[12] G. Caire and S. Shamai, "On the capacity of some channels with channel state information," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2007–2019, Sep. 1999.

[13] V. Lau, Y. Liu, and T. Chen, "Capacity of memoryless channels and block-fading channels with designable cardinality-constrained channel state feedback," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2038–2049, Sep. 2004.

[14] C. Shannon, "Channels with side information at the transmitter," *IBM J. Res. Develop.*, vol. 2, pp. 289–293, 1958.

[15] M. Salehi, "Capacity and coding for memories with real-time noisy defect information at encoder and decoder," *Inst. Electr. Eng. Proc. Commun. Speech Vis.*, vol. 139, pp. 113–117, Apr. 1992.

[16] M. Skoglund and G. Jöngren, "On the capacity of a multiple-antenna communication link with channel side information," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 3, pp. 395–405, Apr. 2003.

[17] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, achievabile rates, and sum-rate capacity of Gaussian MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2658–2668, Oct. 2003.

[18] N. Jindal, S. Vishwanath, and A. Goldsmith, "On the duality of Gaussian multiple-access and broadcast channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 768–783, May 2004.

[19] N. Jindal, S. Vishwanath, and A. Goldsmith, "On the duality of Gaussian multiple-access and broadcast channels," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun.–Jul. 30–5, 2002, p. 500.

[20] V. Lau, Y. Liu, and T. Chen, "On the design of MIMO block-fading channels with feedback-link cardinality constraint," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 62–70, Jan. 2004.

[21] D. J. Love, R. W. Heath Jr., and T. Strohmer, "Grassmannian beamfroming for multiple-input multiple-output wireless systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2735–2747, Oct. 2003.

[22] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, "On beamforming with finite rate feedback in multiple antenna systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2562–2579, Oct. 2003.

[23] D. J. Love and R. W. Heath Jr., "Limited feedback unitary precoding for spatial multiplexing systems," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2967–2976, Aug. 2005.

[24] W. Santipach and M. L. Honig, "Asymptotic performance of MIMO wireless channels with limited feedback," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2003, vol. 1, pp. 141–146.

[25] W. Santipach and M. L. Honig, "Asymptotic capacity of beamforming with limited feedback," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun.–Jul. 27–2, 2004, p. 289.

[26] W. Santipach and M. L. Honig, "Achievable rates for MIMO fading channels with limited feedback and linear receivers," in *Proc. Int. Symp. Spread Spectrum Techniques Appl.*, Aug.–Sep. 30–2, 2004, pp. 1–6.

[27] C. K. Au-Yeung and D. J. Love, "On the performance of random vector quantization limited feedback beamforming in a MISO system," *IEEE Trans. Wireless Commun.*, to be published.

[28] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, pp. 311–335, 1998.

[29] D. Shiu, G. J. Foschini, M. J. Gans, and J. M. Kahn, "Fading correlation and its effect on the capacity of multielement antenna systems," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 502–513, Mar. 2000.

[30] A. Goldsmith, S. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 684–702, Jun. 2003.

[31] D. J. Love and R. W. Heath Jr., "Limited feedback diversity techniques for correlated channels," *IEEE Trans. Veh. Technol.*, vol. 55, no. 2, pp. 718–722, Mar. 2006.

[32] N. Jindal and A. Goldsmith, "Dirty-paper coding versus TDMA for MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1783–1794, May 2005.

[33] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[34] S. Verdú, "Multiple-access channels with memory with and without frame synchronism," *IEEE Trans. Inf. Theory*, vol. 35, no. 3, pp. 605–619, May 1989.

[35] P. Xia, S. Zhou, and G. B. Giannakis, "Achieving the Welch bound with difference sets," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1900–1907, May 2005.

[36] P. Xia and G. B. Giannakis, "Design and analysis of transmit-beamforming based on limited-rate feedback," *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1853–1863, May 2006.


# Construction of Capacity Achieving $(M, d, \infty)$ Constrained Codes With Least Decoder Window Length

Ashwin Kumar and Kees A. Schouhamer Immink, *Fellow, IEEE*

***Abstract*—We present capacity achieving multilevel run-length-limited (ML-RLL) codes that can be decoded by a sliding window of size $2$.**

***Index Terms*—Channel capacity, $(M, d, \infty)$ constrained sequences, run-length-limited sequences.**

## I. Introduction

Multilevel run-length-limited (ML-RLL) codes produce $(M, d, k)$ constrained sequences that have at least $d$ and at most $k$ zeros between consecutive nonzero $M$-ary symbols [1], [2]. The Shannon capacity [3] of an $((M, d, k)$ constrained channel is rational for certain values of $M$ and $k = \infty$ [4]. In [5], capacity achieving codes were constructed with fewest number of states and having a sliding-block decoder of window size $d + 1$. In this correspondence, we propose a new construction procedure that designs an encoder with the fewest number of states and has a decoder with a look ahead of one codeword.

## II. Preliminaries

To measure the efficiency of an $(M, d, \infty)$ constrained code, it is necessary to know its Shannon capacity $C$. The characteristic equation describing an $(M, d, \infty)$ constrained channel is

$$z^{d+1} - z^d - (M - 1) = 0. \tag{1}$$

The Shannon capacity $C$ is

$$C = \log_2 \lambda \tag{2}$$

where $\lambda$ is the largest real root of (1). The efficiency of a code $R/C$, for a code with rate $R$, is less than $1$ for codes having irrational capacity. In [4], it was shown that there exist $(M, d, \infty)$ constrained sequences having rational capacity. These values of $M$ for any $0 < d < \infty$ are given by

$$M = 1 + 2^{md}(2^m - 1) \tag{3}$$

where $m \geq 1$ is an integer. Substituting the value of $M$ into (1), we find $\lambda = 2^m$, and hence $C = m$. In [5], the state-splitting algorithm is used to construct capacity achieving codes. Further, by choosing unity length codewords and finding the eigenvector $v$ guiding the state-splitting algorithm, a lower bound on the number of encoder states is derived to be $2^{md}$. This design results in a sliding-block decoder of window size $d + 1$.

In Section III, we present an alternative procedure for designing an encoder with minimum number of states. The new code has the virtue of having a sliding-block decoder with window size $2$.

Manuscript received December 18, 2007. Current version published September 17, 2008.

A. Kumar is with the Electrical and Computer Engineering Department, National University of Singapore, Republic of Singapore, and Data Storage Institute, A*Star, Singapore, Republic of Singapore.

K. A. S. Immink is with the Institute for Experimental Mathematics, 45326 Essen, Germany (e-mail: immink@exp-math.uni-esen.de).

Communicated by H.-A. Loeliger, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2008.928971

## III. New Design Procedure

Let $S$ represent the set of encoder states, where $|S| = 2^{md}$. The set $S$ is partitioned into $d+1$ subsets $\{S_0, S_1, \ldots, S_d\}$. The cardinality of each subset $S_i$, $i \in \{0, 1, \ldots, d\}$ is

$$|S_0| = 1 \tag{4a}$$

$$|S_i| = 2^{mi} - \sum_{j=0}^{i-1} |S_j|, \qquad i = 1, 2, \ldots, d. \tag{4b}$$

And the accumulative sum of cardinalities of the subsets up to index $i$ is $2^{mi}$. Choosing codeword length $q = d$ and number of user bits $p = md$, we construct capacity achieving codes of rate $R = p/q = m$.

The number of starting zeros for each codeword determines the subset of encoder states to which it is allocated. The codewords can be expressed as $0_n u 0_{d-n-1}$, where $0_n$ is a zero string of length $n$ and $u \in \{1, 2, \ldots, M-1\}$. Thus, $n = d$ gives the all-zero codeword, and $n = 0$ gives codewords starting with a nonzero symbol. The codewords starting with $d - i$ zeros, i.e., codewords of the form $0_{d-i} u 0_{i-1}$ are allocated to states in subset $S_i$. Hence, the all zero codeword is allocated to the state in $S_0$; the codewords starting with $d-1$ zeros are allocated to the states in $S_1$; and so on. This ensures that the codewords allocated to different subsets are disjoint. Further, the codewords allocated to each state in $S_i, i \neq 0$, can be made disjoint by the following allocation. Since $q = d$, the number of codewords having $n$ starting zeros, $n \neq d$, equals $M - 1$. Hence, for each state in subset $S_i, i \neq 0$, the number of codewords allocated are

$$\frac{M-1}{2^{mi} - 2^{m(i-1)}} = 2^{m(d-i+1)}. \tag{5}$$

The $d$ constraint dictates the *concatenation* rules for codewords allocated to the various subsets. With the above allocation, concatenation of codewords in subset $S_i$ with codewords in $\{S_{i-1}, S_{i-2}, \ldots, S_0\}$, preserves the $d$ constraint. Using this codeword *concatenation* rule, the rules for transitions between the encoder states are written as follows.

1) For the state in $S_0$, connect the outgoing edges to states in $S_1, S_2, \ldots, S_d$, including a self-loop to the state in $S_0$. The all-zero codeword is assigned to each edge.
2) For each state in subset $S_i$, connect the outgoing edges to states in the subsets $S_{i-1}, S_{i-2}, \ldots, S_0$. The codewords allocated to each state in $S_i$ are of the form $0_{d-i} j 0_{i-1}$.

Since the codewords in each subset and their allocation to each state in the subset are disjoint, we can assign the same codeword multiple times to different input information words resulting in a unique codeword-next-state pair.

We now prove that the assignment of codewords is sufficient to represent all possible input information words for each state in the encoder. Using (5) and combining it with the transition rules, we find that for a state in $S_i, i \neq 0$, each codeword can be assigned $\sum_{j=0}^{i-1} |S_j|$ times to different outgoing edges. Thus, we get the total number of outgoing edges from each state as

$$\frac{(M-1)\sum_{j=0}^{i-1} |S_j|}{2^{mi} - 2^{m(i-1)}} = \frac{(M-1)2^{m(i-1)}}{2^{mi} - 2^{m(i-1)}} = 2^{md} \tag{6}$$

where we substitute for $\sum_{j=0}^{i-1} |S_j|$ using the accumulative sum of subsets up to index $(i-1)$. Thus, for each state the codewords can be assigned to $2^{md}$ outgoing edges, which equals $2^p$.

Thus, the above rules define an encoder for $(M, d, \infty)$ capacity achieving codes. Since the codeword-next-state pair is unique, the sent information word can be uniquely decoded by observing two consecutive received codewords. Hence, we get a sliding-block decoder with one codeword look-ahead, which is implementable using two lookup tables. Next, we use the above construction procedure and present some examples by designing $(13, 1, \infty)$, $(49, 2, \infty)$, and $(9, 3, \infty)$ capacity achieving codes.

### A. $(13, 1, \infty)$ Code

The $(13, 1, \infty)$ code, obtained by substituting $m = 2$, $d = 1$ in (3), has a rational capacity $C = m = 2$. The minimum number of encoder states required is $2^{md} = 4$. The states are partitioned into two subsets $\{S_0, S_1\}$ with $|S_0| = 1$ and $|S_1| = 3$. Choose codewords of length $d = 1$, and allocate the codewords to subsets as follows.

1) The zero codeword, allocated to the state in subset $S_0$ is assigned to each of the $2^m = 4$ outgoing edges.
2) The remaining $M - 1 = 12$ codewords are partitioned into three sets of four each, and assigned to each of states in set $S_1$. The codewords are assigned to the outgoing edges leading to the state in $S_0$.

A decoding window of length 2, observing two consecutive symbols, is sufficient to decode the sent information word. The above encoder is equivalent to the one presented in [5].

### B. $(49, 2, \infty)$ Code

For $m = 2$, $d = 2$, the minimum number of encoder states is given by $2^{md} = 16$. We partition encoder states into three subsets $\{S_0, S_1, S_2\}$, where $|S_0| = 1$, $|S_1| = 3$, and $|S_2| = 12$, and choose the codeword length $q = d = 2$. Following the design procedure, the state in $S_0$ has the zero codeword on all its outgoing edges. Each state in $S_1$ has 16 codewords starting with a zero, $M - 1 = 48$ codewords partitioned into three sets, and is assigned to the edges leading to the state in $S_0$. Finally, the codewords beginning with a nonzero symbol, i.e., $M - 1 = 48$ codewords partitioned into sets of four each are allocated to the 12 states in $S_3$. They are each assigned to four times to outgoing edges leading to states in sets $S_0$, and $S_1$. Thus, accounting for the 16 outgoing transitions in each state, the decoder is of window size 2.

### C. $(9, 3, \infty)$ Code

Let $m = 1$ and $d = 3$. An encoder with minimum number of states $2^{md} = 8$ with codewords of length $d = 3$ can be constructed as follows. The encoder states are partitioned into four subsets with cardinalities $|S_0| = 1$, $|S_1| = 1$, $|S_2| = 2$, and $|S_3| = 4$. The encoder is designed as follows.

1) The state in $S_0$ has the all-zero codeword on all its outgoing edges to states in subsets $S_1$, $S_2$, and $S_3$,
2) The state in $S_1$ has eight codewords starting with two zeros and assigned to outgoing the edges leading to the state in $S_0$,
3) Each state in $S_2$ has four codewords, each assigned twice to outgoing edges leading to states in $S_0$ and $S_1$,
4) Each state in $S_3$ has two codewords, each assigned four times to outgoing edges leading to states in $S_0$, $S_1$ and $S_2$.

Thus, we can design an encoder with minimum number of states and having a decoder window of size 2.

### D. Code Design With Unity Length Codewords

To construct encoders with unity length codewords, the design procedure is modified as follows. We partition the set of encoder states into subsets as before. The outgoing edges from states in $S_i$ are directed to states in $S_{i+1}$. For the state in $S_0$, we have an additional self-loop, while the outgoing edges from the final state subset $S_d$ are directed to the state in $S_0$. With the exception of outgoing edges from states in $S_d$, all the outgoing edges have zero codeword assigned to them. The $M - 1$ nonzero codewords are assigned to the outgoing edges of states in $S_d$ as given by (5), substituting $i = d$. This design procedure gives a sliding-block decoder of window size $d + 1$ and the resulting encoder structure is similar to the one derived in [5].

## IV. Conclusion

We have presented a code construction procedure that can be used to design encoders for $M$-ary $(d, \infty)$ constrained codes having rational capacity and results in a one codeword look-ahead decoder.

## References


[1] S. W. McLaughlin, "Five runlength-limited codes for $M$-ary recording channels," *IEEE Trans. Magn.*, vol. 33, no. 3, pp. 2442–2450, Mar. 1997.

[2] K. A. S. Immink, *Codes for Mass Data Storage Systems*, 2nd ed. Eindhoven, The Netherlands: Shannon Foundation Publishers, 2004, ISBN 90-74249-27-2.

[3] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul. 1948.

[4] S. W. McLaughlin, J. Luo, and Q. Xie, "On the capacity of $M$-ary runlength limited codes," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1508–1511, Sep. 1995.

[5] S. W. McLaughlin, "The construction of $M$-ary $(d, \infty)$ codes that achieve capacity and have fewest number of encoder states," *IEEE Trans. Inf. Theory.*, vol. 43, no. 2, pp. 699–703, Mar. 1997.


# Computing a Lower Bound of the Smallest Eigenvalue of a Symmetric Positive-Definite Toeplitz Matrix

Teresa Laudadio, Nicola Mastronardi, and Marc Van Barel, *Member, IEEE*

***Abstract*—In this correspondence, several algorithms to compute a lower bound of the smallest eigenvalue of a symmetric positive-definite Toeplitz matrix are described and compared in terms of accuracy and computational efficiency. Exploiting the Toeplitz structure of the considered matrix, new theoretical insights are derived and an efficient implementation of some of the aforementioned algorithms is provided.**

***Index Terms*—Cholesky factorization, eigenvalues, Levinson–Durbin algorithm, QR factorization, symmetric positive-definite matrix, Toeplitz matrix.**

## I. Introduction

Computing a lower bound of the smallest eigenvalue of a symmetric positive-definite (SPD) Toeplitz matrix $T$ is of considerable interest in a variety of signal processing applications and estimation problems [2], [11]. Moreover, the condition number, defined as $\|T\|_2\|T^{-1}\|_2$, is an important information when computing the solution of linear systems involving such matrices, and an estimate for $\|T^{-1}\|_2$, i.e., the smallest singular value of the SPD matrix $T$, is more difficult to achieve [5], [6]. The aim of this paper is manifold: to provide a short survey of the most significant direct algorithms available in the literature for general SPD Toeplitz matrices, to compare their performance, to reveal new useful theoretical insights of the considered methods and, by exploiting the matrix Toeplitz structure, to derive an efficient implementation of some of them.

In particular, in our studies we consider the following algorithms: the method by Ma and Zarowski [8], the two algorithms proposed by W. Sun in [12], the Newton-based method proposed by Mastronardi and Boley [9], and the algorithm by C. Fassino [3].

Although the authors of [8] and [12] suggest that a fast implementation of the methods is possible in the Toeplitz case, no analysis was carried out. Here, we focus on this aspect and report a comparison of the flop counts[1] of the different methods.

An additional iterative algorithm for computing a lower bound of the smallest eigenvalue of a SPD Toeplitz matrix is described in [7]. Unfortunately, the lower bound obtained after the first one or two iterations of this algorithm does not give a significant estimation and it will not be considered in this survey. Nevertheless, it is worth stressing that the lower bound yielded by the methods proposed in this paper could be used as starting value for the algorithm in [7].

We show that one of the methods proposed by Sun is equivalent to Fassino's method, and that both Sun's algorithms can be implemented in an efficient way. The paper is organized as follows: in Section II, we introduce notation, in Sections III–VI, the aforementioned methods and their improved implementations are described; in Section VII, the numerical results are reported; in Section VIII, the main conclusions are formulated; finally, in the Appendix section, the equivalence between the method of Sun and the one of Fassino is proved.

## II. Notation

Let $T_n$ be an $n \times n$ SPD Toeplitz matrix:

$$T_n = \begin{bmatrix} t_0 & t_1 & \cdots & t_{n-1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}.$$

We denote by

$$t^{(n-1)} = [\, t_1 \quad t_2 \quad \ldots \quad t_{n-1} \,]^T$$

the vector comprising of the off-diagonal elements in the first column of $T_n$, by $I_n$ the identity matrix of order $n$, and by $E_{n-1}$ the exchange matrix of order $n-1$

$$E_{n-1} = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}.$$

We partition the matrix $T_n$ as follows:

$$T_n = \begin{bmatrix} T_{n-1} & E_{n-1}t^{(n-1)} \\ t^{(n-1)^T}E_{n-1} & t_0 \end{bmatrix} \tag{1}$$

where $T_{n-1}$ is an $(n-1) \times (n-1)$ matrix, $t_0$ is the diagonal entry of $T_n$, and $t^{(n-1)^T}$ denotes the transpose of $t^{(n-1)}$. It is well known

Manuscript received November 30, 2007; revised June 10, 2008. Current version published September 17, 2008. The research of N. Mastronardi was partially supported by the Italian MIUR under Grant 2004015437. The research of M. Van Barel was partially supported by the Research Council K.U. Leuven, project OT/05/40 (Large rank structured matrix computations), CoE EF/05/006 Optimization in Engineering (OPTEC), by the Fund for Scientific Research–Flanders (Belgium), G.0423.05 (RAM: Rational modeling: optimal conditioning and stable algorithms), and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization). The scientific responsibility rests with its authors.

T. Laudadio and N. Mastronardi are with the Istituto per le Applicazioni del Calcolo "M. Picone," Consiglio Nazionale delle Ricerche, Bari 70126, Italy (e-mail: t.laudadio@ba.iac.cnr.it; n.mastronardi@ba.iac.cnr.it).

M. Van Barel is with the Department of Computer Science, Katholieke Universiteit Leuven, Leuven 3001, Belgium (e-mail: Marc.VanBarel@cs.kuleuven.be).

Communicated by U. Mitra, Associate Editor for At Large.

Digital Object Identifier 10.1109/TIT.2008.928966

[1] As in [4], we consider only the highest order terms in the flop counts neglecting the lower order ones.