{P_{\text{ue}}(C,\bar{p}) \mid 0 \le p_i \le v \text{ for } 1 \le i \le n\Big\}.$$

The *support* of a vector $\bar{c}$ is given by

$$\chi(\bar{c}) = \{i \mid c_i \neq 0\}.$$

For a vector $\bar{c} = (c_1,c_2,\ldots,c_n)$ and a set $X = \{i_1,i_2,\ldots,i_r\}$, where $1 \le i_1 < i_2 < \cdots < i_r \le n$, we let

$$\bar{c}_X = (c_{i_1},c_{i_2},\ldots,c_{i_r}).$$

For an $[n,k;q]$ code C and a set X as above, we define

$$C_X = \{\bar{c}_X \mid \bar{c}\in C \text{ and } \chi(\bar{c}) \subseteq X\}.$$

We use the notation $P^{\text{S}}_{\text{ue}}(C,p)$ for the probability of undetected error when C is used on a q-ary symmetric channel with error probability p. We have

$$P^{\text{S}}_{\text{ue}}(C,p) = P_{\text{ue}}(C,p,p,\ldots,p).$$

Theorem 1 *Let C be an $[n,k;q]$ code. Then*

$$P_{\text{wc}}(C,v) = \max\Big\{P^{\text{S}}_{\text{ue}}(C_X,v) \;\Big|\; X \subseteq \{1,2,\ldots,n\}\Big\}.$$

Theorem 2 *Let C be an $[n,k,d;q]$ code. Then*

$$P_{\text{wc}}(C,1) = \frac{1}{(q-1)^{d-1}}.$$

Theorem 3 *Let C be an $[n,k,d;q]$ code. Let*

$$s = \max\Big\{r \;\Big|\; 1 \le r \le k \text{ and } d_r = d_1 + (r-1)\Big\},$$

where $d_1,d_2,\ldots,d_k$ is the weight hierarchy of C. Then

$$P_{\text{wc}}(C,(q-1)/q) = \frac{q^s-1}{q^{d+s-1}}.$$

Corollary 1 *Let C be an $[n,k,d;q]$ code with minimum distance $d > q$. Then*

$$P_{\text{wc}}(C,(q-1)/q) = \frac{q-1}{q^d}.$$

Corollary 2 *Let C be an $[n,k,d;q]$ code. Then*

$$\frac{q-1}{q^d} \le P_{\text{wc}}(C,(q-1)/q) \le \frac{q-q^{-(k-1)}}{q^d}.$$

We consider a couple of particular classes of codes.

The first class of codes we consider is the binary simplex codes. For each $m \ge 1$ there is a binary simplex code S_m with parameters $n = 2^m-1$, $k=m$, $d_r = 2^m - 2^{m-r}$ for $1 \le r \le m$.

Theorem 4 *For $m \ge 3$, let*

$$v_0(m) = 1-(2^m-1)^{-1/(2^{m-1}-1)}.$$

Then

$$P_{\text{wc}}(S_m,v) = (2^m-1)v^{2^{m-1}}(1-v)^{2^{m-1}-1}$$

for $0 \le v \le v_0(m)$ and

$$P_{\text{wc}}(S_m,v) = v^{2^{m-1}}$$

for $v_0(m) \le v \le 1$.

A similar theorem is true for the first order Reed-Muller codes.

The binary Hamming codes H_m, where $m \ge 1$, have parameters $n = 2^m-1$, $k = 2^m-1-m$, $d=3$. We conjecture that the following result is true for all m (it is true for $m \le 4$).

Conjecture 1 *Define $g_r(v)$ for $r \ge 2$ by*

$$g_r(v) = \frac{1}{2^r}\left(1+(2^r-1)(1-2v)^{2^{r-1}}\right) - (1-v)^{2^r-1}.$$

Let $v_1 = 1$, and for $r \ge 2$ let v_r be the root of the equation $g_r(v) = g_{r+1}(v)$ in the interval $(0,1)$.
Then $v_1 > v_2 > v_3 > v_4 > \cdots$,

$$P_{\text{wc}}(H_m,0,v) = g_m(v)$$

for $0 \le v \le v_{m-1}$, and

$$P_{\text{wc}}(H_m,0,v) = g_r(v)$$

for $v_r \le v \le v_{r-1}$ and $r = 2,3,4,\ldots,m-1$.

We have a similar conjecture for the extended Hamming codes.

Acknowledgements

The research was supported by the Norwegian Research Council, project 100422/410.

Analysing the Computational Performance of Sequential Decoding for the Gilbert-Elliott Channel

Gunilla Bratt, Rolf Johannesson, Kamil Sh. Zigangirov

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Sweden. email: gunilla@dit.lth.se

Abstract — **Fundamental parameters for a sequential decoder are studied, assuming the use of a burst error channel. Expressions are derived for the distribution function of the cumulative metric along the correct path, and for the expected number of computations in an incorrect subtree. They in turn give bounds both on the computational cut-off rate, R_{comp}, and on the maximal transmission rate over the channel, C_D, given metric sets and decoder knowledge assumptions. Some of these results, previously stated in a general form only, are now given with the parameters specified in detail, cf. [Bra 94][1].**

1. Background

Due to its nature, the computational performance of sequential decoding deteriorates drastically when errors occur in bursts, and we have previously proposed a strategy to improve the performance of the decoder in such a situation. Because of its simplicity we have used the stack algorithm when it was necessary to choose a specific algorithm in the analysis.

As our model we chose the Gilbert-Elliott channel, cf. [Gil 60, Ell 63]. Since this channel has two possible states according to the model, the Good and the Bursty, four different channel transitions are possible. Combined with the error probability of each state e_G and e_B, respectively, Fano-like metric increments for the eight situations are formulated. We have chosen to study the case when $0 \leq e_G < e_B \leq 0.5$.

To be able to analyse the behaviour of a decoder working for the Gilbert-Elliott channel we have defined two principal assumptions of the decoder's knowledge of the channel states, namely the *optimistic* and *pessimistic* assumptions, where complete and no knowledge is assumed, respectively. The metric set is developed for each assumption. The results discussed below are derived for both cases. Based on these assumptions we obtain general performance bounds that also are valid for our strategy.

2. Results

An important property of a sequential decoding algorithm is the expected number of computations per decoded node (branch).

By viewing the probabilistic behaviour of the cumulative metric along the correct path as a random walk, we have derived an expression F_D^* for the Gilbert-Elliott channel that is related to the distribution function for the cumulative metric.

With the same approach we have derived an upper bound N_D^* for the expected number of visited nodes in an incorrect subtree. We also show that there is a relation between these expressions.

Combining F_D^* and N_D^* we have found an upper bound $E_D[n]$ for the expected number of computations for correct decoding of one branch. With these results we can also find an expression for $R_{D,comp}$, the maximal transmission rate for which $E_D[n]$ still is finite.

Finally, we derive expressions for the decoding procedure capacities C_D. They are defined as the maximal transmission rates for which we can guarantee that there exists a code such that the probability of decoding error $P_{\mathcal{E}}$ can be chosen arbitrarily small, *given* the two decoding procedure assumptions, respectively. These expressions give only sufficient conditions, in distinction to the ordinary channel capacity definition. We also show that it is necessary that the transmission rate $R < C_D$ in order to obtain a positive expected bit-metric increment along the correct path.

No analytical expression for the channel capacity C_{GE} of the Gilbert-Elliott channel is known, but we show that the optimistic decoding procedure capacity C_o is equal to C_{GE}^R, the channel capacity given that the *receiver* has full channel state knowledge, and also equal to C_{GE}^{TR}, the channel capacity given that *both* the *transmitter* and receiver have full channel state *sequence* knowledge.

References

[Bra 94] G. Bratt, *Sequential Decoding for the Gilbert-Elliott Channel — Strategy and Analysis*, Ph.D. dissertation, Department of Information Theory, Lund University, Lund, Sweden, 1994.

[Ell 63] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels", *The Bell System Technical Journal*, pp. 1977–1997, September 1963.

[Gil 60] E. N. Gilbert, "Capacity of a burst-noise channel", *The Bell System Technical Journal*, vol. 39, pp. 1253–1266, September 1960.

[1] This research was supported in part by the Royal Swedish Academy of Sciences in liaison with the Russian Academy of Sciences, and in part by the Swedish Research Council for Engineering Sciences under Grant 91-91.

Construction of Codes for Localized Errors Based on Ordinary Codes of Even Distance

Per Larsson
Dept. of Electrical Engineering, Linköping University
S-581 83 Linköping, Sweden. Email: perla@isy.liu.se

Abstract — **We give a construction of codes correcting localized errors. It is based on ordinary error correcting codes of even distance. In many cases the new codes outperforms ordinary codes for the same length and error correction capability.**

1. Introduction

We consider binary block codes of length n correcting localized errors. Denote by E a subset of $\{1, 2, \ldots, n\}$ which contains all unreliable positions, i.e. all positions where errors may occur during transmission. It is assumed that there will be no errors outside E. The number of elements in E is denoted by $|E|$. The concept of localized errors, which was introduced by Bassalygo, Gelfand and Pinsker in [1], is characterized by the fact that E is known to the encoder but not to the decoder.

In [2] we present a number of code constructions for localized errors. In many cases those constructions produce useful codes. However, for a lot of codeword lengths, ordinary error correcting codes are still the best known. That motivates the search for new constructions.

The error correction capability of a code is denoted by t. An ordinary error correcting code of length n, size M and minimum distance d will be referred to as an (n, M, d)-code. The maximum size, given n and d, is denoted by $A(n, d)$. A code which can correct t localized errors will be referred to as an (n, M, t)-LE-code.

2. Summary

The codes will be designed to correct t or less errors. Therefore we assume that the size of the set E is less than or equal to t. An ordinary code with minimum distance $2t$ can correct $t - 1$ and detect t errors. When t errors occur there are a number of codewords which are at distance t from the transmitted codeword. The important fact is that the encoder knows where possible errors may occur. Therefore the encoder knows exactly what the decoder has received if t errors are detected. The encoder and the decoder can find the codewords at distance t from a certain vector by the same technique (change one position at a time and decode). These codewords are ordered and numbered in some way and it is important that the encoder and the decoder use the same ordering and numbering. The idea is to add a number, say p, of positions to an ordinary $(n, M, 2t)$ even distance code. If t errors are detected by the decoder of the $(n, M, 2t)$-code the additional p positions are used to determine which codeword (of the possible candidates) was actually transmitted.

Denote by N_t the maximum number of codewords (in the $(n, M, 2t)$-code) at distance t from any vector of length n. Then a sufficient value on p is given by $p = \lceil \log N_t \rceil$. Since we do not know in general how many codewords there are at distance t from an arbitrary vector we may use the following upper bound, $N_t \leq \lfloor n/t \rfloor$. The main result is given by the following theorem.

n	M	t	$A(n, 2t+1)$	Ordinary Code
$2^m + 1$	2^m	2^{m-2}	$2^{m-1} + 2$	S
$2^m + 2$	2^{m+1}	2^{m-2}	$2^m + 4$	R-M
$4t + 2$	$8t$	t	$4t + 4$	H

Table 2: Codes for localized errors with parameters (n, M, t) which exceed $A(n, 2t+1)$ (in these cases the Plotkin bound). The last column indicates which ordinary code is used (S: Simplex, R-M: 1st order Reed-Müller, H: Hadamard).

Theorem 1 *Given an $(n, M, 2t)$-code an $(n+p, M, t)$-LE-code can be constructed, where p is given by the following equation, $p = \lceil \log \lfloor n/t \rfloor \rceil$.*

3. Evaluation

To be of any interest the codes constructed from theorem 1 should in some sense be better than any already known codes. In particular they should be better than ordinary error correcting codes, i.e. codes which do not use the additional channel information. Table 2 shows the parameters of some constructions which exceed the Plotkin upper bound for ordinary codes. Further examples can be found in [3].

4. Remarks

For proofs and a more detailed investigation the reader is referred to [3]. In that paper we also look at the asymptotic performance of the codes.

References

[1] L.A. Bassalygo, S.I. Gelfand, and M.S. Pinsker. Coding for channels with localized errors. In *Proc. Fourth Joint Swedish-Soviet Int. Workshop on Inform. Theory*, pages 95–99, Gotland, Sweden, August 1989.

[2] P. Larsson. Codes for channels with localized errors. Licentiate Thesis LIU-TEK-LIC-1992:38, Linköping University, Linköping, Sweden, November 1992.

[3] P. Larsson. Construction of codes for localized errors based on ordinary codes of even distance. Internal Report LiTH-ISY-R-1619, Linköping University, Linköping, Sweden, 1994.

On the Construction of Quasi-linear Synchronization Codes

A.J. van Wijngaarden

Institute for Experimental Mathematics, Ellernstr. 29, 45326 Essen, Germany

Abstract — **A frame synchronization technique, based on quasi-linear codes [3], provides synchronization of frames with fixed length n in the presence of upto t errors in n consecutive symbols. New code constructions and upperbounds on the redundancy are presented, as well as computer search methods and corresponding results.**

1. Introduction

In digital communication systems the transmitter usually groups data and error control information in so-called frames. A synchronization code can be used to provide the receiver with sufficient information about the position of the frames in the incoming data stream.

A frame containing n q-ary symbols is regarded as a code word $X = x_1x_2x_3\dots x_n$, with $x_i \in \mathcal{A}_q$. The shift operator $T_i(X,Y)$ is defined by $T_i(X,Y) = x_{i+1}x_{i+2}\dots x_ny_1y_2\dots y_i$. The synchronization and error control properties of a code $\mathcal{C} \subset \mathcal{A}_q^n$ are determined by the code distance $d(\mathcal{C})$, and by the code separation $\rho(\mathcal{C})$, defined by

$$\rho(\mathcal{C}) = \min_{\substack{1\leq i\leq n-1\\ X,Y,Z\in\mathcal{C}}} d(T_i(X,Y),Z) \ . \tag{1}$$

Each code $\mathcal{C}$ is comma-free [1] if $\rho(\mathcal{C}) \geq 1$. Correct synchronization and error correction can be guaranteed in the presence of no more than t substitution errors in n successive symbols for a code $\mathcal{C}$ with $d(\mathcal{C}) \geq (2t+1)$ and $\rho(\mathcal{C}) \geq (2t+1)$. Several synchronization code methods have been developed [4] for which $\rho(\mathcal{C}) = 1$, among which comma-free (CF) codes and prefix synchronized (PS) codes. Although the redundancy of the CF-code and PS-code are close to optimal ($\approx \log_q(n)$), the encoding and decoding procedure are complex, and no errors are allowed to occur in the most recent symbols ($t = 0$).

A quasi-linear synchronization (QLS) code, being a coset of a linear code, allows easy encoding and decoding for any separation. A set P of positions, for which the values will be fixed, guarantees separation irrespective the value of the other (data) positions. The redundancy $R(q,n,\rho)$, being equal to $|P|$, is bounded [3] by

$$R_{\min}(q,n,\rho) = \left\lceil \sqrt{\frac{q\rho(n-1)}{q-1}} \right\rceil . \tag{2}$$

An arbitrary code distance $d(\mathcal{C})$ can be obtained using error control codes like BCH-codes and Reed-Solomon codes.

2. Bounds and Code Constructions

The construction of a q-ary QLS-code with arbitrary code separation is generally difficult, especially codes with minimal redundancy $R_{\min}(q,n,\rho)$, so called optimal codes. Using constructions proposed by Clague [2] and Levenshtein [3], optimal binary QLS-codes with separation $\rho \leq 2$ can always be obtained for any length n. For $\rho > 2$, the following upper bounds on the redundancy have been obtained for binary codes, based on construction methods.

Theorem 1 *A binary QLS-code can be constructed with redundancy $R_1(2,n,\rho)$, bounded by*

$$R_1(2,n,\rho) \leq R_{\min}(2,n,\rho) + \varphi(\rho) \tag{3}$$

with $\rho - 2 \leq \varphi(\rho) \leq 3\rho - 2$.

Theorem 2 *For n sufficiently large, a binary QLS-code can be constructed with redundancy $R_2(2,n,\rho)$, bounded by*

$$R_2(2,n,\rho) \leq R_{\min}(2,n,\rho) + \rho - 1. \tag{4}$$

Several search methods can be used to find optimal codes for which $\rho > 2$. Codes have been found for codes with a length n upto 40 as depicted in Figure 1.

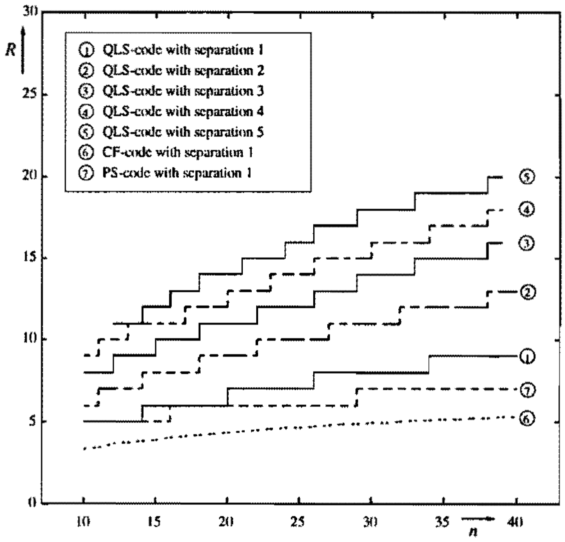


Figure 1. $R(2,n,\rho)$ of CF-code, PS-code and QLS-codes

Using combinatorial methods, some optimal codes can be constructed as well. The development of construction methods to improve the upperbound on the redundancy for any q-ary code of arbitrary length and separation is a topic for further research.

3. Conclusion

Two bounds and code constructions have been obtained. Simulation results support the conjecture, that optimal codes exist for larger separation.

References

[1] S.W. Golomb, B. Gordon, L.R. Welch, "Comma-free codes", *Can. J. Mathematics*, Vol. 10, No. 2, pp. 202-209, 1958.

[2] D.J. Clague, "New Classes of Synchronous Codes", *IEEE Trans. on Electronic Computers*, Vol. EC-16, No. 3, June 1967, pp. 290-298.

[3] V.I. Levenshtein, "One method of constructing quasi-linear codes providing synchronization in the presence of errors", *Problems of Information Transmission*, Vol. 7, No. 3, 1971, pp. 30-40.

[4] J.J. Stiffler, "Theory of Synchronous Communications", *Prentice Hall, Inc.*, Englewood Cliffs, New Jersey, 1971.

Iterative Decoding of Block and Convolutional Codes

Joachim Hagenauer
Chair for Telecommunications, Technical University of Munich, D–80290 M"unchen, Germany
Tel.: +49-89-2105-3466, Fax: +49-89-2105-3490, e-mail: HAG@LNT.e-technik.tu-muenchen.de

Abstract — **In a tutorial manner we will describe some principles of an iterative decoding scheme of two dimensional product codes. With systematic convolutional codes this has been termed "turbo"–(de)coding. It is shown that any combination of block and convolutional codes can be used. The Kullback entropy is used for a simple but effective criterion to stop the iterations.**

Recently [1] systematic feedback convolutional codes have been used to form a kind of interleaved product code by encoding the information twice, directly ("horizontally") and with an interleaved sequence ("vertically" or any other "good" index sequence). The code is binary with elements denoted by +1 and -1. Since the code is systematic the information part has to be transmitted only once. As an example, two rate 2/3 component encoders results in a code of a total rate of 1/2. The codewords are transmitted as the information bits $x_{k,1} = u_k$ and the vertical parity bits $x_{Vk,2}$ as well as the horizontal parity bits $x_{Hk,2}$. Using the respective y values and the channel reliability L_c, the decoding is done in an iterative way. The essence of the iterative decoding steps in [1] is that an almost uncorrelated "extrinsic" information about the bits u_k is passed to the next decoding step and used as a priori log–likelihood ratio L–values

$$L(u_k) = \log \frac{P(u_k = +1)}{P(u_k = -1)}. \qquad (1)$$

The method can be extended to any other binary code in systematic form (block or convolutional, in concatenated or product fashion) as long as a soft–in/soft–out algorithm is used which accepts L–values from the channel and from a priori knowledge, and at the same time produces L–values in the form of

$$L(\hat{u}_k) = \log \frac{P(\hat{u}_k = +1|\mathbf{L_c y})}{P(\hat{u}_k = -1|\mathbf{L_c y})} = L(u_k) + L_{c_{k,1}} y_{k,1} + L_e(\hat{u}_k) \qquad (2)$$

The authors in [1] derive their method by using the Bahl [2] algorithm. Codes represented by a binary trellis can be also decoded by the modified SOVA [3], which is less complex. We have further shown in [6] that for any linear binary block code soft output decoders exist, which have the output format (2).

Using the MAP principle closed formulas for the soft output can be derived from [4] and [5] , which use the code directly or in its dual form. Depending on the code rate, the complexity of the soft-in/soft-out decoder could be less in the dual implementation. Cyclic codes allow an efficient implementation in a pipelined structure.
For hard or soft–outputs the full value of (2) is used. However, for the next vertical or horizontal iterative decoding only the last "extrinsic" term $L_e(\hat{u}_k)$ in (2) is passed on and used as the a priori information $L(u_k)$ in the metric.

One problem of this iterative decoding scheme is to stop the iterations when no further improvement is made, in order to avoid unnecessary iterations. Let $\mathbf{P}^{(i)}(!)$ be the probability distribution and $L_{eV}^{(i)}(!_k)$ be the vertical extrinsic values after the i-th iteration. Then the Kullback entropy as a measure of clossness between two distributions can be asymptotically expressed as

$$\mathrm{E}_{\mathbf{P}^{(i)}}\{\log \frac{\mathbf{P}^{(i)}(!)}{\mathbf{P}^{(i-1)}(!)}\} \approx \sum_k \frac{!_k^{(i)}(L_{eV}^{(i-1)}(!_k) - L_{eV}^{(i)}(!_k))}{1 + \exp|L^{(i)}(!_k)|}. \qquad (3)$$

We stop the iteration when the magnitude of this quantity has dropped by a factor of 1000. This is a very reliable indicator that no more errors will be corrected.

Simulation results will be shown for block and convolutional codes with "turbo"–decoding on Gaussian and fully interleaved Rayleigh fading channels . To mention one result: With two simple memory 2, rate 2/3 convolutional component codes and an interleaver of 30·30 information bits a BER of $8.8 \cdot 10^{-5}$ is achieved with 10 iterations
on an AWGN channel with an E_b/N_0 of 2.5 dB. Using the stop criterion (3) only 3.1 iterations are needed on the average leading to a BER of $10.0 \cdot 10^{-5}$. Therefore we are able to reduce the number of iterations by a factor of 3.3 while missing only a few errors.

References

[1] C. Berrou, A. Glavieux and P. Thitimajhima, "Near Shannon limit error correcting coding and decoding: turbo–codes", Proc. of ICC '93, Geneva, pp. 1064–1070, May 1993.

[2] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", IEEE Trans. Inform. Theory, Vol. IT–20, pp. 284–287, March 1974.

[3] J. Hagenauer, P. Hoeher, "A Viterbi algorithm with soft–decision outputs and its applications", Proc. GLOBECOM '89, Dallas, Texas, pp. 47.1.1- 47.1.7, Nov. 1989.

[4] G.Battail, M.C.Decouvelare, P.Godlewski, "Replication Decoing", IEEE Trans. Inform. Theory, Vol. IT–20, pp. 284–287, March 1974.

[5] C. R. Hartmann, L.D. Rudolph, "An optimum symbol- by- symbol decoding rule for linear codes", IEEE Trans. Inform. Theory, Vol. IT–22, pp. 514–517, September 1976.

[6] J. Hagenauer, E. Offer, L. Papke, "Iterative decoding for block and convolutional codes", under revision for IEEE Trans. Inform. Theory, 1994.

Simulation Results with the 'Turbo' Coding Scheme

Jakob Dahl Andersen

Institute of Circuit Theory and Telecommunication, Technical University of Denmark, DK-2800 Lyngby, Denmark. email : jda@it.dtu.dk

Abstract — **The performance of the 'turbo' coding scheme is measured and an error floor is discovered. To achieve low bit error rates the system is augmented with an outer BCH code. Simulation results for different codes are provided and the complexity of the system is discussed.**

1. Introduction

Recently it has been discovered that a very good performance can be achieved with iterative decoding of a parallel concatenation of small convolutional codes [1]. This coding scheme is named 'turbo' coding. The basic idea is to encode the information sequence twice, the second time after a pseudo-random interleaver, and to do iterative decoding on the two encoded sequences in two decoders. The system can be regarded as a kind of product code. Due to the information exchange among the two decoders the decoding algorithm must provide soft output. We use the Bahl algorithm [2] which actually calculates the a posteriori probability of each information bit. The convolutional codes are used in a recursive systematic form since it gives an improved performance with this system. Consequently, we need a minor modification of the Bahl algorithm.

2. First Simulation Results

The first simulations were based on the recursive systematic code $(1, 1 + D^4/1 + D + D^2 + D^3 + D^4)$. We use the same code for both encoders but for the second one the information sequence is not transmitted. This gives an overall rate of 1/3. We use a block length of 10384 information bits. All numbers including the channel input are represented as floating point values.

As seen from Figure 1, the results achieved with this system are very promising since the Bit Error Rate (BER) after 18 iterations is close to 10^{-5} already at 0.2 dB. Unfortunately the BER decreases very slowly for improved SNR. What we see is many frames with only a few bit errors. This is due to the low free distance of this coding scheme. The free distance might be as low as 10. This is the case when the information pattern for the minimum weight codeword is interleaved to a similar pattern. Although these low weight words exist they might be very rare (only a few specific places in the block of 10384 bits). The actual profile depends on the specific interleaver.

3. Improved Performance

As seen from the first simulations the main problem with the 'turbo' coding scheme is the error floor (or saddle) due to the low free distance. An obvious way to combat this is to use an outer code. Since the bursts consist of very few bit errors, we will use a (10384,10000) BCH code capable of correcting 24 errors. This outer code corrects all the residual errors, but we loose 0.16 dB due to the decreased rate.

Since the occurences of the minimum weight word depend on the interleaver a search for better interleavers might give improved performance. However the performance with interleaver structures like block interleavers are quite poor, and a search among the random interleavers can only remove a couple of the worst low weight words.

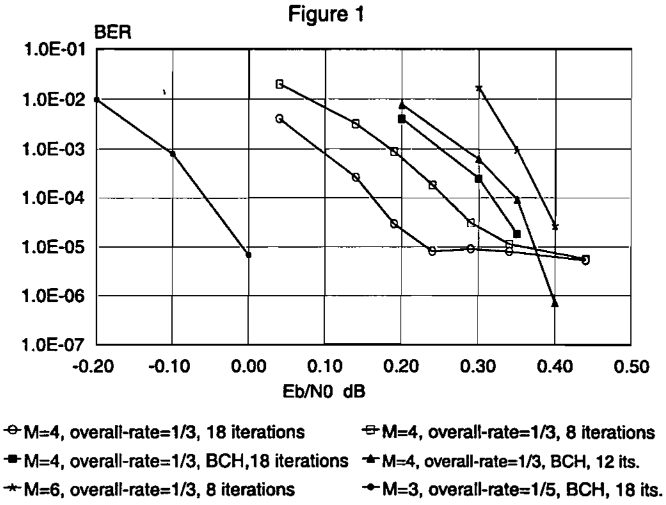
The free distance of the coding scheme can be improved by choosing codes with more states or lower rate. We have made simulations with the CCSDS recommended code with 64 states and rate 1/2. But as seen from Figure 1 there is no improvement for low SNR's. The reason for this might be that the first decoder in the first iteration only has the rate 1/2 code, but the channel capacity is far below 1/2 (but of course not below 1/3).

Finally we have made simulations with a system based on rate 1/3 codes with only 8 states. This gives rate 1/5 for the 'turbo' coding scheme. In this case we have also used the outer BCH code.

4. Complexity

The performance must of course be compared to the complexity. We have tried to estimate the number of operations needed in the Bahl algorithm and conclude that the complexity is about four times the complexity of a Viterbi decoder. This means that the number of operations for 18 iterations with $M = 4$ codes is in the order of 2^{13}. Compared to the Galileo system with iterative use of a $M = 14$ Viterbi decoder the augmented 'turbo' coding scheme can compete on performance as well as complexity.

Figure 1



References

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes(1)," *Proc. ICC'93*, May 1993, pp. 1064–1070.

[2] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, vol. IT-20, March 1974, pp. 284–287.

Interleaving Strategies for Product Codes

Ralf Kötter and Jan Nilsson

Dept. of Electrical Engineering,Linköping University, S-58183 Linköping, Sweden
FOA 72, National Defence Research Establishment, Box 1165, S-58111 Linköping, Sweden

Abstract — **We consider binary block codes that are obtained from short binary component codes. The proposed construction is a generalization of product codes based on combinatorial configurations. The main goal of the paper is to derive codes that are suitable for iterative (turbo) decoding.**

1. Introduction

A very good way to obtain long and powerful codes, which can be decoded efficiently, is to use code concatenation or product codes. Such codes can be decoded by decoding the codes used as components in concatenation (component codes) and combining the results. In [1], Berrou and his co–workers proposed a coding scheme that achieves reliable communication at signal-to-noise ratios very close to the Shannon Limit [1]. They used simple,recursive,systematic convolutional codes as component codes in a product code construction with interleaving.

Our goal is to construct block code based binary codes suitable for the decoding procedure from [1]. The concept of product codes implies that two codewords from the two component codes share precisely one bit. This is in fact the requirement in order to apply the algorithm from [1]. We can interpret an interleaved product code as a bipartite graph where the codewords from the two component codes are the vertices and two vertices are adjacent if the corresponding codewords share a bit. The main objective in the construction of "interleaving rules" is that the girth (the minimum over all cycle lengths in the graph) of the corresponding graph is as large as possible. The reason is that this allows on the one hand to keep the iterated bit-wise estimates obtained in the algorithm from [1] statistically independent over a maximum number of iterations and on the other hand a large girth implies a large minimum distance of the proposed codes.

2. Code Construction

Let C_i,$i = 1, 2$ denote two binary codes with length N_i dimension K_i and minimum Hamming distance D_i. We say that C_i is a $[N_i, K_i, D_i]$ code. We define an interleaving matrix A as an $r \times \nu$ matrix which has precisely N_1 ones in any column, N_2 ones in any row and all other entries equal to zero. A codeword in the interleaved product code C is now obtained by replacing the ones in matrix A with zeros and ones in such a way that the resulting matrix B has the following properties: 1) After row-wise deleting those positions in B, where matrix A has zeros, any row in the resulting matrix is a codeword in the code C_2. 2) After column-wise deleting those positions in B, where matrix A has zeros, any column in the resulting matrix is a codeword in the code C_1. A codeword in C is now defined as matrix B punctured in all position where A had zeros. The resulting code is linear and has length $n = rN_2 = \nu N_1$. Counting the total number of linear conditions imposed by the two requirements on B we see that the dimension k of C satisfies $k \geq n(K_1/N_1 + K_2/N_2 - 1)$. A one in position (i, j) of A implies that in the graph G_A associated with A the vertex corresponding to the i-th codeword from C_2 is adjacent to the vertex corresponding to the j-th codeword from C_1. It is clear that the vertices corresponding to codewords from C_1 and C_2 have degree N_1 and N_2. Let this graph have girth $2l$. We have the following bound on the minimum distance of C. For a proof we refer to a forthcoming paper.

Proposition The minimum Hamming distance d of code C is in case of odd $l \geq 3$ lower bounded by

$$d \geq D_1[1 + D_1(D_2 - 1) \sum_{j=0}^{(l-3)/2} ((D_1 - 1)(D_2 - 1))^j]$$

and in case of even $l \geq 2$ lower bounded by

$$\begin{aligned} d \quad \geq \quad & D_1[1 + D_1(D_2 - 1)[\sum_{j=0}^{(l-4)/2} ((D_1 - 1)(D_2 - 1))^j] \\ & +(D_2 - 1)((D_2 - 1)(D_1 - 1))^{(l-2)/2}]. \end{aligned}$$

The construction of graphs with a large girth and with a fixed number of vertices of a given degree is in general a difficult problem . Given the girth $2l$, we have the following lower bound on the number of edges in the graph or equivalently the length n of the constructed code C:

$$n \geq N_1[1 + N_1(N_2 - 1) \sum_{j=0}^{(l-3)/2} ((N_1 - 1)(N_2 - 1))^j]$$

in case of odd $l \geq 3$ and

$$\begin{aligned} n \quad \geq \quad & N_1[1 + N_1(N_2 - 1)[\sum_{j=0}^{(l-4)/2} ((N_1 - 1)(N_2 - 1))^j] \\ & +(N_2 - 1)((N_2 - 1)(N_1 - 1))^{(l-2)/2}] \end{aligned}$$

in case of even $l \geq 2$.

The task of finding an interleaving matrix for large l is an intriguing and well investigated combinatorial problem. For $l \leq 3$ the problem is reflected in t-design theory. In particular we find that the incidence matrix of a Steiner system $S(2, k, \nu)$ when interpreted as interleaving matrix corresponds to a graph with girth 6 which satisfies the above bound with equality. In this case the component code lengths are $N_1 = \frac{\nu-1}{k-1}$, $N_2 = k$ and the overall length of the code equals $n = \frac{\nu(\nu-1)}{k-1}$. We note that in case of symmetric Steiner systems obtained from projective planes such interleaving rules are easily implemented with the help of difference sets.

References

[1] C. Berrou, A. Glavieux, P. Thitimajshima, *Near Shannon limit error correcting coding and decoding: turbo-codes,* In proceedings of: ICC'93, pp. 1064-1070, Geneva, May, 1993.

Additive Upper Bounds for Turbo–Codes with Perfect Interleaving

Yuri V. Svirid

Chair for Communications, Technical University of Munich, Arcisstr. 21, D-80290 Munich, Germany, *and*
Belarusian State University of Informatics and Radioelectronics, P.Brovka str. 6, 220027 Minsk, Belarus

Abstract — **The linearity of turbo–codes is shown. The criterion for optimal interleaving between two component encoders is given. The union upper bounds on error rate of the whole code with perfect interleaving and component codes with known and binomial (as ideal) weight distribution (WD) are calculated.**

1. Introduction

The codeword of the recently introduced turbo–codes [1] has the following structure

$$F(I) = [I][IG_1][I'G_2], \qquad (1)$$

where $F(\cdot)$ is the function of the encoder, I is the k-tuple of information bits, G_1 and G_2 are the mapping matrices from the space of dimension k to the dimensions r and r' respectively, and I' is a version of I with interleaved (permutated) coordinates. As mappings G_1 and G_2 both systematic block codes and terminated convolutional codes can be used. The rate of the whole code in both cases is $R = k/(k+2r)$ (where here and hereafter we only consider the case $r = r'$), but for terminated convolutional codes the redundancy part is $r = (n_0 - k_0)(k/k_0 + m)$, where it is assumed that each component code has rate $R_c = k_0/n_0$ and memory m.

2. Linearity of Turbo–Codes

The linearity condition for binary codes, $F(A \oplus B) = F(A) \oplus F(B)$, where $\oplus$ denotes the modulo–two addition, should be true for any information vectors A and B. In our case, it immediately implies $(A \oplus B)' = A' \oplus B'$, which is true for any permutation. Thus, only the all–zero codeword has to be transmitted, and only the WD's instead of distance profiles have to be determined.

3. Union Bounds

The classical additive (union) upper bound on the bit error rate for some systematic linear binary $(k+r,k)$ code can be written as follows:

$$P_{BER} \le \sum_{i=1}^{k} \frac{i}{k} \sum_{j=0}^{r} A(i,j) P(C_{i+j}|C_0), \qquad (2)$$

where $A(i,j)$ is the number of codewords with Hamming weight of information bits i and of redundancy bits j, and $P(C_{i+j}|C_0)$ is the probability of error by maximum likelihood decoding for the code of two codewords whose weights are zero (C_0) and $i+j$ (C_{i+j}). This probability is a strictly decreasing function of the weight $i+j$.

Similarly, we can write for the codeword with a structure (1):

$$P_{BER} \le \sum_{i=1}^{k} \frac{i}{k} \sum_{j=0}^{r} \sum_{j'=0}^{r} A(i,j,j') P(C_{i+j+j'}|C_0), \qquad (3)$$

where $A(i,j,j')$ is the number of codewords with weight of information bits i, first redundancy bits j, and second redundancy bits j'.

4. Optimal Interleaving

At first, consider one of two component codes. Dispose all 2^k codewords into k groups so that each i–th ($i = \overline{1,k}$) group consists of $\binom{k}{i}$ codewords of weight i in the information part. Note that if the information vector I belongs to the i–th group, then the permutated vector I' will be also in this group.

Next, let $j(i,l)$, $l = \overline{1,\binom{k}{i}}$, be the weight of the redundancy part of the l–th codeword in the i–th group. Within each group dispose the codewords with non–decreasing weights of the redundancy so that for any l holds: $j(i,l+1) \ge j(i,l)$. Because of the decreasing character of the function $P(C_{i+j+j'}|C_0)$ (3) with increasing weight $i+j+j'$, this average whole weight should be as large as possible. It means, e.g. for the $G_1 = G_2$, that within each group the first redundancy part with small weight after interleaving should be associated with second redundancy part with weight as large as possible and vice versa. In this case we can rewrite (3) as:

$$P_{BER} \le \sum_{i=1}^{k} \frac{i}{k} \sum_{l=1}^{\binom{k}{i}} P(C_{i+j(i,l)+j(i,\binom{k}{i}-l+1)}|C_0), \qquad (4)$$

where for any i and l the $j(i,l)$ are unambigously determined by $A(i,j)$ (2) of the component codes.

In general, the criterion for optimal interleaving can be formulated in terms of WD's of first and second parity parts: $\forall i$: $\overline{j+j'} \longrightarrow \max$, where the expectation is evaluated over above mentioned WD's.

5. WD's of Component Codes and Whole Code

In [2], Battail has proposed the binomial WD: $A(w) = \binom{k+r}{w}/2^r$, where $A(w)$ is the number of codewords with weight w, as optimal WD of some $(k+r,k)$ code. Following the similar idea (random choice of parity–checks inside of each group), we obtain: $A(i,j) = \binom{k}{i}\binom{r}{j}/2^r$. Combining this result with (4), we have

$$P_{BER} \le \sum_{i=1}^{k} \frac{i}{k} \binom{k}{i} P(C_{i+r}|C_0),$$

which is the union upper bound for turbo–codes with optimal component codes, optimal interleaving, and maximum likelihood decoding. The WD of a whole code in this case is exactly (not normalised, but shifted) binomial: $A(0) = 1$, $A(w) = \binom{k}{w-r}$ for $r \le w \le k+r$ and $A(w) = 0$ otherwise.

Acknowledgements: The author would especially like to thank Prof. J. Hagenauer for his wonderful introduction into turbo–codes.

References

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error–correcting coding and decoding: Turbo–codes (1)," *International Conference on Communications (ICC'93)*, Geneva, Switzerland, pp. 1064–1070, May 1993.

[2] G. Battail, "Coding for the gaussian channel: the promise of weighted–otput decoding," *International Journal of Satellite Communications*, vol. 7, pp. 183–192, No. 3, 1989.

Methods for Computing Reliability Information in Concatenated Coding Schemes

Kristian Wahlgren

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden.

Abstract — **For complexity reasons it is often more practical to use a concatenated coding scheme than a single code. To achieve good error correcting characteristics from the outer codes, however, it is necessary that the inner codes supply some form of reliability information in addition to the decoded information sequence. Here it is investigated how to perform this for a convolutional code without adding any trace-back steps to the classical Viterbi algorithm and with a minimum of complexity. It is discussed what inputs are required to do the estimation and an approximation for the a posteriori probabilities for erroneous decoding is analytically derived. It is also shown how to do the estimation using a neural network trained with the a posteriori probabilities for erroneous decisions in a Viterbi decoding.**

1. Introduction

In many situations, when the complexity of the decoder is limited, it is preferable to use sub-optimal decoding of a code that is too large to decode optimally. One way of doing this is to use a concatenated coding scheme instead of a single code. Most decoders perform better if they are provided with soft inputs, *i.e.*, some form of reliability information about the input symbols. To exploit this capability for the outer decoder it is therefore necessary that the inner decoder deliver reliability information in addition to its output sequence.

Since the output errors from the inner decoder is correlated it is not unambiguous how the reliability information is best provided. However, assuming adequate interleaving, reliability information on a symbol-by-symbol basis seems like a good choice.

2. Theory

Our goal is to derive an analytical expression for the reliability information, given only a *window* of the estimated noise. The reason for this is to compute the reliability information using as little complexity as possible.

We assume an AWGN channel with known SNR. We define the reliability sequence $\mathbf{p}$ as the probability for each bit in the received sequence $\hat{\mathbf{x}}^{(i)}$ to be an erroneous estimate of the sent sequence $\mathbf{x}^{(i)}$ conditioned on the received sequence $\mathbf{r}$:

$$p_k = Prob\{\hat{x}_k^{(i)} \neq x_k^{(i)} | \mathbf{r}\}. \qquad (1)$$

By looking at only a window of $\mathbf{r}$ we get an estimate, $\hat{\mathbf{p}}$, of $\mathbf{p}$. With some calculations we can split this expression into parts that with good accuracy can be approximated with Gaussian functions.

3. Two algorithms

We present two algorithms to compute reliability information: the forward-backward algorithm first described in the context of coding by Bahl *et al.* [1], and the soft output Viterbi algorithm (SOVA) suggested by Hagenauer and Hoeher in [2].

4. Estimating reliability using neural networks

In this chapter we use a feed-forward neural network to estimate the *a posteriori* probability that $\hat{x}_k \neq x_k$, given a window of the estimated noise sequence $\hat{\mathbf{e}}$. For large window sizes, the expectation is that the neural network will find good approximations of the reliability information with only small complexity.

We use the neural network as a data fitting function. For this we supply the network with a *training set* of input-output patterns from simulations of the forward-backward algorithm. We try different methods to choose the training set, and we try different sizes of the network.

There are many parameters in the neural network that can be changed, and some of them we choose quite arbitrarily.

Since the neural network contains many multiplications and non-linear functions, it requires that the parallelism of the network be exploited. Thus, a software implementation is not likely to be computationally effective.

5. Results from simulations of the methods

In simulations we see that a single code with comparable rate and complexity to a concatenated coding scheme performs better than any of the chosen concatenated schemes. This was expected since the decoder of the single code uses maximum likelihood decoding.

When the outer decoder uses hard decisions the performance becomes very bad compared to the single code. However, using reliability information computed with the forward-backward algorithm gives a performance close to the single code.

As expected, the performance using neural networks, trained with the numbers that the forward-backward algorithm computes, lies between the performance of hard decisions and MAP-decisions. We also see that the performance for the neural networks becomes better for increasingly big window size. It is somewhat disappointing to see that a relatively large window is necessary to approach the performance of the MAP-decisions, especially for a large code.

We also note that the performance when the outer decoder uses reliability information calculated according to the analytical expression becomes very close to the performance when it uses information from the neural networks.

Acknowledgements

I am grateful to Professor Mitchell Trott at MIT for his guidance throughout this research.

References

[1] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. "Optimal decoding of linear codes for minimizing symbol error rate." *IEEE Transactions on Information Theory*, 20:284-287, March 1974.

[2] J. Hagenauer and P. Hoeher. "A Viterbi algorithm with soft-decision outputs and its applications." *GLOBECOM '89*, 3:1680-1686, October 1989.

Decoding of the Quaternary Goethals Code

Tor Helleseth and P. Vijay Kumar

Department of Informatics, University of Bergen, Høyteknologisenteret, N-5020 Bergen, Norway and Communication Sciences Institute, EE-Systems, University of Southern California, Los Angeles, CA 90089-2565, USA

Abstract — **The quaternary Goethals code is a Z_4-linear code of length 2^m and minimum Lee distance 8. A decoding algorithm is presented which corrects all errors of weight ≤ 3.**

1. Introduction

Let Z_l denote the ring of integers modulo l. Let $\mu : Z_4 \to Z_2$ denote the modulo 2 reduction map. We extend μ to $Z_4[x]$ in the natural way. A monic polynomial $g(x) \in Z_4[x]$ is said to be a monic basic irreducible if $\mu(g(x))$ is a monic irreducible polynomial in $Z_2[x]$. A Galois ring R of characteristic 4 with 4^m elements is isomorphic to the ring $Z_4[x]/(f(x))$, where $f(x)$ is a monic basic irreducible of degree m. The multiplicative group of units R^* of R contains a subgroup of order $2^m - 1$. Let $\beta \in R^*$ be a generator for the multiplicative cyclic subgroup. Let $\mathcal{T} = \{0, 1, \beta, \cdots, \beta^{2^m-2}\}$. It can be shown that $\mu(\mathcal{T}) = GF(2^m)$ and that every element $r \in R$ can be expressed uniquely as $r = A + 2B$ where $A, B \in \mathcal{T}$.

Let $n = 2^m - 1$ where m is an odd integer. The quaternary Goethals code C is the code with parity check matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & \beta & \beta^2 & \cdots & \beta^{2^m-2} \\ 0 & 2 & 2\beta^3 & 2\beta^6 & \cdots & 2\beta^{3(2^m-2)} \end{bmatrix}.$$

In Hammons, Kumar, Calderbank, Sloane and Sole [1], it is shown that if m is odd, then C has minimum Lee distance 8.

2. Decoding of the Goethals code

We index the components of a vector $\mathbf{r} \in Z_4^{n+1}$ by the elements of $\mathcal{T}$. The syndrome of a received vector is $\mathbf{S} = \mathbf{r}H^{tr} = \mathbf{e}H^{tr} = (t, A+2B, 2C)$ where H^{tr} denotes the transpose of H and $\mathbf{e}$ the coset leader. The syndrom equations become

$$\begin{aligned} \sum_{X \in \mathcal{T}} e_X &= t, \quad t \in Z_4 \\ \sum_{X \in \mathcal{T}} e_X X &= A + 2B, \quad A, B \in \mathcal{T} \\ 2\sum_{X \in \mathcal{T}} e_X X^3 &= 2C, \quad C \in \mathcal{T}. \end{aligned}$$

For any coset containing a vector of weight ≤ 3, we will determine the error locations X, Y, Z and the corresponding error values e_X, e_Y, e_Z of a coset leader. Since $\mu(\mathcal{T}) = GF(2^m)$, it is sufficient to find the projections x, y, and z in $GF(2^m)$ and the corresponding values e_X, e_Y, and e_Z in Z_4.

Theorem 1 *Let $\mathbf{S} = (0, A+2B, 2C)$ denote the syndrome.*
(i) If $a = b = c = 0$, then $\mathbf{0}$ is the coset leader.
(ii) If $a \neq 0$ and $a^4 + a^2b^2 + ac + b^4 = 0$, then the coset leader has Lee weight 2 and is uniquely determined by $x = b^2/a + a$, $e_X = 1$, $y = b^2/a$ and $e_Y = -1$.
(iii) If (i) and (ii) do not hold, then any coset leader has Lee weight ≥ 4.

Theorem 2 *Let $\mathbf{S} = (1, A+2B, 2C)$ denote the syndrome.*
(i) If $b = 0$ and $c = a^3$, then the coset leader has Lee weight 1 and is uniquely determined by $x = a$ and $e_X = 1$.
(ii) If $b \neq 0$ and $c = a^3$, then the coset leader has Lee weight 3 and is uniquely determined by $x = a + b$, $e_X = 2$, $y = a$ and $e_Y = -1$.
(iii) If $b \neq 0$, $c \neq a^3$ and $Tr(b^3/(a^3+c)) = 0$, then the coset leader has Lee weight 3. The coset leader is uniquely determined such that x and y are solutions of $b^2u^2 + (a^3+c)u + a^4 + a^2b^2 + ac + b^4 = 0$, $e_X = e_Y = 1$, $z = a + \frac{a^3+c}{b^2}$ and $e_Z = -1$.
(iv) If $\sigma(u) = u^3 + au^2 + (a^2+b^2)u + ab^2 + c$ has three distinct zeros in F then a coset leader has Lee weight 3 and is uniquely determined such that x, y, z are the three distinct zeros in F of $\sigma(u)$ and $e_X = e_Y = e_Z = -1$.
(v) If none of (i)-(iv) hold, then any coset leader has Lee weight ≥ 5.

Theorem 3 *Let $\mathbf{S} = (2, A+2B, 2C)$ denote the syndrome.*
(i) If $a = c = 0$, then the coset leader has Lee weight 2 and is uniquely determined by $x = b$ and $e_X = 2$.
(ii) If $a \neq 0$, $c = a^3 + ab^2$ and $Tr(b/a) = 0$, then the coset leader has Lee weight 2 and is uniquely determined such that x and y are zeros of $u^2 + au + b^2 = 0$ and $e_X = e_Y = 1$.
(iii) If $a \neq 0$, $c = ab^2$ and $Tr(b/a) = 1$, then the coset leader has Lee weight 2 and is uniquely determined such that x and y are zeros of $u^2 + au + a^2 + b^2 = 0$ and $e_X = e_Y = -1$.
(iii) If (i), (ii) and (iii) do not hold, then any coset leader has Lee weight ≥ 4.

Theorem 4 *Let $\mathbf{S} = (3, A+2B, 2C)$ denote the syndrome.*
(i) If $a = b$ and $c = a^3$, then the coset leader has Lee weight 1 and is uniquely determined by $x = a$ and $e_X = -1$.
(ii) If $a \neq b$ and $c = a^3$, then the coset leader has Lee weight 3 and is uniquely determined by $x = b$, $e_X = 2$, $y = a$ and $e_Y = 1$.
(iii) If $a \neq b$, $c \neq a^3$ and $Tr((a+b)^3/(a^3+c)) = 0$, then the coset leader has Lee weight 3. The coset leader is uniquely determined such that x and y are solutions of $(a^2+b^2)u^2 + (a^3+c)u + a^4 + a^2b^2 + ac + b^4 = 0$, $e_X = e_Y = -1$, $z = a + \frac{a^3+c}{a^2+b^2}$ and $e_Z = 1$.
(iv) If $\sigma(u) = u^3 + au^2 + b^2u + a^3 + ab^2 + c$ has three distinct zeros in F then a coset leader has Lee weight 3 and is uniquely determined such that x, y, z are the three distinct zeros in F of $\sigma(u)$ and $e_X = e_Y = e_Z = 1$.
(v) If none of (i)-(iv) hold, then any coset leader has Lee weight ≥ 5.

References

[1] R. Hammons, P.V. Kumar, N.J.A. Sloane, R. Calderbank and P.Sole, The Z_4-Linearity of Kerdock, Preparata, Goethals, and Related Codes, IEEE Trans. on Inform. Theory, 40 (1994) 301-319.

Some Comments on the Theory of Bent Functions

Ernst M. Gabidulin

Moscow Institute of Physics and Technology, 141700 Dolgoprudnyi, Russia

Abstract — **Classification of the bent functions is given if $n = p^m$, p is a prime.**

1. Introduction

Let $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ be a complex valued sequence of length n. The periodic autocorrelation function of $\mathbf{x}$ is defined by $R_{\mathbf{x}}(\tau) = \sum_{s=0}^{n-1} x_s x_{s+\tau}^*$, $\tau = 0, 1, \ldots, n-1$, where all indices are calculated *mod* n and x^* denotes the complex conjugation of x.

Definition 1 *A sequence* $\mathbf{x}$ *is called a* **perfect** *sequence if all the out-of-phase autocorrelation coefficients are equal to* 0, *i.e.*

$$\sum_{s=0}^{n-1} x_s x_{s+\tau}^* = 0, \ \tau = 1, 2, \ldots, n-1. \qquad (1)$$

Definition 2 *A sequence* $\mathbf{x}$ *is called a* **polyphase** *sequence if all the components* x_s *are nth roots of unity.*

Let ζ be an nth primitive root of unity. A polyphase sequence can be represented in the form

$$\mathbf{x} = \left(\zeta^{f(0)}, \zeta^{f(1)}, \ldots, \zeta^{f(n-1)}\right), \qquad (2)$$

where $f(s)$, $s = 0, 1, \ldots, n-1$, are integers *mod* n. A function $f(x)$ is called an *index* function.

Definition 3 *An index function* $f(x)$ *is called a* **bent function** *if and only if the corresponding sequence* $\mathbf{x} = \left(\zeta^{f(0)}, \zeta^{f(1)}, \ldots, \zeta^{f(n-1)}\right)$ *is perfect.*

It is clear that the number of different bent functions is finite. A general construction of bent functions is given in [1]. Nevertheless, this construction does not describe *all* the bent functions. In this paper, we give the full classification of bent functions if $n = p^m$, p is a prime.

2. General properties of bent functions

Let a, b, c be integers and let d be an integer coprime to n, $\gcd(d, n) = 1$.

Lemma 1 *If* $f(x)$ *is the bent function then*

$$f_1(x) = f(dx + c) + ax + b \qquad (3)$$

is also the bent function.

We refer to the bent function $f_1(x)$ as the equivalent bent function.

Corollary 1 *If a bent function* $f(x)$ *is a polynomial, then there exists the equivalent bent function* $f_1(x)$ *of the standard form* $f_1(x) = x^2 + g(x)$, *where* $g(x)$ *is a polynomial of degree not less than* 3, *or* 0.

Theorem 1 *If* $n = 2m$, m *is an odd integer, then a bent function does not exist.*

Theorem 2 *Let* $n = p$, p *is a prime,* $p \geq 3$. *All the bent functions are quadratic polynomials* $f(x) = ax^2 + bx + c$, $\quad a, b, c \in \mathbf{Z}_p$, $\quad a \neq 0$.

Corollary 2 *For this case, all the bent functions are equivalent to the standard bent function* $f(x) = x^2$ *(see Corollary 1).*

Let $n = n_1 n_2$, where $\gcd(n_1, n_2) = 1$. By Chinese Remainder Theorem, each integer x, $0 \leq x \leq n-1$, can be represented in the form $x = x_1 N_1 + x_2 N_2 \ mod\ n$, where $N_1 \equiv b_1 n_2 \equiv 1 \ mod\, n_1$, $N_1^2 \equiv N_1 \ mod\, n$, and $N_2 \equiv b_2 n_1 \equiv 1 \ mod\, n_2$, $N_2^2 \equiv N_2 \ mod\ n$.

The direct-product construction is proposed in [1]. This construction is given by

Lemma 2 *[1] Let* $n = n_1 n_2$, *where* $\gcd(n_1, n_2) = 1$. *If* $f_1(x_1)$, $0 \leq x_1 \leq n_1 - 1$, *and* $f_2(x_2)$, $0 \leq x_2 \leq n_2 - 1$, *are bent functions of sequences of lengths* n_1, n_2, *respectively, then a function*

$$\begin{gathered} f(x) = f_1(x_1)\, n_2 + f_2(x_2)\, n_1, \\ x = x_1 N_1 + x_2 N_2 \ mod\ n, \end{gathered} \qquad (4)$$

is the bent function of a sequence of length n.

This Lemma can be inverted to some extent.

Theorem 3 *Let* $n = p_1 p_2$, *where* p_1 *and* p_2 *are distinct odd primes. Then* $f(x)$ *is a bent function if and only if it can be represented in the form (4).*

Corollary 3 *Let* n *be a square free integer. For this case, all the bent functions can be obtained by the direct-product construction.*

Theorem 4 *Let* $n = p^{2k}$ *be even power of a prime* p. *Let* x_0 *and* x_1 *be the unique representation of* x *given by* $x = x_0 + x_1 p^k$, *where* $0 \leq x_0 \leq p^k - 1$, $0 \leq x_1 \leq p^k - 1$. *Then all the bent functions are given by*

$$f(x) = F(x_0) + x_1 G(x_0)\, p^k, \qquad (5)$$

where $F(x_0)$ *is a function taking values in* $\mathbf{Z}_n$ *and* $G(x_0)$ *is a function taking values in* $\mathbf{Z}_{p^k}$ *such that* $G(a) \neq G(b)$, *if* $a \neq b$, $a, b \in \mathbf{Z}_{p^k}$.

Theorem 5 *Let* $n = p^{2k+1}$ *be odd power of a prime* p. *Let* x_0, x_2 *and* x_1 *be the unique representation of* x *given by* $x = x_0 + x_1 p^k + x_2 p^{k+1}$, *where* $0 \leq x_0 \leq p^k - 1$, $0 \leq x_2 \leq p^k - 1$, $0 \leq x_1 \leq p - 1$. *Then all the bent functions are given by*

$$f(x) = F(x_0) + x_0 x_1 p^k + x_0 G(x_2)\, p^{k+1} + \left[a(x_0)\, x_1^2 + b(x_0)\, x_1\right] p^{2k}, \qquad (6)$$

where i) $F(x_0)$ *is a function taking values in* $\mathbf{Z}_n$, *ii)* $G(x_2)$ *is a function taking values in* $\mathbf{Z}_{p^k}$ *such that* $G(a) \neq G(b)$, *if* $a \neq b$, $a, b \in \mathbf{Z}_{p^k}$, *iii)* $a(x_0)$ *is a function taking non zero values in* $\mathbf{Z}_p$, *iv)* $b(x_0)$ *is a function taking values in* $\mathbf{Z}_p$.

References

[1] H. Chung and P.V. Kumar, "A New General Construction for Generalized Bent Functions," IEEE Trans. Inform. Theory, vol.IT-35, pp. 206-209, 1989

The Correspondence between Projective Codes and 2-Weight Codes

A.E. Brouwer and M. van Eupen
Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract — **The hyperplanes intersecting a 2-weight code in the same number of points obviously form the point set of a projective code. On the other hand, if we have a projective code C, then we can make a 2-weight code by taking the multiset of points $< c > \in PC$ with multiplicity $\gamma(w)$, where w is the weight of $c \in C$ and $\gamma(w) = \alpha w + \beta$ for some rational α and β depending on the weight enumerator of C. In this way we find a 1-1 correspondence between projective codes and 2-weight codes. The second construction can be generalized by taking for $\gamma(w)$ a polynomial of higher degree. In that case more information about the cosets of the dual of C is needed. Several new ternary codes will be constructed in this way.**

Let C be a projective q-ary $[n, k, d]$ code, with nonzero weights $w_1, ..., w_t$. Each subcode D of codimension 1 in C has nonzero weights $w_1, ..., w_t$ with respective frequencies $f_1, ..., f_t$, say, and these frequencies satisfy

$$\sum f_i = q^{k-1} - 1$$

(this follows by counting all nonzero vectors in D), and

$$\sum (n_D - w_i) f_i = n_D (q^{k-2} - 1),$$

where n_D is the effective length of D, that is, the number of coordinate positions where D is not identically zero (this follows by counting the zero entries of all vectors in D).

Since C is projective, we have $n_D = n - 1$ for n subcodes D, and $n_D = n$ for the remaining $(q^k - 1)/(q - 1) - n$ subcodes of codimension 1.

It follows that for arbitrary choice of α, β the sum

$$\sum (\alpha w_i + \beta) f_i$$

does not depend on D but only on n_D, and hence only takes two values.

Fix α, β in such a way that all numbers $\alpha w_i + \beta$ are nonnegative integers, and consider the multiset X (in the projective space PC) consisting of the 1-spaces $\langle c \rangle$ with $c \in C$ taken $\alpha w + \beta$ times, if w is the weight of c. Then X is the point set of a 2-weight code.

For example from a ternary [16,5,9] code with weight enumerator $0^1\ 9^{116}\ 12^{114}\ 15^{12}$ we can construct a ternary [69,5,45] code with weight enumerator $0^1\ 45^{210}\ 54^{32}$ by taking $\alpha = \frac{1}{3}$ and $\beta = -3$.

Now, let C be a 2-weight linear q-ary $[n, k, d]$ code, with nonzero weights u and v. Let X be the corresponding multiset in $PG(k-1, q)$, so that $|X| = n$, and each hyperplane meets X in either $|X| - u$ or $|X| - v$ points. The hyperplanes meeting X in $|X| - u$ points ($|X| - v$ points, respectively) obviously form the point set of a projective code. This construction can be said to be the inverse of the first construction.

For example a ternary [149,5,99] code has been proved to have weight enumerator $0^1\ 99^{222}\ 108^{20}$ if it exists [2]. The second construction would then yield a projective ternary self-orthogonal [10,5,3] code, which cannot exist since 10 is not a multiple of 4.

The first construction can be generalized in the following way: Suppose we have an $[n, k, d]$ code C over $GF(q)$ with nonzero weights $w_1, \ldots, w_t$. Let D be a subcode of codimension 1 of C. Let the frequencies of $w_1, \ldots, w_t$ in D be denoted by $f_1, \ldots, f_t$. Then the Pless power moments [3] give us:

$$\sum_{i=1}^{t} v_i^r f_i = \sum_{j=0}^{n} B_j \left(\sum_{\nu=0}^{r} \nu! S(r, \nu) q^{k-1-\nu} \binom{n-j}{n-\nu} \right) - n^r,$$

where $v_i = n - w_i$, B_j is the number of codewords of weight j in the dual of D and $S(r, \nu)$ is a Stirling number of the second kind. Let $p_i(v) = \sum_{s=0}^{t-1} p_s^{(i)} v^s$ be the polynomial that is 0 for $v = v_h$, $h \neq i$ and is 1 for $v = v_i$, $(i = 1, \ldots, t)$. Consider the set X_i in the projective space PC consisting of 1-spaces $< c >$ $(c \in C)$ with multiplicity $p_i(v)$, where $w = n - v$ is the weight of c. Then X_i is a projective code that is intersected by D in $(\sum_{h=1}^{t} p_i(v_h) f_h)/(q-1)$ points. So if we can compute the weight enumerator up to weight $t - 1$ of the dual of any codimension 1 subcode of C (which corresponds to a coset of the dual of C), then we can compute the weights in X_i $(i = 1, \ldots, t)$, using the Pless power moments. Once we know the weight in X_i corresponding to each coset of the dual for every i, we can construct codes by taking the union of some X_i's.

For example if we take for C the [12,6,6] extended ternary Golay code, then we find a ternary [220,6,144] and a [232,6,153] code, which both improve on the bounds in [1]. If we take for C the ternary [7,6,2] code, then we find a [140,6,90] and a [203,6,132] code, which also improve on the bounds in [1].

References

[1] N. Hamada, A Survey of Recent Work on Characterization of Minihypers in $PG(t, q)$ and Nonbinary Linear Codes Meeting the Griesmer Bound, *J. Combin. Inform. System. Sci.*, Vol. 18, to appear.

[2] R. Hill & D.E. Newton, Optimal ternary codes, *Designs, Codes & Cryptography*, Vol. 2 (1992) pp. 137-157.

[3] V. Pless, Power Moment Identities on Weight Distributions in Error Correcting Codes, *Information and Control*, Vol. 6 (1963) pp. 147-152.

A Parallel Decoding Algorithm for First Order Reed-Muller Codes

P.W. Heijnen

Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract — **A method is described to decode first order Reed-Muller codes by means of parallel processes. At the beginning of every parallel path, the coordinates of the received word are being permuted. The decoding proces tries to find the errors, made during transmission, in the permuted words and corrects these.**

After that, the coordinates of the permuted word are permuted back to their original order and the word passes through some test to conclude if the corrected word is a codeword or not. If it is a codeword and the number of errors is less than or equal to the error correcting capability of the code, then it will be the sent codeword.

Let $\underline{u}_i$ be the binary representation of the integer $i, (0 \leq i < n = 2^m)$, with the least significant bit below. Hence $\underline{u}_0, \underline{u}_1, ..., \underline{u}_{n-1}$ are the successive points of V_m. The first order Reed-Muller code can be written by means of the set of polynomials of degree ≤ 1.

$$\begin{aligned} \mathcal{RM}(1,m) &:= \{(f(\underline{u}_0), f(\underline{u}_1), ..., f(\underline{u}_{n-1})) | f(\underline{x}) \\ &= a_0 + \sum_{i=1}^{m} a_i x_i, a_i \in GF(2)\}. \end{aligned}$$

The vector $\underline{c} := (f(\underline{u}_0), f(\underline{u}_1), \ldots, f(\underline{u}_{n-1}))$ is called the characteristic vector of polynomial f. Let $\underline{c}$ be the characteristic vector of a polynomial $f \in \mathcal{RM}(1,m)$. Then we can find the following 2^{m-1} equations for the coefficient a_m:

$$\left\{ \begin{array}{lcl} a_m & = & c_0 + c_1 \\ a_m & = & c_2 + c_3 \\ & \vdots & \\ a_m & = & c_{n-2} + c_{n-1}, \end{array} \right. \tag{1}$$

and for the coefficient a_{m-1}:

$$\left\{ \begin{array}{lcl} a_{m-1} & = & c_0 + c_2 \\ a_{m-1} & = & c_1 + c_3 \\ & \vdots & \\ a_{m-1} & = & c_{n-4} + c_{n-2} \\ a_{m-1} & = & c_{n-3} + c_{n-1}. \end{array} \right. \tag{2}$$

Suppose, we receive the word $\underline{y}$, while the codeword $\underline{c} \in \mathcal{RM}(1,m)$ has been transmitted. We assume that $\underline{y} = \underline{c} + \underline{e}$ with $w(\underline{e}) \leq 2^{m-2} - 1$. Let us consider the sets:

$$\left\{ \begin{array}{lcl} A_m & := & \{y_0 + y_1, y_2 + y_3, ..., y_{n-2} + y_{n-1}\} \\ A_{m-1} & := & \{y_0 + y_2, y_1 + y_3, ..., y_{n-3} + y_{n-1}\}. \end{array} \right.$$

Because there are at most $2^{m-2} - 1$ errors, we find that a_m is equal to the majority of the values in A_m. The same holds for a_{m-1}: a_{m-1} is equal to the majority in A_{m-1}.

Now, we can decode $\mathcal{RM}(1,m)$ with a number of parallel paths in which only calculation of Equations (1) and (2) will be needed.

We divide the coordinates of $\underline{y}, \underline{e}$ and $\underline{c}$ in blocks of four: $(4i, 4i+1, 4i+2, 4i+3)$ with $0 \leq i < 2^{m-2}$. When we now calculate

$$Y := \{y_{4i} + y_{4i+1}, y_{4i+2} + y_{4i+3}, y_{4i} + y_{4i+2}, y_{4i+1} + y_{4i+3}\}$$

and compare these values with a_m and a_{m-1} (found from set A_m, respectively set A_{m-1}), we can find back $(e_{4i}, e_{4i+1}, e_{4i+2}, e_{4i+3})$, if $w((e_{4i}, e_{4i+1}, e_{4i+2}, e_{4i+3})) \leq 1$. If $w((e_{4i}, e_{4i+1}, e_{4i+2}, e_{4i+3})) > 1$, we cannot find back $(e_{4i}, e_{4i+1}, e_{4i+2}, e_{4i+3})$.

So, if we receive some word $\underline{y}$ with an errorvector $\underline{e}$ we have to find a

$\mathcal{P} \in \text{Aut}(\mathcal{RM}(1,m))$ so that:

$$w((e_{\underline{\tilde{u}}_{4i}}, e_{\underline{\tilde{u}}_{4i+1}}, e_{\underline{\tilde{u}}_{4i+2}}, e_{\underline{\tilde{u}}_{4i+3}})) \leq 1, \forall i, 0 \leq i < 2^{m-2},$$

where

$$\underline{\tilde{u}}_j := \mathcal{P}\underline{u}_j, 0 \leq j \leq n-1.$$

In the above way the decoding problem leads us to the following Key Problem:

We are looking for $\mathcal{P}_1, ..., \mathcal{P}_k \in \text{Aut}(\mathcal{RM}(1,m))$ with $\mathcal{P}_1 = I$, such that k is minimal with the property, that for $\forall \underline{e} \in GF(2)^n$ with $w(\underline{e}) \leq 2^{m-2} - 1$ there exists an κ, $(1 \leq \kappa \leq k)$ with:

$$\begin{aligned} & w((e_{\tilde{u}_{4i}}, e_{\tilde{u}_{4i+1}}, e_{\tilde{u}_{4i+2}}, e_{\tilde{u}_{4i+3}})) \leq 1, \\ & \qquad \forall i, 0 \leq i < 2^{m-2}, \underline{\tilde{u}} := \mathcal{P}_\kappa \underline{u}. \end{aligned}$$

These k permutations and the decoding of the received word after that, can be passed through at the same time in k parallel paths, but there must be some test at the end of every path to conclude if the received word has been corrected completely and correctly and thus if the resulting word is the codeword which was transmitted.

References

[1] MacWilliams,F.J.,Sloane, N.J.A. The Theory of Error-Correcting Codes. (1977)

[2] Tilborg van,H. Error-correcting Codes - a first course. (1993)

Minimal Vectors in Linear Codes and Sharing of Secrets

Alexei Ashikhmin and Alexander Barg

Institute for Inform. Trans. Problems, 19 Ermolovoi st., Moscow GSP-4, 101447

Institute for Inform. Trans. Problems, 19 Ermolovoi st., Moscow GSP-4, 101447, and Eindhoven University, Den Dolech 2,
P.O. Box 513, 5600 MB Eindhoven

Abstract — **We study access structures in secret sharing schemes determined by linear codes. They are known to be characterized by the set of minimal codewords, also termed the projecting set of a code. After stating some simple properties of these sets, we find them for random linear codes, the Hamming codes, and the binary second-order Reed–Muller codes.**

1. Introduction

A center D has to create a system of distributed access to a certain information s_0. Toward this end, it gives out to the users $p_1, \dots, p_{n-1}$ of the system some portions (shares) of information. The goal of the center is to ensure that only authorized coalitions of users, putting their shares together, can learn s_0, while all other (unauthorized) coalitions can obtain from their joint knowledge no information about s_0. Suppose the shares $s_i, 1 \le i \le n-1$, and the value s_0 are taken from a finite set S. The set of authorized coalitions is called an access structure, denoted Γ. A subset $\Gamma^- \subseteq \Gamma$ with the property that $\gamma_1, \gamma_2 \in \Gamma^-$ implies that neither $\gamma_1 \subseteq \gamma_2$ nor $\gamma_2 \subseteq \gamma_1$ is called a *minimal* access structure.

To define a secret-sharing scheme it is necessary to define a set of distribution rules, i.e., of functions that assign shares to the users. If those functions are linear over some finite field, the scheme is called *linear.*

2. Access Structures from Linear Codes

The definitions below were introduced in [1, 2, 3]. Let $H = \|h_{ij}\|, 1 \le i \le r, 0 \le j \le n-1$, be a q-ary matrix and E an $\mathbf{F}_q$-linear space of dimension r. Define a linear transform f by $f(e) = eH, e \in E$. The secret and the shares of the users are formed by coordinate 0 and coordinates 1 to $n-1$ of the vector $f(e)$, respectively. When e runs over E, we obtain the entire collection of distribution rules of the system defined by H.

Thus, we associate the users p_0 (the center) and $p_1, \dots, p_{n-1}$ with the columns of H. A conference of users $p_j \in \gamma \subseteq \{p_1, \dots, p_{n-1}\}$ can reconstruct s_0 iff their columns span column 0 of H. Therefore, an access structure Γ of the secret-sharing scheme defined by H is formed by the subset of the null space of H formed by vectors with a nonzero first coordinate. Let C be a linear code.

Definition. A codeword $c \in C - \{\mathbf{0}\}$, whose leftmost nonzero coordinate is one, is called minimal if it covers no other codeword with the leftmost nonzero coordinate equal to one.

The set of minimal codewords in a linear code C characterizes the minimal access structure Γ of the corresponding scheme and the set of minimal codewords in $C^\perp$ does the same for the dual access structure [4].

We shall discuss simple properties of minimal codewords, which will enable us to give an immediate answer about access structures corresponding to binary Golay codes, binary codes dual to the BCH codes correcting a small number of errors, MDS, and "near-MDS" codes.

Typical linear codes. Let H be a randomly chosen matrix with independent entries taken from $\mathbf{F}_q$ with uniform distribution and $C = \ker H$ be the corresponding $[n, k]$ code.

Theorem 1 *Let C_w be the subset of words of weight $w \le n-k+1$ in C. Then*

$$\mathbf{E}|C_w \cap \mathcal{P}| = \binom{n}{w} \frac{(q-1)^{w-1}}{q^{w(n-k)}} \prod_{i=0}^{w-2} (q^{n-k} - q^i).$$

Hamming codes. Let C be the q-ary Hamming code of length $n = (q^m - 1)/(q-1)$.

Theorem 2 *The set $\mathcal{P}(C)$ is formed by B_w vectors of every weight $w, 3 \le w \le m+1$, were*

$$B_w = \frac{1}{w!(q-1)} \prod_{i=0}^{w-2} (q^m - q^i).$$

Second-order Reed-Muller codes. Let $C = \mathrm{RM}(2, m)$ be the second order binary Reed–Muller code and A_w the number of its words of weight w. Then $A_w = 0$ except for $w = 2^{m-1}, w = 2^{m-1} \pm 2^{m-1-h}, 0 \le h \le \lfloor m/2 \rfloor$. Let $B_w = |C_w \cap \mathcal{P}|$ be the number of its minimal codewords of weight $w > 0$.

Theorem 3 *For $w = 2^{m-1} + 2^{m-1-h}, h = 0, 1, 2$, there are no minimal codewords ($B_w = 0$). Otherwise, $B_w = A_w$, except for the case $w = 2^{m-1}$, when*

$$B_w = \sum_{h=2}^{\lfloor m/2 \rfloor} A_{2^{m-1} - 2^{m-1-h}} (2^{m-2h+1} - 2).$$

Proofs and some further results are given in a manuscript by the same authors available on request.

References

[1] E. F. Brickell, "Some ideal secret sharing schemes," *J. Combin. Math. Combin. Comput.*, **9** (1989), 105–114.

[2] J. Massey, "Minimal codewords and secret sharing," in: *Proc. Sixth Joint Swedish-Russian Workshop Inf. Theory, Mölle, Sweden* (1993), pp. 246–249.

[3] G. R. Blakley and G. A. Kabatianskii, "Linear algebra approach to secret sharing schemes," in: *Error Control, Cryptology, and Speech Compression*, Selected papers from Int. Workshop on Inf. Protection, Moscow, Dec. 1993, Springer Lect. Notes. Comput. Sci., **829** (1994), pp. 33-40.

[4] W.-A. Jackson and K. M. Martin, "Geometric secret sharing schemes and their duals," *Designs, Codes and Cryptography*, **4** (1994), 83–95.

Broadcast Channels with Confidential Messages, with Tampering – the Binary Symmetric Case

Marten van Dijk

Dep. of Math. and Comp. Sc., Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven The Netherlands

Abstract — **The assumptions on which the broadcast channel with confidential messages is based are discussed. Slightly changed, more realistic assumptions lead to a new model, the broadcast channel with confidential messages, with tampering. In order to generate a secret key the legitimate users need to take a certain worst-case scenario into account such that the tampering of the enemy does no harm.**

1. The BCC

Csiszár and Körner introduced the broadcast channel with confidential messages (BCC). It consists of three participants: two legitimate users of the main channel, Alice and Bob, and a wire-tapper, Eve, the enemy. We consider the case where the main channel is a BSC(e_A) (that is a binary symmetric channel with error probability e_A) cascaded with a BSC(e_B). Between these two channels Eve taps the wire with a BSC(e_E). Alice and Bob generate a secret key such that Eve can only obtain a negligible amount of information about it. In order to generate a secret key Alice and Bob first agree upon codes and a protocol to be used.

2. Its Assumptions

The assumptions on which the modelling of the BCC is based are the following:

A1: the protocol and the codes used by Alice and Bob are known to Eve,

A2: Eve knows e_A, e_B, and e_E,

A3: Alice and Bob know e_E, and

A4: Alice and Bob know e_A and e_B.

Suppose we change assumption *A4* into **A4'**:

(i) There exists a continuous injective function f such that for all parts P of the main channel the noise characteristics of P are expected to be equal to $f(w)$ with very small standard deviation ($\approx\approx 0$), where w is the length of part P.

(ii) Alice and Bob know an approximation l' of the length $l > 0$ of the main channel of which they know it is binary symmetric.

(iii) Alice and Bob know an approximation w' of the distance w from Alice at which Eve is wire-tapping.

Let ε be arbitrarily close to 0. The main channel can be seen as the cascading of parts with length ε. By A4'.1 all ε-parts behave similarly with high probability. Therefore A4'.1 can be interpreted by 'each part of the main channel has been made by the same medium (this medium are these ε-parts)'. We conclude that A4' describes a more realistic situation than A4.

We may assume that before Alice and Bob start to generate a secret key they communicate over public noiseless channels to agree on the protocol and codes. During this public communication they can approximate the error probability of the main channel $e = e_A(1-e_B) + (1-e_A)e_B$. Alice transmits n 0s over the main channel, and over the public channel Bob transmits m the number of 1s Bob received. Hence, for n large enough e is expected to be $e' = m/n$ with small standard deviation. Given A4' we can prove that the noise characteristics of a part of the main channel with length w is expected to be a BSC($\frac{1}{2}(1-(1-2e)^{w/l})$) with verry small standard deviation. Hence, $e_A \approx \frac{1}{2}(1-(1-2e')^{w'/l'})$, and $e_B \approx \frac{1}{2}(1-(1-2e')^{(l'-w')/l'})$.

3. The BCC, with Tampering

Suppose Eve tampers by producing extra binary symmetric noise with a source T on the main channel. Let us assume that

A5: at the moment Alice and Bob start to communicate over the main channel they do not know the noise characteristics of T with which Eve tampers.

Then Alice and Bob wil not detect this tampering. Hence, they wil misjudge the situation and they wil generate a key, which they think is secret, and of which Eve obtains a non-negligible amount of information. We conclude that Alice and Bob need to take a special worst-case scenario into account.

Suppose prior to the estimation of e Alice and Bob know that the error probability of 1 meter of the main channel is expected to be a with standard deviation s (not $\approx\approx 0$ as in A4'.1). Then we can prove that given this knowledge the error probability of the main channel is expected by $\frac{1}{2}(1-(1-2a)^l) \approx la$ with standard deviation $\frac{1}{2}\sqrt{((1-2a)^2+4s^2)^l-(1-2a)^{2l}} \approx s\sqrt{l}$. Now, prior to the estimation of e Alice and Bob agree on a set of error probabilities $\mathcal{M}$ indicating when Alice and Bob will use the BCC for secret key generation; that is if and only if $e' \in \mathcal{M}$. Let $\mathcal{M} = [0, l'a + 3s\sqrt{l'}]$. Then they know that with very high probability $e' \in \mathcal{M}$.

Suppose Alice and Bob use the BCC. Then we can prove that with high probability $T \in \{\mathrm{BSC}(p) : 0 \leq p \leq e_T\}$, where $e_T = (e' - l'a + 3s\sqrt{l'})/(1 - 2l'a + 6s\sqrt{l'})$. Now, Alice and Bob wil use a coding strategy for generating a secret key in the situation that Eve tampers with $T = \mathrm{BSC}(e_T)$ (we notice that for this situation they can approximate the corresponding e_A and e_B, see Section II). We can prove that the key generated by this coding strategy remains with high probability secret for Eve in the real situation.

4. Concluding Remarks

The presented case can be generalized towards other BCC's. Also more realistic assumptions for A2 and A3 can be considered. We conclude that in a more realistic situation a certain worst-case scenario has to be taken into account and the noise characteristics of the main channel need regularly be checked (such that changes in the enemy's attack can be detected).

Secure Multiround Authentication Protocols

Christian Gehrmann

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden. email:chris@dit.lth.se

1. Summary

Gemmell and Naor [1] proposed an authentication scheme (without secrecy) in which several messages are passed back and forth to obtain a (Cartesian) A-code in which the keysize is almost independent of the message length.

However the security analysis made by Gemmell and Naor only took into account a certain substitution attack. By also considering the impersonation attack Gehrmann [2] showed that the number of rounds have to be an odd number to avoid impersonation attacks. Further he introduced a special six step substitution attack for which the probability calculation made by Gemmell and Naor did not hold. In this paper, the analysis is developed further. We propose new protocols and prove their security[1]

First we make an analysis of the different possible attacks on a multiround protocol. In a k round authentication protocol there is a transmitter A who wants to send an authenticated message m to a receiver B by using a transmission channel k times. An opponent O might interfere at any time in the communication and put new own false messages into the channel or substitute observed ones. Let P_I and P_s be the probability of a successful impersonation and substitution attack respectively , denote by $\underline{M}^k$ the set of possible message sequences, $\underline{m}_k^A = m_0^A, m_1^A, ..., m_{k-1}^A$ is a by A sent and received message sequence and similar $\underline{m}_k^B = m_0^B, m_1^B, ..., m_{k-1}^B$ is a by B sent and received message sequence. Furthermore let K be the secrete key and denote by $\underline{M}^k(K) \in \underline{M}^k$, the subset of correct sequences under the specific key K.

In the analysis we will use a chosen message substitution scenario, in which we assume that O may freely choose the message part of m_0^A and we then describe the by O controlled sequence as

$$\underline{m}_k^O = m_0'^A, m_0^B, m_1^A, m_2^B, ..., m_{k-2}^A, m_{k-1}^B,$$

where the $'$ mark that O maybe not might control m_0^A completely. We give a proof of the following theorem:

Theorem 1 *The number of possible chosen message substitution sequences $\underline{m}_k^O$ equals*

$$I_c(k) = \binom{k+1}{\frac{(k+1)}{2}}. \tag{1}$$

Next we give a modified secure $k = 3$ round protocol.

Protocol: Let $p > \frac{1}{n}$ and C be a code over $GF(Q)$ with length n and minimum distance d satisfying

$$d \geq n - np$$

and C^A a Cartesian A-code for which the probability for a successful substitution attack equals P_s and the probability of a successful impersonation attack equals $P_I < P_s$. Denote by $C_i(m) \in GF(Q)$ the code symbol at the i-th coordinate of the codeword corresponding to the message m.

(i) A chooses a random number $j, 1 \leq j \leq l$ and sends the message $m_0^A = (j, m)$.

(ii) B receives message m_0^B and chooses a random number $i, 1 \leq i \leq n$. B sends message $m_1^B = i$.

(iii) A receives message m_1^A and uses the code C^A to transmit $m_2^A = C^A(m_1^A, C_{m_1^A}(m_0^A)) =$
$C^A(m_1^A, C_{m_1^A}(j, m))$.

(iv) B receives message m_2^B and calculates
$C^A(m_1^B, C_{m_1^B}(m_0^B)$ and accepts the message sequence as authentic if and only if $m_2^B = C_K^A(m_1^B, C_{m_1^B}(m_0^B))$.

For the protocol above it is possible, by using the previous analysis, to prove the following:

Theorem 2 *Let $a = max_{m,i,c}|\{j : C_i(j, m) = c\}|$. For the $k = 3$ round protocol above*

$$P_s = \max(\frac{a}{l} + (1 - \frac{a}{l})p_s, p + p_s - pp_s) \tag{2}$$

the probability for a successful substitution attack when we also take into account the chosen-message attack.

Construction: Let $Q = 2^r, r = v2^{v-t-1}, l = 2^t$ and let C be an RS-code over $GF(Q)$ with $k = 2^s, r - s = t$. Hence

$$n = Q = 2^r, \qquad d = n - k = 2^r - 2^s.$$

Thus $p = (n - d)/n = k/n = 2^s/2^r = 2^{r-t}/2^r = 2^{-t}$. Furthermore let (j, m) be regarded as the k-tuple $(j \circ m_0, m_1, \cdots, m_{k-1})$ over $GF(Q)$, where j is the first t bits and m_0 the next $r - t$ bits of the element $j \circ m_0 \in GF(Q)$. The code symbol of index β is obtained by evaluating the polynomial $C_\beta(j, m) = j + m_0\beta + \cdots + m_{k-1}\beta^{k-1}$. Let the code C^A be the A-code obtained from a RS-code over $GF(2^v), k = 2^{v-t}$, as suggested in [3], i.e., $P_I = 2^{-v}; P_s = 2^{v-t}/2^v = 2^{-t}$. Thus we have a construction which needs t random bits at the transmission side, r random bits at the receiver side and with a key size of $2v$ bits. Furthermore the construction admits the message size:

$$\log|M| = v2^{v-t-1}2^{v2^{v-t-1}-t} - t. \tag{3}$$

Theorem 3 *For the construction above*

$$P_s < 2^{1-t}. \tag{4}$$

References

[1] P. Gemmell, M. Naor,"Codes for interactive authentication", *Proceedings of CRYPTO '93*, 1993, pp. 355-367.

[2] C. Gehrmann, "Cryptanalysis of the Gemmell and Naor Multiround Authentication Protocol", *Proceedings of CRYPTO '94*, 1994, pp. 121-128.

[3] T. Johansson, G. Kabatanskii, B. Smeets, "On the relation between A-codes and codes correcting independent errors", *Proceedings of Eurocrypt '93*, 1993, pp. 1-11.

[1] This work was supported in part by the Swedish Research Council for Engineering Sciences under Grants 92-662 and 94-457.

Success Probability of Partitioning Cryptanalysis

Carlo Harpes
Signal & Info. Proc. Lab., Swiss Federal Inst. Tech, CH-8092 Zurich, Switzerland

Abstract — **Matsui's linear cryptanalysis of iterated block ciphers has been extended to an attack called partitioning cryptanalysis. This attack exploits a potential weakness of the cipher, namely that one can find a partition of the plaintext space and a partition of the last round input space satisfying the requirement that inputs to the last round are irregularly distributed over the classes of the second partition when the plaintexts are taken from a particular class of the first partition. The success probability of partitioning cryptanalysis is estimated by generalizing a theorem Matsui used to estimate the success probability of linear cryptanalysis.**

1. Introduction

In linear cryptanalysis, Matsui exploits a linear expression between the plaintext and the last round input [3, 1]. His attack is successful if he can find a linear expression whose probability differs substantially from $\frac{1}{2}$ and if he has access to enough plaintext/ciphertext pairs. He then roughly proceeds as follows. For each possible key of the last round, he derives the inputs to this last round from the ciphertexts by decrypting the last round. Thus he is able to verify the linear expression for each plaintext/ciphertext pair, and after considering many plaintext/ciphertext pairs, he estimates the probability of the linear expression under the assumption that the considered last round key is the true key. According to [4], the last round key yielding a probability most distant from $\frac{1}{2}$ is the maximum likelihood estimate of the true last round key in linear cryptanalysis.

Partitioning cryptanalysis [2] uses a partition $\mathcal{A} = \{A_0, \dots, A_{a-1}\}$ of the set of plaintexts, called the input partition, and a partition $\mathcal{B} = \{B_0, \dots, B_{b-1}\}$ of the set of inputs to the last round, called the output partition, with the following property: the inputs to this last round are irregularly distributed over the output classes $B_0, \dots, B_{b-1}$ if all plaintexts are randomly chosen from some fixed input class $A \in \mathcal{A}$. If such partitions exist, we use a maximum-likelihood estimation, just as in linear cryptanalysis, to derive information about the last round key.

The maximum transition probability for the input class A given the fixed key $k^{(1..r-1)}$ is defined as

$$\max_{B \in \mathcal{B}} P[Y^{(r-1)} \in B \mid X \in A, K^{(1..r-1)} = k^{(1..r-1)}]$$

where X is uniformly distributed over A, and denoted by $p_{A|k^{(1..r-1)}}$.

If we replace $Y^{(r-1)}$ by an estimate $\tilde{Y}^{(r-1)} = F_{\tilde{k}}^{-1}(Y^{(r-1)})$ based on the ciphertext and the assumption that $\tilde{k}$ is the true key used in the last round, the probability is supposed to be 0 for all $\tilde{k} \in \tilde{\mathcal{K}} \setminus \{k^{(r)}\}$; $\tilde{\mathcal{K}}$ is the set of potential last round keys among which the attack has to find the true one. This will be called the strong hypothesis of wrong key randomization. Let $|\tilde{\mathcal{K}}| = 2^m$. Note that if $\tilde{k} = k^{(r)}$, $\tilde{Y}^{(r-1)} = Y^{(r-1)}$.

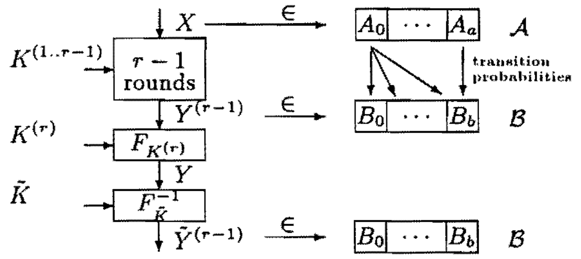


Figure 4: Notation used for partitioning cryptanalysis of iterated block ciphers.

2. Results

Matsui gave a theorem estimating the success probability of linear cryptanalysis. We generalize his idea and prove a theorem estimating the success probability of partitioning cryptanalysis, i.e., the probability that partitioning cryptanalysis finds the true last round key within $\tilde{\mathcal{K}}$.

Theorem 1: *If the strong hypothesis of wrong key randomization is fulfilled, if the counter values can be supposed to be independent, and if the number N of analyzed plaintext/ciphertext pairs with plaintext randomly chosen in A is sufficiently large, then the success probability of a partitioning cryptanalysis exploiting the partition-pair $(\mathcal{A}, \mathcal{B})$ and attacking a cipher with key $k^{(1..r)}$ is given by*

$$\frac{1}{\sqrt{\pi}} \cdot \int_{-\infty}^{\infty} e^{-(u-\sqrt{\frac{R}{2}})^2} \cdot \left(\tfrac{1}{2} + \tfrac{1}{2} \cdot erf(u)\right)^{(2^m-1)b} du \ , \quad (1)$$

where $R := Nb(p_{A|k^{(1..r-1)}} - \frac{1}{b})^2$.

3. Conclusion

We conclude that the success probability is approximately an increasing function of $p_{A|k^{(1..r-1)}}$ (for fixed b and m), and thus $p_{A|k^{(1..r-1)}}$ is a valuable measure for the usefulness of the corresponding partition-pair.

Acknowledgements

It is a pleasure to thank James Massey and Gerhard Krämer for detailed and helpful discussions.

References

[1] C. Harpes, G.G. Kramer, J.L. Massey "Generalized Linear Cryptanalysis and Applicability of the Piling-up Lemma." Fifteenth Symposium on Information Theory in the Benelux, May (1994).

[2] C. Harpes, "Partitioning Cryptanalysis." Post-diploma thesis, ISI, ETH Zurich, October (1994).

[3] M. Matsui, "Linear Cryptanalysis Method for DES Cipher." Abstracts of EUROCRYPT'93, Lofthus, Norway (1993).

[4] S. Murphy, F. Piper, M. Walker, P. Wild, "Likelihood Estimation for Block Cipher Keys", (1994) submitted for publication.

A Parallel Permutation Multiplier Architecture

Tamâs Horvâth, Spyros S. Magliveras and Tran van Trung
Institute for Experimental Mathematics, University of Essen, Ellernstr. 29, 45326 Essen, Germany, and
Department of Computer Science and Engineering, University of Nebraska, Lincoln NE, 68588-0115, U.S.A.

Abstract — **The symmetric key cryptosystem PGM is based on computations in finite permutation groups. PGM is intended to be used in cryptosystems with high data rates. This requires exploitation of the potential parallelism in composition (multiplication) of permutations. As a first step towards a full VLSI implementation, a parallel multiplier has been designed and implemented on an FPGA chip. Here we explain the principles of the architecture and report on the performance of the prototype chip.**

1. Introduction

The symmetric key cryptosystem PGM based on *logarithmic signatures* for finite permutation groups was invented by S. Magliveras in the late 1970's. The system was described in [1]. More literature about PGM itself can be found in [2]. Here we restrict ourselves to implementation aspects.

To effect the fastest possible PGM encryption and decryption operations, one must compute efficiently products and inverses of permutations. Unlike multiplication of integers, composition of permutations is inherently parallelizable. Hence, we can achieve fast computation by designing a parallel permutation multiplier.

2. Principles of multiplication

For easy understanding, we shall explain the principles by means of a simple example. We consider permutations of degree 4 on $\{0,1,2,3\}$, and represent them in *cartesian* form, $\pi = [\pi(0), \pi(1), \pi(2), \pi(3)]$. For example, $\pi = [2,3,0,1]$ is our notation for the permutation $\pi = (0\ 2)(1\ 3)$ as the product of disjoint cycles. This form is particularly convenient for representing permutations in hardware, and needs, in general, $nlog_2n$ bits to represent a permutation of degree n.

The multiplication unit is in essence a crossbar switching network, adapted for the special purpose. A 4x4 switching matrix is depicted in Figure 1. The matrix has three input ports, labeled A, B and C respectively, and one output port named Q. Ports B and C are connected to the vertical lines in the matrix, whereas A and Q to the horizontal lines. At the crosspoints of vertical and horizontal lines reside the *switching cells*, each consisting of a *cell-logic* and a *transfer gate.* If the gate is open (denoted by an asterisk $*$ in the figure), it connects the corresponding vertical and horizontal lines. The input signals, coming from port C, pass through the open gates, and propagate simultaneously towards the output port Q. In the meanwhile signals become rearranged (permuted) according to the configuration of open gates.

But how to configure the gates so that it effects a certain permutation? We found an efficient method of doing this, such that the configuration is computed on the spot, that is in the network itself. The signals coming from port A on horizontal and from B on vertical lines, are compared at each cell by the comparator logic which is responsible for controlling the corresponding gate. If the signals on the neighboring A and B lines are equal, the logic opens, if they are not, the logic closes the transfer gate.

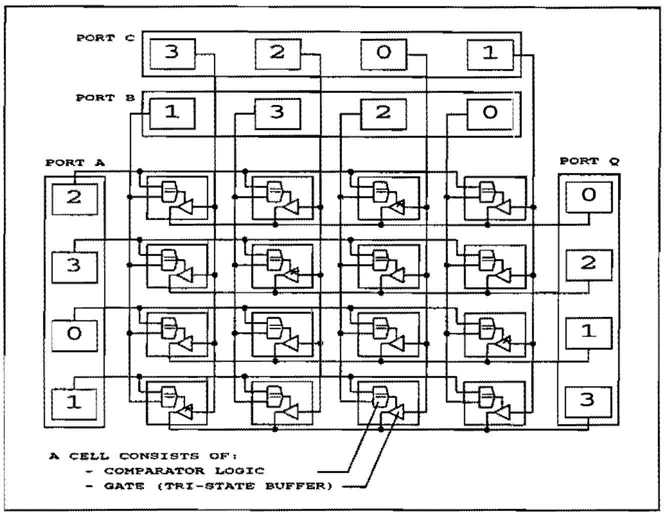


Figure 5: Multiplication in a crossbar network

It is now relatively easy to see that the result Q can be expressed in terms of the other operands as $Q = (A \circ B^{-1}) \circ C$, where $\circ$ denotes composition of permutations and B^{-1} is the inverse of B. Expressions of any kind, composed by using multiplication and inversion, can be evaluated in the network, possibly needing more iterative steps and substitution of some operands by the identity permutation.

3. Implementation details

We have implemented a multiplier on an FPGA (Field Programmable Gate Array) chip. The chip is connected to a DSP (Digital Signal Processor) system. The DSP uses the multiplier chip as a co-processor, it provides the operands and instructions to control the assembly in the multiplier chip. More details of the implementation can be found in [2].

Unfortunately, the FPGA technology allowed us to realize a circuit only for degree $n = 16$. This is of course too small for real applications, however, our multiplier architecture is scalable to larger n. The processing speed is satisfactory, the chip is capable to perform 2.5 million multiplications per second.

As continuation of the project, we intend to implement the entire PGM algorithm for degree 32 on an ASIC (Application Specific IC), and we expect to achieve a speed of 5 million multiplications per second.

References

[1] S. Magliveras, "A cryptosystem from logarithmic signatures of finite groups", In *Proceedings of the 29'th Midwest Symposium on Circuits and Systems,* Elsevier Publishing Company (1986), pp 972–975.

[2] Horváth, S. Magliveras, Tran, "A Parallel Permutation Multiplier for a PGM Crypto-chip", In *Proceedings of Crypto'94,* Springer-Verlag (1994), pp 108-113.

Secrecy Codes for Source Messages of Finite Length

Thomas Johansson

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden, email:thomas@dit.lth.se

Abstract — **In authentication or pure secrecy situations with a source that has a nonuniform distribution, it is desirable to convert the source distribution into a uniform one. For a source with a fixed number of plaintext messages, we show how this can be done using homophonic coding. We also introduce a new measure of secrecy, the maximum guessing probability, and show a relation to $H(M|C)$, the source equivocation[1].**

1. The model

We use the model introduced by Simmons for authentication. The transmitter sends the plaintext message, denoted by m, and taken from the finite set $\mathcal{M}$, to the receiver by mapping m into a ciphertext message c from the finite set $\mathcal{C}$. The mapping is determined by the shared key k chosen from $\mathcal{K}$. When the receiver receives a message, he uses k to determine the plaintext message m.

2. Secrecy codes

The amount of secrecy that a certain code provides is, according to Shannon's theory of secrecy, described by the *source equivocation* $H(M|C)$. If $H(M|C) = H(M)$, the code is said to provide perfect secrecy. We will now introduce a second measure. This will be the maximum probability that the enemy *guesses* the correct plaintext after observing a ciphertext. This probability is denoted P_G. Observing c, the enemy's guess will be a plaintext message that maximizes $P(m|c)$, i.e., $\max_m P(m|c)$. By maximizing over $c \in \mathcal{C}$, we formally define the *maximum guessing probability* P_G as

$$P_G = \max_{m,s} P(s|m). \tag{1}$$

The two measures are related by the following inequality.
Lemma 1

$$\log P_G \geq -H(M|C) \geq H(K|C). \tag{2}$$

Note that this relation is similar to the two relations for P_I and P_S given in Simmons' bounds in authentication, i.e., $\log P_I \geq -I(C;K)$ and $\log P_S \geq -H(K|C)$.

For uniformly distributed sources, secrecy codes having zero redundancy can easily be provided. Shannon's results together with Stinson's results [Stin90] give that for perfect secrecy $|\mathcal{K}| \geq |\mathcal{M}|$. Perfect secrecy can be obtained by the Vernam cipher

$$c = (m + k), \tag{3}$$

where $c, s, k \in Z_n$. Since we have perfect secrecy we also have that $P_G = \max_m P(m) = 1/n$.

Assume a nonuniform source distribution for $\mathcal{M}$. We consider the following way of constructing a uniform distribution using homophonic coding [Günt88]. Put the plaintext messages in "subsets", such that the probability of each subset is approximately the same as the most probable plaintext. The subsets are protected with perfect secrecy. We then add some unprotected bits to specify the particular plaintext message in the encrypted subset. We will give away information about M, and thus we will not have perfect secrecy, but P_G will remain approximately the same!

The strategy is implemented by the following algorithm. Let the probability of the most probable plaintext message be denoted p_0, and assume $|\mathcal{M}| \geq 3$. Let $x = \lfloor 1/p_0 \rfloor$. Here x will be the cardinality of the key set, which means that we will have x subsets.

Let $p_0 \geq p_1 \geq \ldots \geq p_{|\mathcal{M}|-1}$, where $P(M = m_i) = p_i$, and p_i is a rational number. Write $x^{-1}, p_0, \ldots, p_{|\mathcal{M}|-1}$ in common rational form

$$x^{-1} = \frac{b}{y},\ p_0 = \frac{c_0}{y}, \ldots,\ p_{|\mathcal{M}|-1} = \frac{c_{|\mathcal{M}|-1}}{y}.$$

In subset 0 we put m_0. The subset is of size b and m_0 of size c_0. Thus $b - c_0$ is the remaining part of the subset. Here we put the next plaintext message, in this case m_1. We continue like this until we reach the case $b - c_0 - \cdots - c_k < 0$. Then m_k must also be put in subset 1, and the size for m_k in subset 1 will be $c_k + \cdots + c_0 - b$. It is necessary that *the unprotected bits corresponding to m_k in subset 0 and in subset 1 are different.* We continue to fill up all the subsets in the same way. Then, when we want to transmit a certain M, say $M = m_k$, we randomly choose among the c_k possibilities, and transmit the subset with perfect secrecy and the remaining bits without secrecy.

An opponent observing the unprotected bits knows exactly to which source state each subset corresponds. Since the subsets are encrypted with perfect secrecy and all subsets correspond to different plaintext messages, he can do no better than to guess the subset, and succeed with probability $1/x$, i.e., $P_G = 1/x$.

The conclusion of our discussion is the difference between demanding perfect secrecy and demanding lowest possible P_G, i.e., $P_G = p_0$.

Theorem 2 *If we demand perfect secrecy, then*

$$|\mathcal{K}| \geq |\mathcal{M}|, \tag{4}$$

where equality can always be obtained by a Vernam cipher.
If we demand $P_G = \max_s P(s) = p_0$, then

$$|\mathcal{K}| \geq 1/p_0, \tag{5}$$

where equality can be obtained if p_0 can be written as the reciprocal of an integer.

References

[Günt88] C.G. Günther "A universal algorithm for homophonic coding", *Proceedings of Eurocrypt'88*, Davos, Switzerland, 1988, LNCS 330, Berlin: Springer-Verlag, pp. 405–414.

[Stin90] D.R. Stinson, "The combinatorics of authentication and secrecy codes", *Journal of Cryptology*, vol. 2, no. 1, 1990, pp. 23–49.

[1] This work was supported in part by the Swedish Research Council for Engineering Sciences under Grants 92-662 and 94-457.

Asymptotically Good Codes with Algebraic Curves

Ruud Pellikaan

Discrete Mathematics, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

A sequence of codes (C_m) with parameters $[n_m, k_m, d_m]$ over a fixed finite field $\mathbf{F}_q$ is called asymptotically good if n_m tends to infinity, and d_m/n_m tends to a non-zero constant δ, and k_m/n_m tends to a non-zero constant R, if m tends to infinity. If $R > 1-H_q(\delta)$, then the codes exceed the Gilbert-Varshamov bound. It was shown by Tsfasman, Vladut and Zink [6, 5] that there exist asymptotically good geometric Goppa codes on modular curves such that $\delta + R \geq 1 - (\sqrt{q} - 1)^{-1}$. If moreover $q \geq 49$, then these codes are better that the Gilbert-Varshamov bound. The theory of modular curves is very deep and the construction of some these curves and their codes can be done in theory with polynomial complexity [4, 5] but are still to involved to have been constructed.

In this lecture I will discuss the attempt of Feng and Rao [1, 2] to construct asymptotically good codes with complexity $\mathcal{O}(n^3)$ using generalized Klein curves. Up to now their methods were not sufficient to prove their claims, but by a slight change of the equations of the curves Garcia and Stichtenoth [3] could prove that these curves have the required properties.

References

[1] G.-L. Feng and T.R.N. Rao, Improved geometric Goppa codes Part I, Basic Theory, preprint 1994 .

[2] G.-L. Feng and T.R.N. Rao, Improved geometric Goppa codes Part II, Generalized Klein codes, peprint 1994.

[3] A. Garcia and H. Stichtenoth, Algebraic function fields with many rational places, Workshop on Algebraic Geometry and Coding Theory, Valladolid, October 10-15, 1994.

[4] Yu.I. Manin and S.G. Vlăduţ, Linear codes and modular curves, Journ. Sov. Math. **30** (1985), 2611-2643.

[5] M.A. Tsfasman and S.G. Vlăduţ, Algebraic-geometric codes, Mathematics and its Applications **58**, Kluwer Acad. Publ., Dordrecht, 1991.

[6] M.A. Tsfasman, S.G. Vlăduţ and T. Zink, Modular curves, Shimura curves and Goppa codes, better than Varshamov-Gilbert bound, Math. Nachrichten **109** (1982), 21-28.

A Note on Implementing Sakata's Algorithm

Jan Åberg

Department of Information Theory, Lund University, P.O. Box 118, S-221 00 Lund, Sweden. email: jan@dit.lth.se

Abstract — **In [1], Sakata extended the Berlekamp-Massey algorithm [2] to n-dimensional arrays. Here further analysis of the structures described there is made, and a few corollaries are reformulated accordingly to produce a version of the algorithm suited for software implementation.**

Given an n-dimensional array u over a field K, Sakatas's algorithm finds a minimal set of n-variate polynomials with coefficients in K that are valid for u, i.e., a set of linear recurring relations capable of generating the array. The principle is the same as for the one-dimensional case; to iteratively modify the polynomials by adding some multiple of a polynomial valid for a smaller part of the array using the Berlekamp procedure. If the set of polynomials satisfies the criterion that it constitutes a Gröbner basis over the n-dimensional polynomial ring $K[z]$ ([1],[3]), the polynomial set defines an n-dimensional linear feedback shift register that produces the array as its output.

In the one-dimensional case, at each iteration the degree of the minimal (not necessarily unique) polynomial capable of generating the sequence as seen so far is computed directly, whereupon the Berlekamp procedure is used to find a valid polynomial of this degree. Applying the corresponding theorem in the n-dimensional case, however, does not directly give the minimal degree set S, but a set of points C defining the *excluded point set* Γ_C of the sub-array seen so far (Figure 1). This set corresponds to the shape of a LFSR defined by a set of minimal valid polynomials. Even though this set of polynomials may not be unique, the excluded point set, and thus the shape of the LFSR, are.

Thus, the set S of minimal degrees must be obtained from the set of points C defining the excluded point set, at each iteration. For any new minimal degree $s^+ > s$, $s^+ = q - c$ or $s^+ = \max(q - c, s)$ for some $c \in C$, where q is the point currently treated. To obtain the set S of minimal degrees in an efficient way from the set of points C defining the excluded point set, it is necessary to keep track of not only the points in each set, but also of their mutual relations. Specifically, the set ISC of pairs of *adjoined points* is of interest. These are defined as follows.

> A point s is adjoined to a point c iff, for some $i \in I = \{1, \dots, n\}$, $s_i = c_i + 1$, $s_j \leq c_j$ $(j \neq i)$.

This relation is written as $s \vdash c$. Using two lemmas presented in [1], the sets S^+ and C^+, i.e., S and C for the next iteration, can be computed from the current S, C and ISC in an efficient way at each iteration.

Also, at each iteration ISC must be updated. Two corollaries of the lemmata mentioned above provide criteria for identifying all points in $c^+ \in C^+$ to which each new minimal degree s^+ is adjoined. From an implementation point of view, it is desirable that each c^+ satisfies exactly one of these conditions, to make sure that it is found only once. We show that by removing two conditions from each corollary this will be fulfilled.

ISC must also be updated for the points s that are not changed during the iteration. We give a slight modification of one of the conditions mentioned above, which will provide a sufficient criterion. Regrettably, in this case the updating procedure will be of higher complexity than in the other cases. For bounded $|S|$ and $|C|$, i.e., for periodic arrays, the overall complexity will still be of order $O(|p|^2)$, where $|p|$ is the number of points in the array, dominated by the Berlekamp procedure. For a non-periodic array not satisfying any set of linear recurring relations, the updating may dominate, with an approximate worst-case complexity of order $O(|p|^{(6-\frac{5}{n})})$ for an n-dimensional array.

References

[1] S. Sakata, "Extension of the Berlekamp-Massey algorithm to N dimensions", *Information and Computation*, vol. 84, no. 2, pp. 207–239, Feb. 1990.

[2] J.L. Massey, "Shift-register synthesis and BCH decoding", *IEEE Transactions on Information Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.

[3] B. Buchberger, "Gröbner bases: An algorithmic method in polynomial ideal theory", *Multidimensional Systems Theory* (N. K. Bose, Ed.), Chapter 6, pp. 184–232, Reidel, Dordrecht, 1985.

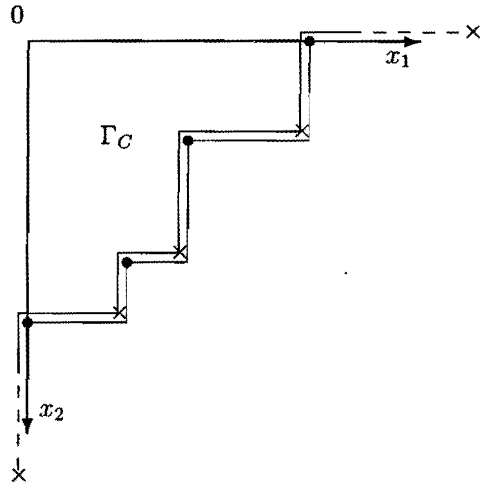


Figure 6: Γ_C in the 2D case ($\bullet = s \in S$, $\times = c \in C$).

The Generalized Hamming Weights of Some Hyperelliptic Codes

Mario A. de Boer

Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, the Netherlands

Abstract — **In this paper we determine all generalized Hamming weights of a class of algebraic geometric codes arising from hyperelliptic curves.**

1. Generalized Hamming weights

For an arbitrary code D we define the support as

$$\text{supp}(D) = \{i \mid \text{there is a } d \in D \text{ with } d_i \neq 0\}.$$

Let C be a code with parameters $[n, k, d]$. For any r, $1 \leq r \leq k$ we define the r-th generalized Hamming weight as

$$d_r = \min\{\#\text{supp}(D) \mid D \text{ } r\text{-dimensional subcode of } C\}.$$

Since the definition of generalized Hamming weights by Wei in [2], many papers have appeared that investigate these parameters for different classes of codes.

2. Algebraic geometric codes

Let X be an absolutely irreducible smooth curve over $\mathbb{F}_q$ of genus g. For a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of rational points of X and a rational divisor G of X with $\deg(G) < n$ and $\text{supp}(G) \cap \mathcal{P} = \emptyset$, we define the algebraic geometric code $C(\mathcal{P}, G)$ as the image of the map

$$\phi : \ L(G) \longrightarrow \mathbb{F}_q^n, \quad f \longmapsto (f(P_1), \ldots, f(P_n)).$$

The code $C(\mathcal{P}, G)$ is linear with parameters $[n, k, d]$ satisfying $k = l(G) \geq \deg(G) + 1 - g$ and $d \geq n - \deg(G)$.
In the papers [1] and [4] the authors study the generalized Hamming weights of algebraic geometric codes.

3. Hyperelliptic codes

An absolutely irreducible smooth curve X is hyperelliptic if and only if its genus is at least two and there exists a morphism of degree two from X to the projective line. X allows a unique involution (conjugation), the hyperelliptic involution, denoted by σ. The fixed points of σ are called hyperelliptic points. In this paper P_∞ is a fixed hyperelliptic point and $\mathcal{H} = \{H_1, H_2, \ldots, H_h\}$ is the set of all (not necessarily $\mathbb{F}_q$-rational) hyperelliptic points on X different from P_∞.
Here we consider algebraic geometric codes $C(\mathcal{P}, G)$ arising from hyperelliptic curves, with the properties that for any rational point $P \in \mathcal{P}$ we have that $\sigma(P) \in \mathcal{P}$, and G is a hyperelliptic divisor (which means $G \sim 2lP_\infty$ for some l) of degree $\deg(G) < n$. From Clifford's theorem and the Riemann-Roch theorem we find that the dimension of these codes is $k = l+1$ if $l \leq g-1$ and $k = 2l + 1 - g$ if $l > g - 1$. Remark that this class of codes includes the most studied form of algebraic geometric code: codes $C(\mathcal{P}, G)$ with $G = mP_\infty$ and $\mathcal{P}$ all rational points on X except P_∞.
By determining all generalized Hamming weights of these codes we generalize a result by Xing ([3]) who determined their minimum distance if $l > g - 1$ and q odd. It also generalizes some results in [1], in which Munuera gives some bounds on the generalized Hamming weights of these codes.

4. The generalized Hamming weights

After proving some lemma's concerning divisors of hyperelliptic curves it is possible to prove the following results.
Let $W_1, \ldots, W_\omega \in \mathcal{H}$ be $\mathbb{F}_q$-rational hyperelliptic points and let $P_i, \sigma(P_i)$, $i = 1, \ldots, \pi$ be pairs of distinct conjugated $\mathbb{F}_q$-rational points of X. Then we have the following theorem.
Theorem 1 *Let*
$\mathcal{P} = \{W_1, \ldots, W_\omega, P_1, \sigma(P_1), \ldots, P_\pi, \sigma(P_\pi)\}$ *and* $G \sim 2lP_\infty$ *with* $2l < n = 2\pi + \omega$. *Let* $\pi = l - \Delta$ *for some* Δ. *Then the code* $C(\mathcal{P}, G)$ *has generalized Hamming weights*

$$d_r = \begin{cases} n - 2l + 2(r-1) + \min\{\Delta - r + 1, 2g + 2 - \omega\} \\ \qquad \text{if } 1 \leq r \leq \min\{l - g, \Delta\} \\ n - 2l + r - 1 + \Delta \\ \qquad \text{if } l - g + 1 \leq r \leq \Delta \\ n - 2l + 2(r-1) \\ \qquad \text{if } \Delta + 1 \leq r \leq g \\ n - k + r \\ \qquad \text{if } r \geq g + 1. \end{cases}$$

Here $k = l + 1$ *if* $l \leq g - 1$ *and* $k = 2l + 1 - g$ *if* $l \geq g$.

5. Examples: maximal hyperelliptic curves

In order to construct long codes of the type that we are considering in this paper, we need hyperelliptic curves with both many $\mathbb{F}_q$-rational points and many hyperelliptic points. In this section we will give examples of curves that attain the Weil bound and have the maximal possible number of hyperelliptic points.
Let q be odd. Then a hyperelliptic curve X of genus g has a (singular) plane model of the form $y^2 = f(x)$, with f a square-free polynomial of degree $2g + 1$ or $2g + 2$. The following proposition gives a class of hyperelliptic curves that meet the Weil bound.
Proposition 1 *Let* $g \geq 2$ *such that* $p = 2g + 1$ *is a prime power. Set* $q = p^2$. *Let* N *be the number of* $\mathbb{F}_q$*-rational points on the hyperelliptic curve* X *with plane model*

$$y^2 = x^p + x.$$

Then X *has genus* g, *contains* $2g + 2$ $\mathbb{F}_q$*-rational hyperelliptic points and* $N = q + 1 + 2g\sqrt{q}$.

References

[1] C. Munuera, "On the Generalized Hamming Weights of Geometric Goppa Codes," to appear in IEEE Trans. Inform. Theory.

[2] V.K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inform. Theory*, 37, pp. 1412–1418, 1991.

[3] C.-P. Xing, "Hyperelliptic Function Fields and Codes," *Journal of Pure and Applied Algebra*, 74, pp. 109–118, 1991.

[4] K. Yang, P.V. Kumar, H. Stichtenoth, "On the Weight Hierarchy of Geometric Goppa Codes," *IEEE Trans. Inform. Theory*, 40, pp. 913–920, 1994.

On Termination Criteria for Decoding Algorithms

Iwan M. Duursma

LMD Equipe ATI, Case 930, 13288 MARSEILLE CEDEX 9, France.

Abstract — **In the rapid development of decoding algorithms for geometric Goppa codes, termination criteria seem to form a neglected part. We present two criteria that obtain their information from the decoding process and not from a priori assumptions on the error pattern. Their use should fasten correction of small errors and allow for correction of some errors of large weight.**

Let R be an affine ring (the ring of functions regular outside a fixed rational point P on a smooth complete absolutely irreducible curve of genus g defined over a finite field F). The parity checks of a *one-point geometric Goppa code* $C(m)$ are obtained by evaluation of functions from R of pole order at most m in the rational points different from P. Decoding algorithms for these codes use recursion relations, corresponding to functions from R, on the finite array S of known syndromes.

Theorem. Let the syndromes be known up to order m. Let f_0 be a recursion relation of smallest degree t_0 on the finite array S of known syndromes. For $m \geq 2t_0 + 2g - 1$, the recursion relation f_0 generates the unique infinite array $S' \supset S$ of smallest rank that is compatible with the known syndromes. Let $I = Rf_0 + Rf_1 + \ldots + Rf_v$ be the R-ideal generated by the recursion relations on the finite array S, with v minimal, and such that the pole orders $t_0, t_1, \ldots, t_v$ of $f_0, f_1, \ldots, f_v$ satisfy $t_0 < t_1 < \ldots < t_v$. For $m \geq t_0 + t_v + 2g - 1$, the ideal I describes the recursion relations valid on the infinite array S'.

Example. The case of a t-error-correcting Reed-Solomon code and error pattern of weight t corresponds to $R = F[X]$, $g = 0$, $m = 2t - 1$, $t_0 = t$, $v = 0$.

We remark that the bounds will in general apply only after some unknown syndromes have been computed with the Feng-Rao majority scheme. But since the bounds do not depend on an a priori assumption on the weight of the error, they will be useful once the error is small compared to the capability of the code. On the other hand they allow correction of some error patterns of weight beyond the capability of the code. An extended version of this abstract including a proof of the theorem, a comparison with known termination criteria, and nontrivial examples is in preperation.

Acknowledgement. This work was supported by NWO, The Netherlands.