



LUND UNIVERSITY

On the performance of turbo codes and convolutional codes of low rate

Leanderson, Carl Fredrik; Edfors, Ove; Maseng, Torleiv; Ottosson, Tony

Published in:
Proc. IEEE Vehicular Technology Conference

DOI:
[10.1109/VETECF.1999.801556](https://doi.org/10.1109/VETECF.1999.801556)

1999

[Link to publication](#)

Citation for published version (APA):

Leanderson, C. F., Edfors, O., Maseng, T., & Ottosson, T. (1999). On the performance of turbo codes and convolutional codes of low rate. In *Proc. IEEE Vehicular Technology Conference* (Vol. 3, pp. 1560-1564). IEEE - Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/VETECF.1999.801556>

Total number of authors:
4

General rights

Unless other specific re-use rights are stated the following general rights apply:
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

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

On the Performance of Turbo Codes and Convolutional Codes of Low Rate

Carl Fredrik Leanderson¹, Ove Edfors¹, Torleiv Maseng¹ and Tony Ottosson²

¹Dept. of Applied Electronics, Lund University, Lund, Sweden. E-mail: cfl@tde.lth.se

²Dept. of Signals and Systems, Chalmers University of Techn., Gothenburg, Sweden

Abstract—Recently two new classes of low-rate codes have been presented. The first class is the super-orthogonal turbo codes (SOTCs) and the second is the maximum free distance (MFD) convolutional codes.

In this paper we present an evaluation of the performance vs. arithmetic decoding complexity for these codes and compare them with the previously reported super-orthogonal convolutional codes (SOCCs). For all classes of codes, the arithmetic decoding complexity is estimated, and the error performance on the additive white Gaussian noise channel is simulated. The SOCCs offer performance comparable to that of the MFD codes. However, the existence of good SOCCs is restricted to a small number of rates while the MFD codes give high performance for a multitude of rates. For the parameters used in this investigation the SOTCs yield higher performance at lower arithmetic decoding complexity than the MFD codes.

I. INTRODUCTION

The high spreading factor in a direct-sequence code-division multiple-access (DS-CDMA) system encourage the employment of low-rate channel codes to improve the performance. However, this has not been a widespread technique due to the lack of good low-rate codes.

Recently, two new classes of low-rate codes with improved performance have been proposed. In [1] a coding scheme that combines turbo codes [2] with super-orthogonal convolutional codes (SOCC) [3], into super-orthogonal turbo codes (SOTC) was proposed. A different approach was taken in [4], where a class of nested rate-compatible convolutional codes, with maximum free distance (MFD), was derived.

An important consideration when applying coding in a communication system is the performance vs. implementation complexity. For the two classes of codes mentioned above, we investigate this relationship and compare them to the previously reported SOCCs. This is done by combining estimates of arithmetic decoding complexity with results from error-performance simulations. As well as arithmetic complexity, memory requirement is an important implementation issue. However, due to space limitations we here focus on the arithmetic complexity part of the different decoding schemes. Three different algorithms providing soft information used for iterative decoding of turbo codes are investigated: log-MAP [5], max-log-MAP and SOVA [6][7].

The paper is organized as follows: First we give a short introduction to low-rate codes in Section II. Section III contains both derivations of estimates of arithmetic decoding complexity and a brief description of the soft decision algorithms used for iterative decoding. In Section IV, the error-performance simulations are described and performance vs. complexity is compared for a number of codes from the investigated classes.

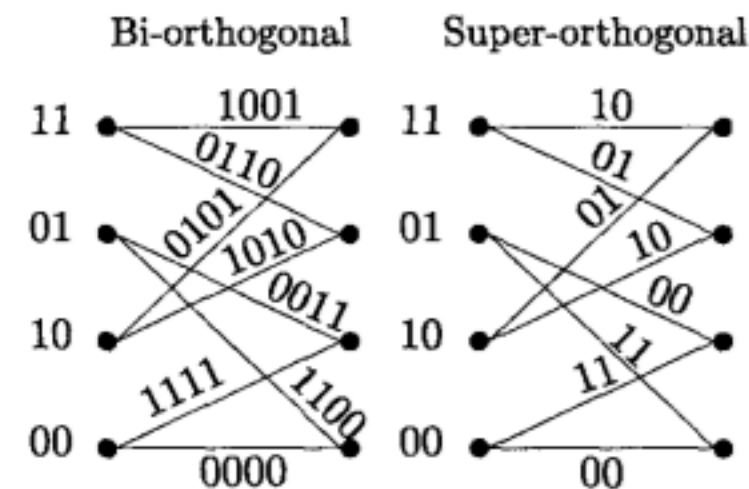


Fig. 1. Trellises of bi- and super-orthogonal convolutional codes.

II. LOW RATE CODES

A. Super-orthogonal Convolutional Codes (SOCCs)

Orthogonal block codes are known to perform well on very noisy channels. In [8] a method to find orthogonal convolutional codes having similar properties was presented. However, orthogonal convolutional codes imply a large bandwidth expansion. Several related coding schemes with good distance spectrum, but less bandwidth requirements, have been proposed in [9][3][10]. These include the SOCCs which are used in the SOTCs.

An SOCC with memory m and degree d has 2^{m-d} distinct transition sequences in the trellis and a code rate r equal to $2^{-(m-d-1)}$. The mapping rule guarantees that transition sequences emerging from, and reemerging to, each state are antipodal. The transition sequences are obtained from a Hadamard matrix H_M ($M = 2^{m-d-1}$), defined according to

$$H_1 = [0] \text{ and } H_{i+1} = \begin{bmatrix} H_i & H_i \\ H_i & \bar{H}_i \end{bmatrix}.$$

Both the Hadamard sequences H_M and their complements \bar{H}_M are mapped on the 2^{m+1} transitions of the trellis. The trellis of an SOCC of degree 0 and memory equal to 2 is shown in Figure 1.

The transition sequences of another low-rate coding scheme, the bi-orthogonal convolutional code (BOCC) described in [9] are also obtained from the Hadamard matrix, cf. Figure 1.

B. Maximum Free Distance Convolutional Codes (MFDs)

The MFD codes studied belong to a family of rate-compatible convolutional codes [4] with maximum free distance. The structure of these codes is similar to ordinary convolutional codes, cf. Figure 2. Given a specific memory length and code rate, the generator polynomials yielding maximum free distance are derived from a rate 1/4 mother code by nested code search.

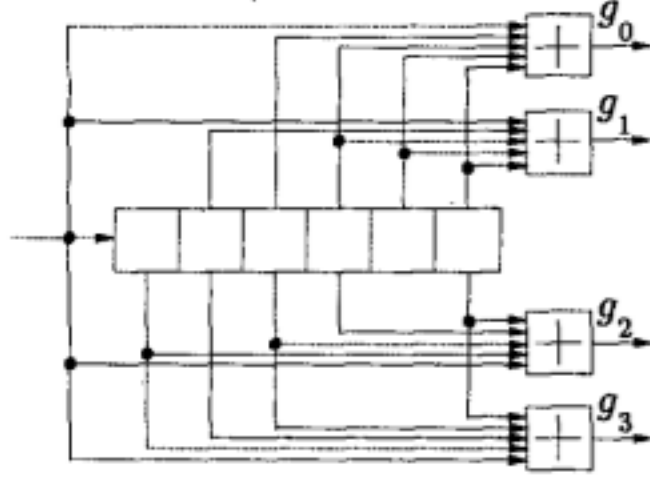


Fig. 2. Maximum free distance convolutional code.

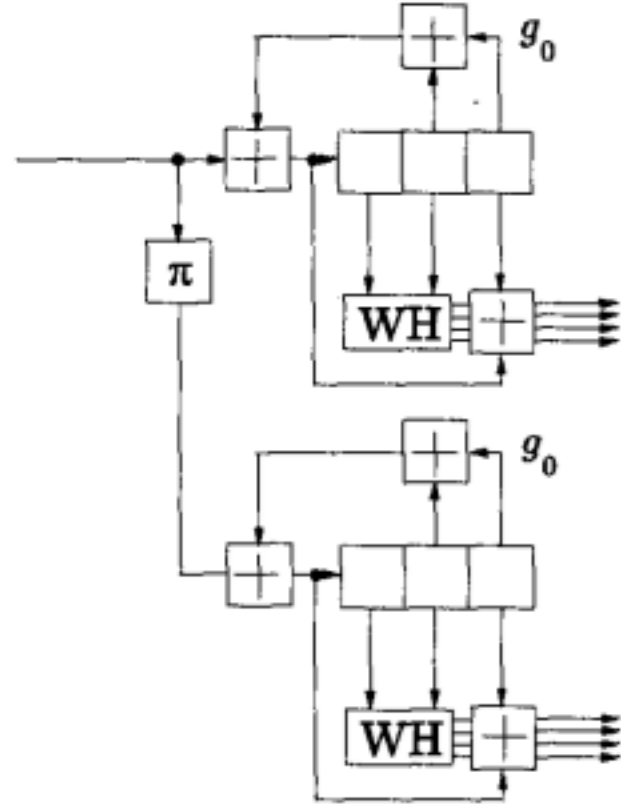


Fig. 3. Structure of super-orthogonal turbo encoder.

An important characteristic of the nested convolutional codes is the reuse of generator polynomials to achieve low rates. An MFD convolutional code of rate $1/n$ can thus be generated by $a \leq n$ different generator polynomials. This repeating structure can be used to decrease the encoding-, as well as the decoding complexity of the code. Complexity is discussed further in Section IV.

The MFD convolutional codes, as well as the SOCCs described above, are antipodal, *i.e.* they have the property that the branches emerging from the same state, or remerging to the same state, are antipodal.

C. Super-orthogonal Turbo Codes (SOTCs)

The SOTC coding schemes studied in this paper consist of two component encoders with one intermediate interleaver, as shown in Figure 3. Two SOCCs are used as component codes. The parity sequences of the component codes are generated by modulo-two adding the first and the last bits of the register to an orthogonal Walsh-Hadamard (WH) sequence. Similar to the turbo codes studied in [1], the component codes here are non-systematic.

Low-rate turbo codes can be achieved by different strategies; through 1) the use of low-rate component encoders, or 2) extending the structure in Figure 3 with additional component encoders. The first strategy is chosen in this investigation where the SOCCs, discussed in Section II-A, are taken as an ad-hoc approach to the design of low-rate component encoders for turbo codes. The reason for not addressing the second strategy is the expected increase in decoding complexity when using many component encoders.

III. DECODING

A. Decoding of Convolutional Codes

The structure of both the SOCCs and the MFD convolutional codes enables the metric calculations to be done at reduced complexity. In the SOCCs, the transition sequences are equal, antipodal or orthogonal. This fact enables the use of the Walsh-Hadamard (WH) transform [11] for metric calculation.

Considering Figure 1, the trellis of a SOCC of rate $\frac{1}{n}$ is seen to have only $2 \times n$ different transition sequences. As these are derived from the Hadamard matrix and its complement, the metric can be calculated in $n \times \log_2 n$ additions and n multiplications with -1 . Since the rate of a SOCC of degree zero is directly related to the memory of the code ($m = \log_2 n + 1$) the metric calculation-complexity can be done in $2^{m-1} (m-1)$ additions and 2^{m-1} multiplications with -1 per trellis stage.

For the MFD codes, the number of generator polynomials a , defined as the number of unique generator polynomials used in the code, and the memory determine the decoding complexity. If there is reuse of generator polynomials, the spreading factor n becomes larger than a . For every transition sequence in the trellis, the last $n - a$ bits are repetitions of earlier parity bits.

Summing the signal samples corresponding to the same generator polynomial (de-spreading), the correlation operation only have to be performed once per unique generator polynomial. The metric increment to be calculated by the Viterbi algorithm is given in Equation 1. The transmitted parity symbols of path m and the received signal samples at trellis step k are denoted $b_k^{(m)}(i)$ and $r_k(i)$ respectively.

$$\begin{aligned} \Delta_k^{(m)} &= \sum_{i=1}^n b_k^{(m)}(i) r_k(i) \\ &= \sum_{i=1}^a b_k^{(m)}(i) \sum_{\substack{j \in [1, n] \\ b_k^{(m)}(j) = b_k^{(m)}(i)}} r_k(j) \quad (1) \end{aligned}$$

This summation requires $n - a$ additions per trellis stage for each repeated generator polynomial. Another a additions are in general required per trellis transition to complete the metric calculation. However, since the MFD codes are antipodal, each transition sequence and its complement occur twice in the trellis. This implies $a/4$ additions and $1/4$ multiplications with -1 per transition, rather than a additions.

The metric-calculation-complexity per trellis stage thus becomes $2^{m+1} (a+1)/4 + n - a$ additions.

For both the SOCCs and the MFD codes at least another 2^{m+1} additions and 2^m max operations per trellis step are needed to complete the Add Compare Select (ACS) operations. Selection-, normalization- and traceback complexities are neglected in the complexity analysis here.

B. Decoding of Super-orthogonal Turbo codes (SOTCs)

The parallel concatenated structure of the turbo code permits it to be decoded in parts, by two or more decoders. Several different types of a posteriori

probability estimators, with different degree of complexity, such as the maximum a posteriori probability (MAP), the suboptimal max-log-MAP- and the soft output Viterbi algorithm (SOVA) can be used to provide soft information in the turbo decoder.

The Log-MAP algorithm considers all codewords in order to determine the a posteriori probability of a transmitted symbol. In the turbo decoder the extrinsic information from one soft output decoder is fed to the other decoder as a priori information about the transmitted symbols. The introduction of a priori information in the Log-MAP algorithm requires addition of the extrinsic information to all correlation metrics that correspond to transmitted information symbols equal to one. The increase of complexity is thus 2^m additions per decoding operation.

Further, the correlation metric of the received samples is only calculated once per transition sequence during the first iteration. Following the discussion in section III-A the WH transform can be used to efficiently calculate the correlation metrics also for the SOTC. We conclude that the correlation complexity may be neglected in the analysis.

By transferring the MAP algorithm to the logarithm domain, multiplications can be replaced by additions. However, this instead causes the need to calculate sums of exponentials representing probabilities of states or transitions. The summation of these makes up a substantial part of the complexity of the Log-MAP algorithm. A significant decrease of decoding complexity can be achieved using the approximation

$$\ln(e^{p_1} + \dots + e^{p_n}) \approx \max(p_1, \dots, p_n), \quad (2)$$

where p_i represents the logarithm of the probability of a state or a transition in the trellis.

The resulting decoding algorithm is the Max-Log-MAP. It modifies the Log-MAP to only take into account the most likely path having a particular symbol at position k and the most likely path having the opposite symbol at the same position when estimating the a posteriori probability of a transmitted symbol. The SOVA is a modified Viterbi algorithm. It is similar to the Max-Log-MAP algorithm in that it always finds the maximum likelihood (ML) path having a particular symbol at position k . The difference is that the SOVA not always uses the most likely path having the opposite symbol at the same position to evaluate the estimated a posteriori probability.

When the SOVA is used in a turbo decoder, the extrinsic information $L_{e,k}$ from the preceding decoder is treated as a priori information. The a priori probabilities are introduced in the metric of path m according to

$$M_k^{(m)} = M_{k-1}^{(m)} + u_k^{(m)} L_{e,k} + \frac{2a_k}{\sigma^2} \sum_{i=1}^n b_k^{(m)}(i) r_k(i), \quad (3)$$

where the sign of the information symbol transmitted is denoted $u_k^{(m)}$.

The SOVA investigated here uses the updating rule proposed in [6] to achieve the reliability measures. Depending on the size of the decoding window, δ , and the number of trellis steps where the information

Dec.	add	$\times(-1)$	max	lookup
L-M	$7 \times 2^{m+1}$	2^{m+1}	2^{m+1}	2^{m+2}
M-L-M	2^{m+3}	1	2^{m+2}	
SOVA	$3 \times 2^m + \delta/2$	2^m	$2^m + \delta$	

TABLE I
ELEMENTARY OPERATIONS REQUIRED BY THE LOG-MAP,
MAX-LOG-MAP AND SOVA SOFT DECODING ALGORITHMS

symbols of the most likely path differs from the information symbols of the path it is merging with, a different number of additions is required. For simplicity we have assumed here that the paths differ in $\delta/2$ positions on the average. The used decoding window is equal to six times the constraint length of the constituent codes, i.e. $\delta = 6(m+1)$. Without going into further detail, the estimated number of elementary operations required in the Log-MAP, Max-Log-MAP and SOVA decoders are given in Table I.

IV. COMPLEXITY AND PERFORMANCE

A. Arithmetic Decoding Complexity

A comparison of the arithmetic complexity of different choices of codes and decoding algorithms require an estimate of the implementation complexity of elementary operations. However, there are several different measures of complexity as well as many different implementation architectures. We chose to compare the different low-rate schemes as if the decoders were implemented on a standard DSP. We estimate the arithmetic decoding complexity of the addition-, multiplication with -1 , max- and lookup operations to be approximately equal, represented here as one complexity unit. Furthermore, on-chip data- and program storage is assumed. Thus additional complexity from external access is neglected.

For the turbo decoding process we need to consider that I decoding iterations are required. Further, each iteration consists of two consecutive decoding operations with intermediate interleaving. According to Section III-B, the transfer of extrinsic information to one decoder requires 2^m additions. In total this increases the complexity with 2^{m+1} additions per iteration.

As the interleaver used here is small we assume that it could be stored on the chip. Thus interleaving of one bit would require one lookup operation. During each iteration two interleaving operations are performed that yield additional complexity equal to two times the block length N for each iteration. An overview of the estimated arithmetic decoding complexity for the evaluated schemes, derived from the results in Table I is given in Table II.

The number of unique generator polynomials, a , used in each of the investigated MFD codes is shown in Table III.

B. Error-performance Simulations

Error-performance simulations on the AWGN channel are done for a block length of 100 bits. The m last bits are devoted to trellis termination. The MFD codes with memory m needs m trailing zeros to bring the encoder back to the zero state. We have chosen to only terminate the first component encoder

Codec	Decoding Complexity
SOCC, Viterbi	$2^{m-1}(m+8)$
MFD, Viterbi	$2^{m-1}(a+9) + n - a$
SOTC, L-M	$(23 \times 2^{m+1} + 2 \times N) \times I$
SOTC, M-L-M	$(13 \times 2^{m+1} + 2 \times N) \times I$
SOTC, SOVA	$(6 \times 2^{m+1} + 3 \times \delta + 2 \times N) \times I$

TABLE II
ESTIMATED ARITHMETIC DECODING COMPLEXITY FOR DSP
IMPLEMENTATION OF LOW-RATE DECODING SCHEMES

m	6	6	6	7	7	7	8	8	9	9
n	8	16	32	8	16	32	8	16	8	16
a	7	11	12	7	9	10	8	9	8	15

TABLE III
NUMBER OF UNIQUE GENERATOR POLYNOMIALS FOR MAXIMUM
FREE DISTANCE CODES OF DIFFERENT MEMORY m AND
SPREADING n .

in the SOTCs. The trellis termination implies a small rate loss, by a factor $(100 - m)/100$. The rate loss which appear as a loss of power in the simulations causes the comparison of codes of some what different data rates. In this analysis the differences are small and are considered as negligible. For shorter block lengths however, the differences in rate need to be reconsidered.

The interleaver used for the SOTCs is the correlation-designed interleaver introduced in [12]. The feedback polynomials for the SOTCs of memory 3 and 4 were 13 and 31 (octal form).

C. Evaluation

In this evaluation we focus on code rates of 1/8 and 1/16. For MFD codes we have the freedom to chose between different memory lengths, and have chosen $m = 6, 7, 8$, and 9. On the other hand, for the SOCCs as well as the SOTCs the memory is restricted to certain values determined by the code rate. By choosing SOCCs with long memory and puncturing them to different degrees according to Section II-A it is possible to achieve SOCCs of higher rate while maintaining the long memory. However, as seen in Figure 4, this strategy gives a significant performance degradation compared to the non-punctured SOCC and the corresponding MFD code. The performance of the MFD codes of memory 6 and rate 1/16 and 1/8 are not shown in Figure 4 but only a slight degradation in performance of the MFD code with memory 6 and rate 1/32 is suffered when increasing the rate to 1/16 and 1/8. This indicates that the SOCCs should be restricted to have degree zero to be competitive to the MFD codes. The non-punctured SOCCs appear to be a good alternative to the MFD codes if decoding complexity is of higher concern than rate flexibility.

Using the complexity estimates in Section IV we have calculated the arithmetic decoding complexity for a number of different low-rate codes with different error performance. MFD codes of memory 6, 7, 8 and 9 are compared to SOTCs with 2, 4, 8 and 16 decoding iterations. The performance vs. complexity of the different coding schemes is summarized in Figure 5 and 6 showing the required SNR and complexity at

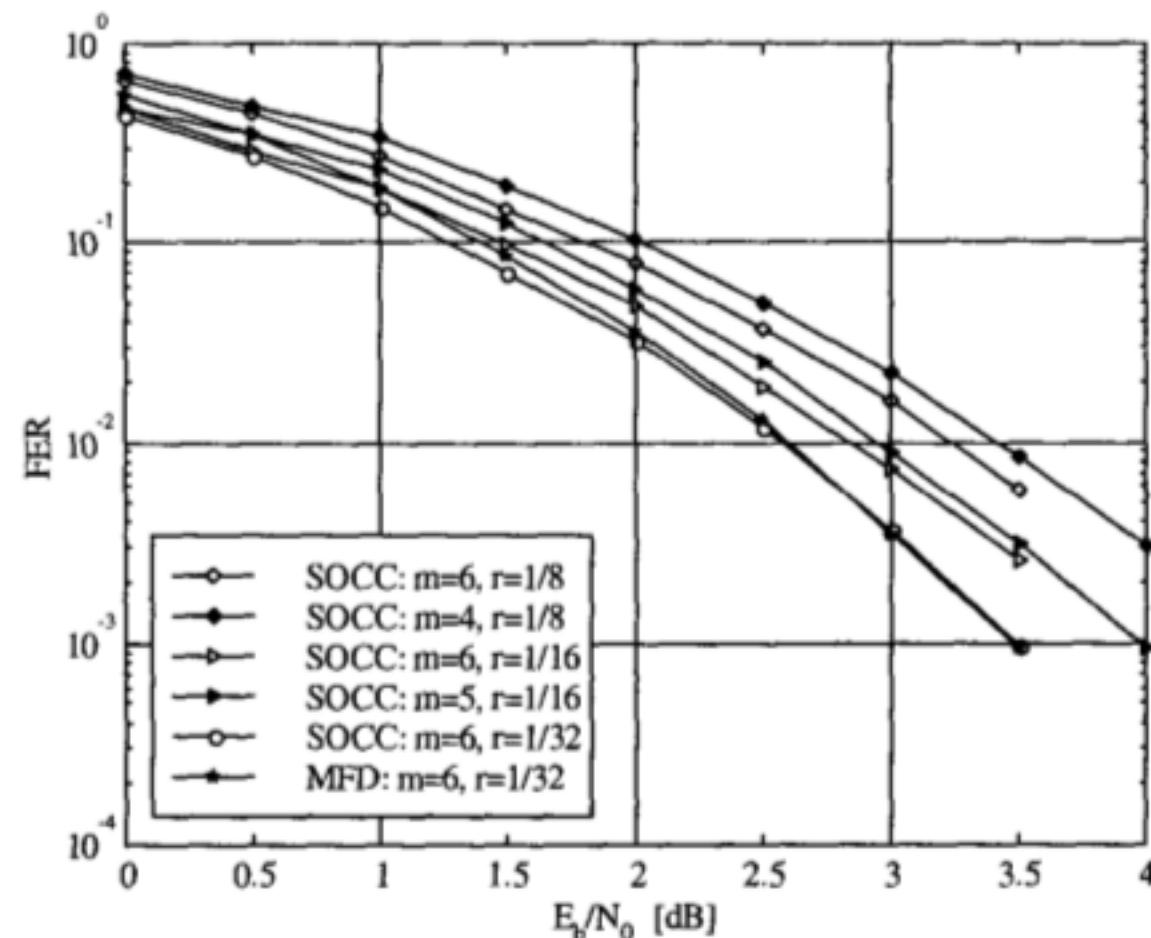


Fig. 4. Frame-error rate of MFD code of memory 6 and rate 1/32 compared to punctured and non-punctured SOCCs of memory 6, 5 and 4 and rate 1/32, 1/16 and 1/8.

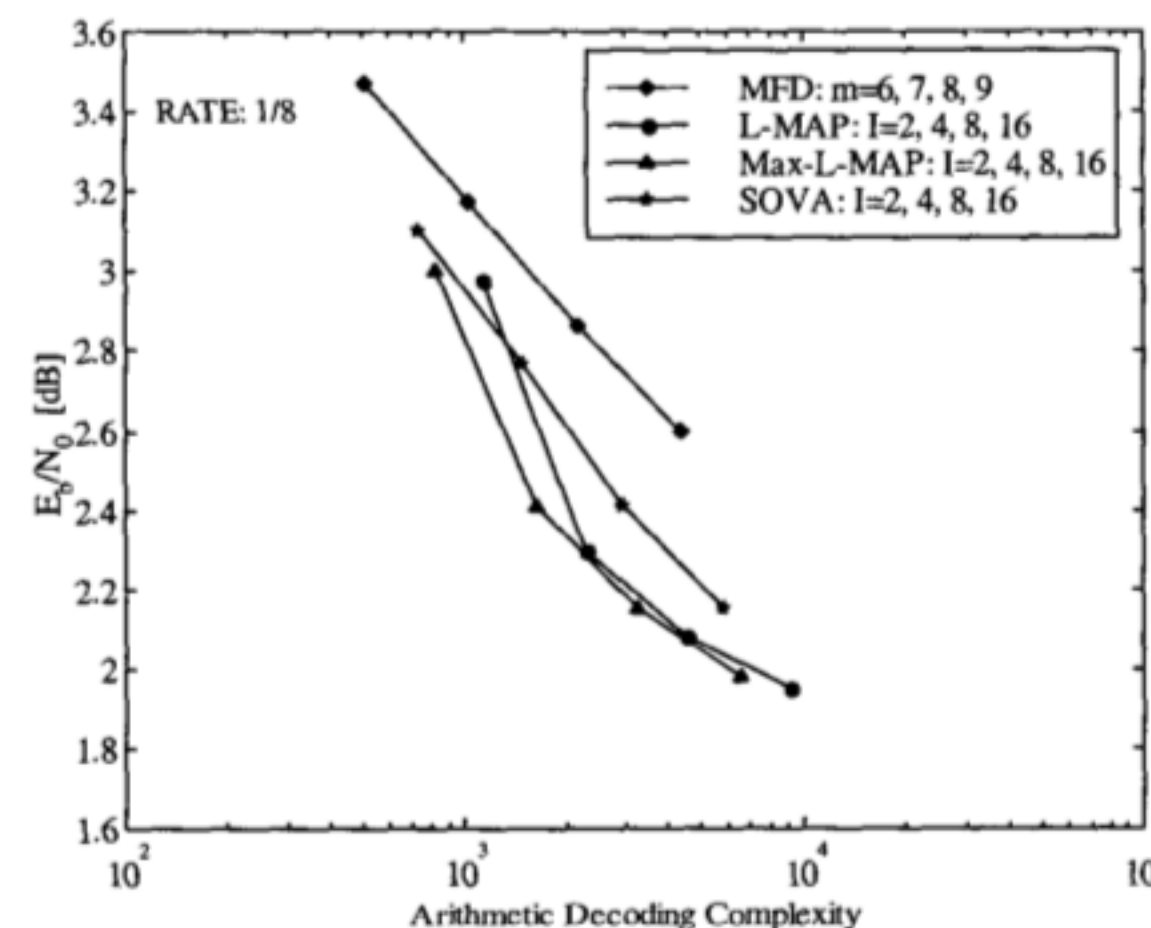


Fig. 5. Required E_b/N_0 vs. decoding complexity at FER 10^{-3} and rate 1/8 for MFD codes of memory 6, 7, 8 and 9 and SOTCs of memory 3 with 2, 4, 8 and 16 decoding iterations.

frame-error rate (FER) of 10^{-3} for codes of rate 1/8 and 1/16 respectively.

The difference in error performance between the Max-Log-MAP and the Log-MAP algorithms is seen to be small. However the Max-Log-MAP is much less complex to implement. For the investigated codes, the SOTCs yield better frame-error performance at lower complexity than the MFD codes regardless of the soft decision algorithm used in the iterative decoding. The relative improvement in performance resulting from lowering the rate of the MFD codes depends on the number of new generator polynomials introduced. The number of unique generator polynomials is also closely related to the complexity as follows from Table II. For the codes where a changes little the decrease in rate is mainly achieved through spreading. Thus the increase in complexity is small, as well as the improvement in performance.

An example of the relative bit- and frame error performance of the MFD codes and the SOTCs iteratively decoded using the Max-Log-MAP algorithm is shown in Figure 7 and Figure 8. As previously has been observed the SOTCs have better frame-error

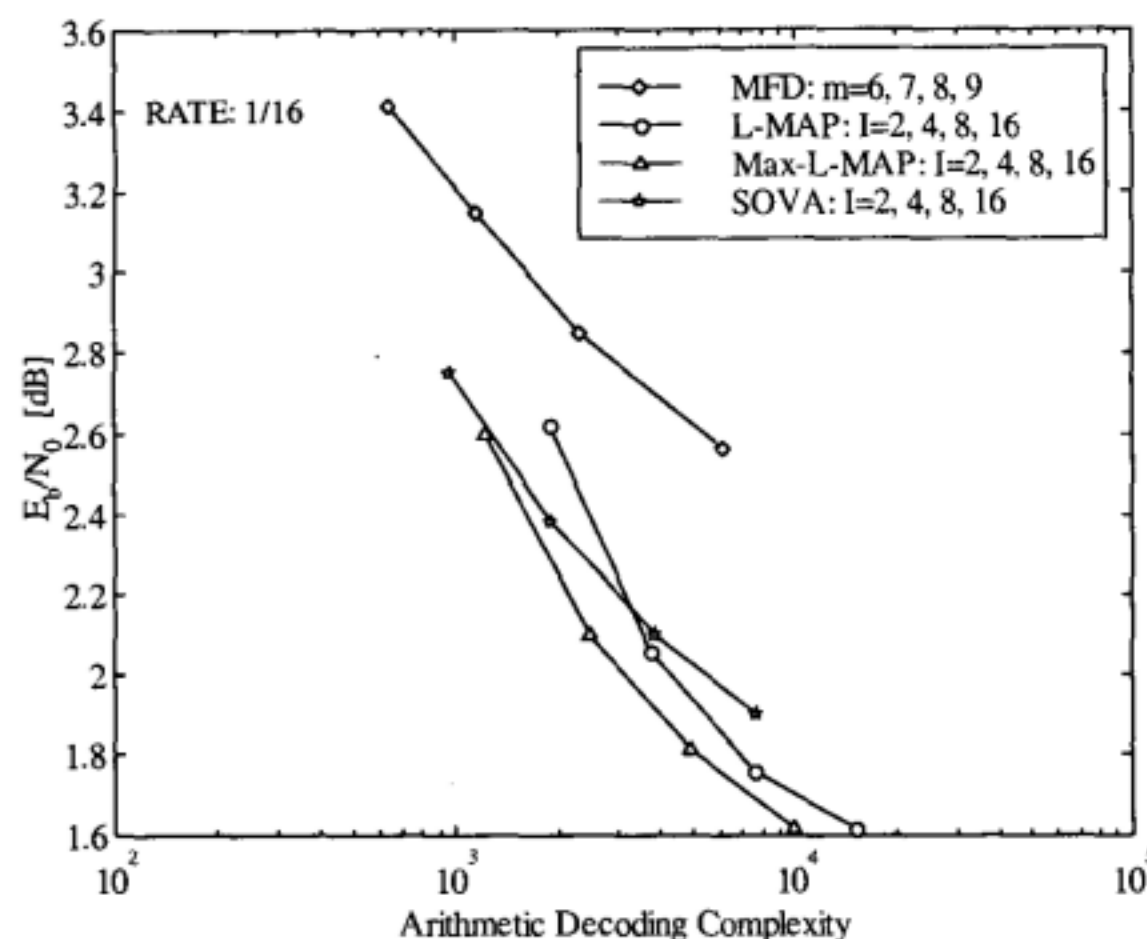


Fig. 6. Required E_b/N_0 vs. decoding complexity at FER 10^{-3} and rate 1/16 for MFD codes of memory 6, 7, 8 and 9 and SOTCs of memory 3 with 2, 4, 8 and 16 decoding iterations.

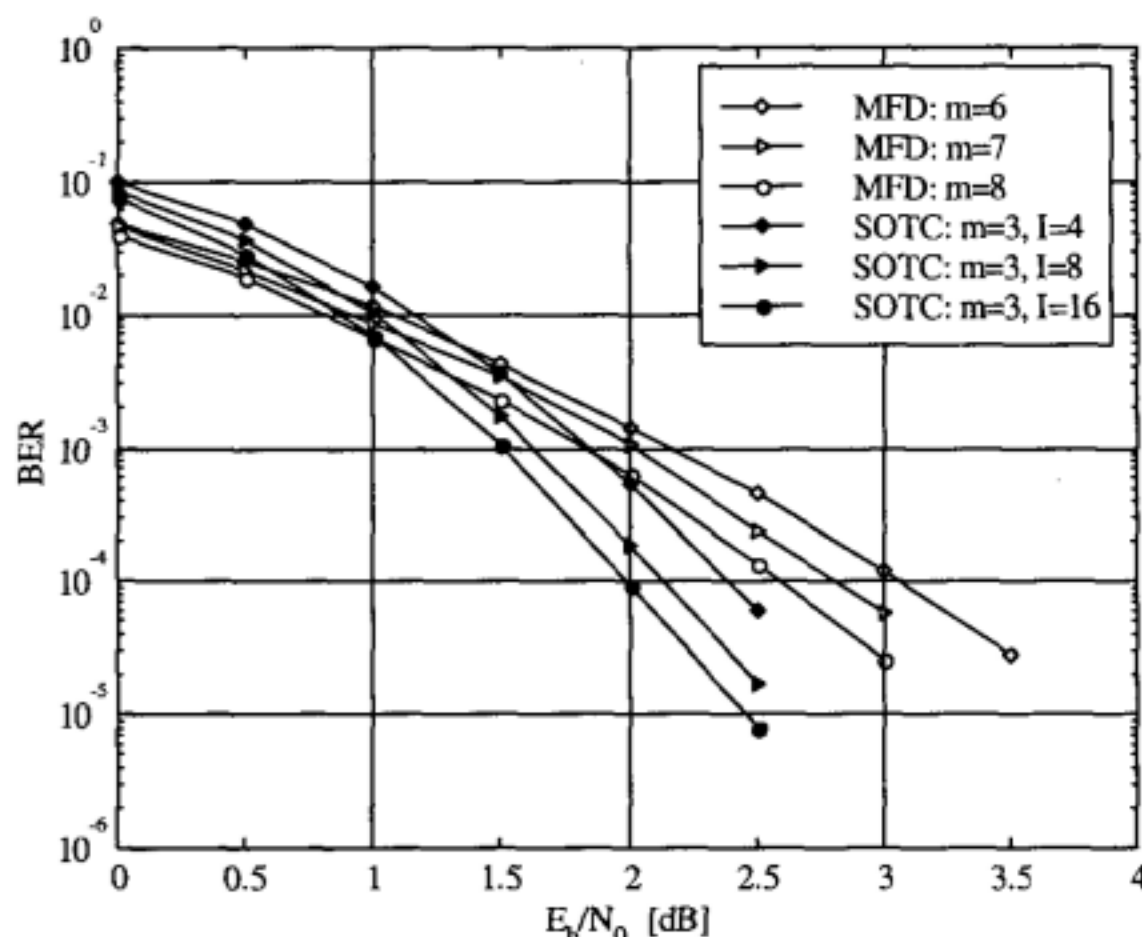


Fig. 7. Bit-error performance of rate-1/8 MFD codes of memory 6,7 and 8 and SOTCs of memory 3 decoded with 4, 8 and 16 iterations I , using the Max-Log-Map algorithm.

performance the the MFD codes. This is true also for the bit-error performance. For very low SNRs however, the MFD codes show better bit-error performance than the SOTCs.

V. CONCLUSIONS

We have presented a performance vs. arithmetic complexity evaluation of three low-rate coding schemes. An equally important source of complexity is their respective memory requirements. However, due to space limitations this issue is not considered in this investigation.

The SOCCs offer performance comparable to that of the MFD codes, requiring lower decoding complexity. On the other hand, the existence of good SOCCs is restricted to a small number of rates while the MFD codes give high performance for a multitude of rates.

For the investigated codes the SOTCs yield higher performance to lower complexity at FER equal to 10^{-3} than the MFD codes. However, as well as the SOCCs the SOTCs are limited to few rates determined by the memory of the component codes. Due to the limited number of codes evaluated here as well

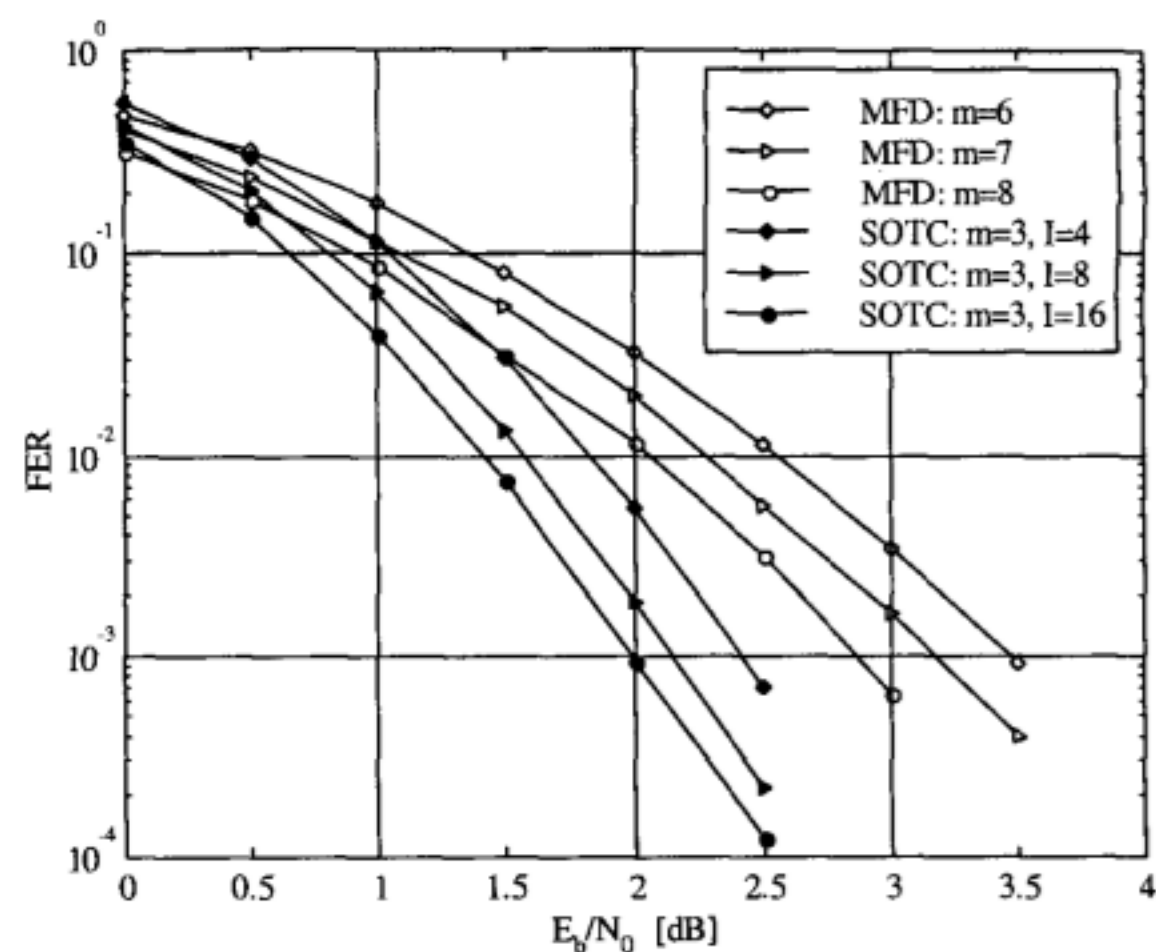


Fig. 8. Frame-error performance of rate-1/8 MFD codes of memory 6,7 and 8 and SOTCs of memory 3 decoded with 4, 8 and 16 iterations I , using the Max-Log-Map algorithm.

as the close relation between arithmetic complexity requirements and memory consumption, the conclusions should not be taken too far, but the results are interesting enough to prompt an extended investigation.

REFERENCES

- [1] P. Komulainen and K. Pehkonen, "Performance evaluation of superorthogonal turbo codes in AWGN and flat rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 196-205, Feb. 1998.
- [2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Transactions on Communications*, vol. 44, pp. 1261-1271, Oct. 1996.
- [3] E. Zehavi and A. Viterbi, "On new classes of orthogonal convolutional codes," *Communication, Control and Signal Processing*, pp. 257-263, 1990.
- [4] P. Frenger, P. Orten, and T. Ottosson, "Code-spread CDMA using maximum free distance low-rate convolutional codes," *submitted to IEEE Transactions on Communications*, 1998.
- [5] 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. 20, pp. 284-287, March 1974.
- [6] G. Battail, "Pondération des symboles décodés par l'algorithme de viterbi," *Ann. Telecommun.*, vol. 42, pp. 31-38, Jan 1987.
- [7] J. Hagenauer and P. Hoeher, "A viterbi algorithm with soft decision outputs and its applications," in *Proc. Globecom, Dallas*, pp. 1680-1686, Nov 1989.
- [8] A. J. Viterbi, "Orthogonal tree codes for communication in the presence of white noise," *IEEE Transactions on Communications*, vol. 15, pp. 238-242, April 1967.
- [9] T. Maseng and B. Rislow, "Codes for very noisy channels," *Proc. 4th Joint Swedish-Soviet Int. Workshop on Information Theory, Gotland, Sweden*, pp. 43-48, 1989.
- [10] K. Rikkinen, "Comparison of very low rate coding methods for CDMA radio communications system," *Proc. IEEE Int. Symp. Spread Spectrum Techniques Appl. Oulu, Finland*, vol. 1, pp. 268-272, July 1994.
- [11] D. F. Elliot and K. R. Rao, *Fast Transforms, Algorithms, Analyses, Applications*. Academic Press, 1982.
- [12] J. Hokfelt, O. Edfors, and T. Maseng, "Interleaver design for turbo codes based on the performance of iterative decoding," in *IEEE International Conference on Communications*, (Vancouver, BC, Canada), June 1999.