Autocorrelation properties of channel encoded sequences — applicability to blind equalization

J. Mannerkoski* and V. Koivunen, Senior Member, IEEE

Helsinki University of Technology

Signal Processing Laboratory

P.O. Box 3000, FIN-02015 Helsinki, Finland

fax: +358-9-460 224

email: {jukka.mannerkoski, visa.koivunen}@hut.fi

EDICS: SP 3-CEQU

Abstract

Many blind channel equalization/identification algorithms, e.g. [1, 8, 11, 14] are derived assuming the transmitted information sequence to be white. In practical communication systems, redundancy is added to the source sequence in order to detect and correct symbol errors in the receiver. It is not obvious how channel encoding will affect the assumption of whiteness. The autocorrelation function of some commonly used channel codes is analyzed in order to study the validity of assumptions used in blind equalization. The codes are presented in terms of a Markov model, for which the autocorrelation is analytically expressed. The various encoded sequences are used in the prediction-error filtering based blind equalizer of [1] and the performance is empirically compared to the case of unencoded data. A blind equalization example using a practical GSM speech encoder combined with a convolutional channel encoder is also given [1].

[1]This work was supported by the Academy of Finland. Some results of this paper were presented in [10].

1 Introduction

A blind channel equalization/identification algorithm attempts to mitigate the effects of the Inter-Symbol Interference (ISI) introduced by the effective communication channel. Because no known training signal is available, some statistical criteria need to be used. A number of blind equalizers, e.g. [1, 8, 11, 14] are derived assuming that the transmitted sequence $x(n)$ is white, i.e. the autocorrelation function $r_x(j) = \mathrm{E}[x(n)x^*(n-j)] = \delta(j)$. This is a reasonable assumption for efficiently source encoded data.

In a channel encoder, the compressed source sequence is augmented with parity bits for the purpose of error detection and correction. This work is motivated by a common belief in the blind equalization community that channel encoding cannot be used with equalizers such as [1, 8, 11, 14] due to a violation of the whiteness assumption. The emphasis is in studying the practicality of the assumption often used in deriving blind equalizers and not in contributing to the theory of codes. In this paper, we study how commonly used channel codes shape the autocorrelation function of the transmitted sequence and how the performance of the equalizers is consequently affected. A Markov model is used for the encoded sequences, because it provides a flexible tool for analyzing a wide variety of codes [2].

In linear block encoding, a block of M L-ary symbols from the source sequence $a(n)$ is represented by a length-N L-ary code word $\mathbf{c}_i, i = 0, \ldots, K-1$, where the number of all code words $K = L^M$. The L-ary alphabet can be mapped to k bits, if $L = 2^k$, resulting in a binary code. A block of a consistent code consists of two parts, the M information symbols unchanged appended with a parity check pattern that is computed from the M symbols. The code word can be generated by a linear shift register, as in Fig. 4, where $M = 4$ and $N = 7$ [12]. After the information symbols have been fed through as they are, the switch is turned to the parity generating part and the parity symbols are then concatenated to the information bits. In this paper, the bits of the binary code word are mapped to the transmitted sequence $x(n)$ as $0 \to -1$ and $1 \to +1$.

In convolutional encoding, L-ary source symbols are processed in a serial manner in the encoder. The L-ary alphabet can be mapped to k bits, if $L = 2^k$, resulting in a binary code, where k bits at a time enter the encoder. For every k bits of the source $a(n)$, the encoder outputs N bits comprising the code word $\mathbf{c}_i, i = 0, \ldots, K-1$. The code word can be generated by a linear shift register, consisting of a number of k-bit stages. Fig. 1 presents a binary code with three one-bit stages, $k = 1$ and code word length $N = 3$ (code C_1) [12]. Figs. 2 and 3 show a quaternary code ($k = 2$) with two stages and code word length $N = 3$ (code C_2) and $N = 4$ (code C_3) respectively. The bits of the code word are read serially from the N outputs and mapped to the transmitted sequence $x(n)$, resulting in a code rate of k/N.

The contribution of this paper is to show that assuming whiteness of the transmitted sequence is not unrealistic when deriving blind equalization algorithms for practical communication systems. The paper is organized as follows: Section 2 contains an analytical expression for the autocorrelation function of a Markov modeled channel code. In section 3, the analysis is applied to linear block codes and in section 4 to convolutional codes. The required Markov models are defined and analytical and experimental autocorrelation functions are presented. The performance of a prediction-based blind equalizer [1] for channel encoded sequences is investigated in simulation in section 5. The effect of a practical GSM speech encoder on the performance is also studied. Section 6 concludes the paper.

2 Autocorrelation function of a channel encoded sequence

In order to analyze the autocorrelation function of the encoded sequence, we model it as a Markov chain, as given in [3, 6, 2]. We consider a *Mealy machine*, so that the output code word is a function of the current state only. When the state at time n is $s_i, i = 0, \ldots, K-1$, the corresponding length-N output code word is $\mathbf{c}_i = [c_i(0), \ldots, c_i(N-1)]$. The state s_j at time $n+1$ — and the output code word — depends on the current state and k new source bits. The mapping from state s_i to code word $\mathbf{c}_i$ is unique, but not necessarily vice versa. The elements p_{ij} of the $(K \times K)$ state transition matrix Π are the probabilities of the transition from state s_i to state s_j. We will also need a vector $\mathbf{p} = [p_0, \ldots, p_{K-1}]$

containing the steady-state probabilities p_i of states s_i. It can be obtained as the unit-norm eigen vector of Π corresponding to the eigen value equal to unity [3]. Complete statistics of the Markov chain are thus provided by Π. When Π and all the possible code words are defined, the autocorrelation function of the encoded sequence can be computed.

First, let us construct a matrix $\mathbf{C}$ containing the K code words $\mathbf{c}_i$ corresponding to the K states s_i

$$\mathbf{C} = \begin{bmatrix} c_0(0) & \dots & c_0(N-1) \\ \vdots & \ddots & \vdots \\ c_{K-1}(0) & \dots & c_{K-1}(N-1) \end{bmatrix} \tag{1}$$

and define a column vector $\mathbf{c}(i) = [c_0(i), \dots, c_{K-1}(i)]^T$. The discrete autocorrelation function of the encoded sequence is then [2]

$$r_x(j) = \sum_k \sum_i \mathbf{c}^H(i)\mathbf{R}(k)\mathbf{c}(i+j-Nk) \tag{2}$$

where the range of i and k is defined by $(0 \leq i \leq N-1 \cap 0 \leq i+j-Nk \leq N-1)$. The state correlation matrix is defined as

$$\mathbf{R}(k) = \begin{cases} \mathbf{D}\Pi^k & , k \geq 0 \\ (\Pi^H)^{-k}\mathbf{D} & , k < 0 \end{cases} \tag{3}$$

where $\mathbf{D}$ is $diag(p_0, \dots, p_{K-1})$. It should be noted that the encoded sequences are cyclo-stationary and their second order statistics not completely determined by $r_x(j)$ [7].

3 Analysis of linear block codes

In the previous section, we presented a generic method for computing the autocorrelation function of a channel encoded sequence. In this section, we concentrate on linear block codes. Since we are employing a Mealy machine, the Markov model for a linearly block encoded sequence is trivial. For an L-ary code, the number of states $K = L^M$. The transition probability from state s_i to any of the K states s_j — as well as every element in the $(K \times K)$-matrix Π — equals $1/K$:

$$\Pi = \begin{bmatrix} 1/K & \dots & 1/K \\ & \ddots & \\ 1/K & \dots & 1/K \end{bmatrix} \tag{4}$$

Also, the steady state probabilities $p_i = 1/K$, for all $i = 0, \dots, K-1$, so that

$$\mathbf{D} = diag(1/K, \dots, 1/K) \tag{5}$$

Consequently, $\mathbf{R}(k) = \mathbf{D}$ for $k = 0$ and $\mathbf{R}(k) = \mathbf{D}\Pi$ otherwise. By constructing the code word matrix (1), we can compute the autocorrelation function $r_x(j)$ from (2). The order of the row vectors of $\mathbf{C}$ is in practice arbitrary, because all the elements in Π are equal.

3.1 Binary codes

A practical representation of a linear block code is a generator matrix. In this paper, we first employ the binary $M = 4$ and $N = 7$ block code of Fig. 4 (code B_1). At time instant n_0, one of $K = 2^4$ code words $\mathbf{c}_i$ is obtained from a block $\mathbf{a} = [a(n_0), \dots, a(n_0 + M - 1)]$ of the information sequence as

$$\mathbf{c}_i = \mathbf{a}\mathbf{G} \tag{6}$$

where the generator matrix [12]

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{7}$$

This is a systematic Hamming code, with $\mathbf{G}$ consisting of an identity matrix and a parity sequence generating part $\mathbf{G} = [\mathbf{I}|\mathbf{P}]$. The number of states and code words is now $K = 2^4$. Matrix (1) can be constructed by using all permutations of the source sequence in (6). Fig. 5a shows the absolute value of the analytical $r_x(j)$ (2) of the encoded sequence $x(n)$. Clearly, $r_x(j) = \delta(j)$, implying whiteness of $x(n)$ and consequently algorithms using the whiteness assumption of the data should not suffer any performance degradation when the encoded data is used. The analytical result is verified with the corresponding sample estimated autocorrelation $\hat{r}_x(j) = 1/N_s \Sigma_{n=1}^{N_s}[x(n)x^*(n-j)]$, where $N_s = 2000$, in the same plot.

Not all block codes result in perfectly white sequences. We demonstrate this by evaluating $r_x(j)$ using all possible generator matrices of a systematic $M = 4, N = 7$ code that have unique and non-zero columns in the parity generator $\mathbf{P}$. Performing an exhaustive computer search, we can note that there are a total 2730 of such realizations in $\mathbf{P}$. Fig. 5b shows the average analytical autocorrelation function of the resulting codes. It should be noted that some of these codes may not be practical. The number of perfectly white sequences is 990.

An example of a non-white block code is a repetition code [9]. Each source symbol is repeated N times in the transmitted sequence. The generator matrix of an $N = 3$ repetition code is simply $\mathbf{G} = [1\ 1\ 1]$. There are two states s_0 and s_1 corresponding to source symbols 0 and 1. The state transmission matrix $\mathbf{Pi}$ is now a (2×2)-matrix with every element equal to 1/2. Finally, $r_x(j)$ can be computed after defining

$$\mathbf{C} = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

Fig. 5c shows that the encoded sequence is not white, which is also intuitively clear.

3.2 L-ary Reed-Solomon codes

Next, we evaluate the autocorrelation properties of Reed-Solomon (RS) codes. This family of cyclic block codes has been applied in practice in e.g. space technology [13]. The lengths of the L-ary source and code words are related by $N = L - 1 = 2^k - 1, M = 1, \dots, N-1$.

We employ a $N = 7, M = 3$ 8-QAM RS-code (or equivalently a binary $N_b = 21, M_b = 9$ code). There are now a total of $K = 8^3$ code words and states. Constructing the matrices needed in (2) and evaluating the expression for $r_x(j)$ results in the analytical autocorrelation function shown in Fig. 6a (the absolute value is shown). For a $N = 15, M = 5$ 16-QAM RS-code, there are $K = 16^5$ distinct code words. Computations on the $(K \times K)$-matrix Π are now too expensive and we settle for presenting the sample estimated autocorrelation function $\hat{r}_x(j)$ in Fig. 6b. Clearly, for both RS-codes, the whiteness assumption holds.

4 Convolutional codes

Defining the Markov model for convolutional codes is only slightly more involved than for block codes, considered in section 3. However, the result in not as general, because the elements of the state transition matrix are not identical and the code words need to be correctly sorted in (1). For code C_1, there are $K = 2^3$ possible code words and consequently 2^3 states. The register contents define the states of the Markov chain. They are assigned in Table 1, which also shows the output code word for each state. The transition probability matrix is now

$$\Pi = \begin{bmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \end{bmatrix}$$

The code word matrix (1) is constructed from the rightmost column of Table 1 by mapping $0 \to -1$ and $1 \to +1$ to obtain a zero-mean sequence. The autocorrelation can then be computed from (2). Fig. 7a shows the absolute value of $r_x(j)$ for code C_1. The whiteness assumption clearly holds. The analytical result is verified by the corresponding sample estimated autocorrelation $\hat{r}_x(j)$ given in the same plot ($N_s = 2000$).

We continue by analyzing the autocorrelation properties of codes C_2 and C_3. The states and outputs of the codes are given in Table 2. For both codes, the (16×16)-matrix Π is constructed using the following information: From all states $s_0, \ldots, s_3$, states s_0, s_4, s_8, s_{12} each follow with probability $1/4$. Similarly, $s_4, \ldots, s_7 \to s_1, s_5, s_9, s_{13}$ and $s_8, \ldots, s_{11} \to s_2, s_6, s_{10}, s_{14}$ and $s_{12}, \ldots, s_{15} \to s_3, s_7, s_{11}, s_{15}$ with probability $1/4$. Again,

Table 1: Register contents and output code words for code C_1

state s_i	reg. contents	code C_1
s_0	000	000
s_1	001	011
s_2	010	001
s_3	011	010
s_4	100	111
s_5	101	100
s_6	110	110
s_7	101	101

the code word matrix (1) is obtained from Table 2 for each code with $0 \rightarrow -1$ and $1 \rightarrow +1$. The analytical and sample estimated autocorrelation functions are shown in Figs. 7b and 7c. It can be seen that these sequences are also white.

Code C_4 is an example of a colored convolutional code. The state transition matrix Π is the same as for codes C_2 and C_3 and the code word matrix $\mathbf{C}$ is obtained from the rightmost column of Table 2, as before. Fig. 7d shows that there is correlation in the encoded sequence. Examining Fig. 8, it becomes obvious that the second code bit at time instant n_0 equals the last code bit at time instant n_0+1, which results in correlation at lags $j = \pm 4$.

5 Blind equalizer performance for channel encoded data

In this section, the performance of the prediction-error filter (PEF) based blind equalizer [1] is evaluated in simulation for unencoded and encoded channel inputs. The employed equalizer employs a Single-Input Multiple-Output (SIMO) model, i.e. fractional sampling. The linear one-step prediction error vector of the received multichannel signal $\mathbf{y}(n)$ is denoted by $\mathbf{f}(n)$ and its covariance matrix by $\mathbf{P}_f$. The zero-delay MMSE-equalized symbol

Table 2: Register contents and output code words for codes C_2 – C_4

state s_i	reg.	code C_2	code C_3	code C_4
s_0	0000	000	0000	000
s_1	0001	110	0101	001
s_2	0010	101	1010	101
s_3	0011	011	1111	100
s_4	0100	010	0110	110
s_5	0101	100	0011	111
s_6	0110	111	1100	011
s_7	0111	001	1001	010
s_8	1000	111	1011	010
s_9	1001	001	1110	011
s_{10}	1010	010	0001	111
s_{11}	1011	100	0100	110
s_{12}	1100	101	1101	100
s_{13}	1101	011	1000	101
s_{14}	1110	000	0111	001
s_{15}	1111	110	0010	000

estimate is now

$$\hat{x}(n) = \sigma_x^2 \mathbf{h}^H(0)\mathbf{P}_f^{-1}\mathbf{f}(n) \tag{8}$$

where the first channel coefficient vector $\mathbf{h}(0)$ is obtained from the eigen decomposition of $\mathbf{P}_f$ and σ_x^2 is the variance of the transmitted symbol. The predictor coefficients are obtained from the Yule-Walker equations as $\mathbf{w} = \mathbf{r}\mathbf{R}^{-1}$, where $\mathbf{R}$ is the block-Toeplitz autocorrelation matrix and $\mathbf{r}$ is the lagged autocorrelation of the received sequence. See [15, 1] for details of the SIMO model and the equalizer algorithm.

Unless otherwise stated, the unencoded sequence is white 4-QAM with unit variance. The I- and Q-components of the sequence are independently encoded. The additive noise is white and Gaussian. The SNR is defined as

$$\mathrm{SNR} := \frac{\mathrm{E}[\Sigma_{j=0}^{P-1}|y_j'(n)|^2]}{\mathrm{E}[\Sigma_{k=0}^{P-1}|v_k(n)|^2]} \tag{9}$$

where $y_j'(n)$ is the signal component of $y_j(n)$. The employed communication channel (Fig. 9) was obtained from the web-site *http://spib.rice.edu/spib/microwave.html* and was further time-decimated to length 16 as explained in [4]. At each evaluated point, 5000 test symbols were used to compute the estimate of the MSE or the symbol error rate (SER) and the procedure was repeated 100 times. No error detection/correction has been performed in order to keep the results of the encoded and unencoded sequences comparable.

Fig. 10 shows the MSE of the blind equalizer for an unencoded input sequence, codes B_1 and the $N = 3$ repetition code at different SNRs and a sample size of 1000. BPSK is used for the repetition coding in this experiment. No major difference in the performance can be noted between the white and the B_1-encoded sequences. As expected, the equalizer is not applicable to repetition coding due to the correlation at non-zero lags.

The convolutional codes C_2 and C_4 are considered next. In Fig. 11, the SER is plotted for the SNR of 10 dB as a function of the sample size. Again, the performance for code C_2, which was previously shown white, equals the performance for the unencoded sequence. The correlated code C_4 results in a significantly higher SER.

As our last experiment, we employ a speech signal shown in Fig. 12(a). A European digital cellular telecommunications system 12200 bits/s speech encoder for enhanced full rate speech traffic channels is used to compress the signal [5]. The odd and even samples of the GSM speech encoder output are used to construct a 4-QAM sequence. Fig. 12(b) shows the $\hat{r}_x(j)$ of the sequence. The result indicates that our previous assumption that efficiently source encoded sequences can be considered white holds. Next, the compressed sequence is used as a source to the convolutional encoder C_3. The $\hat{r}_x(j)$ of the channel encoded sequence is presented in Fig. 12(c). The reasonably white sequence is then used in the blind equalization experiment. We can see from Fig. 13 that the blind equalizer is feasible also in this practical situation.

6 Conclusions

The effect of channel encoding on the performance of a prediction-based blind equalizer was studied. Because the equalizer uses the assumption of a white transmitted sequence, autocorrelation properties of channel encoded data sequences were first analyzed. The Markov model based analysis was performed for various codes and the results were verified with Monte-Carlo simulations. Furthermore, the encoded sequences were used in the blind equalization and the resulting performance was compared to the ideal case of a white Bernoulli sequence. A practical case of a GSM speech encoded source sequence combined with convolutional coding was also considered. From the analytical and simulation results of this paper, it can be concluded that channel encoded sequences can be used in blind equalizer algorithms — or any other applications — that are derived assuming white input sequences. Not all codes can be used, however, and the encoding scheme should be carefully chosen. The required analysis can easily be performed employing the tools used in this paper.

References

[1] K. Abed-Meraim, E. Moulines, and P. Loubaton. Prediction error method for second-order blind identification. *IEEE Tr. Signal Proc.*, 45(3):694–705, March 1997.

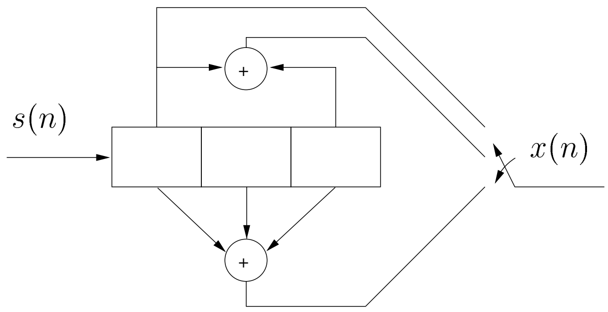

Figure 1: A shift register to produce a rate 1/3 convolutional code (Code C_1) [12].

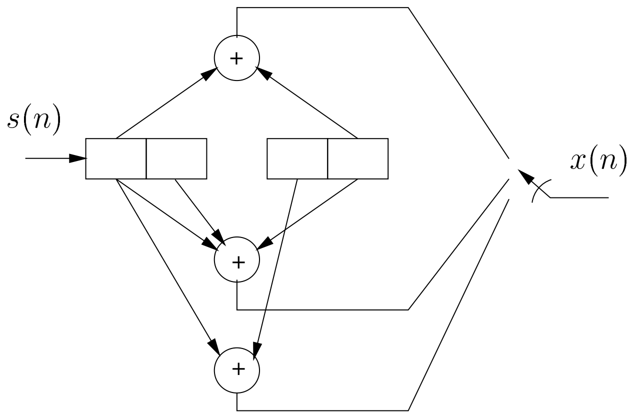

Figure 2: A shift register to produce a rate 2/3 convolutional code (Code C_2) [12].

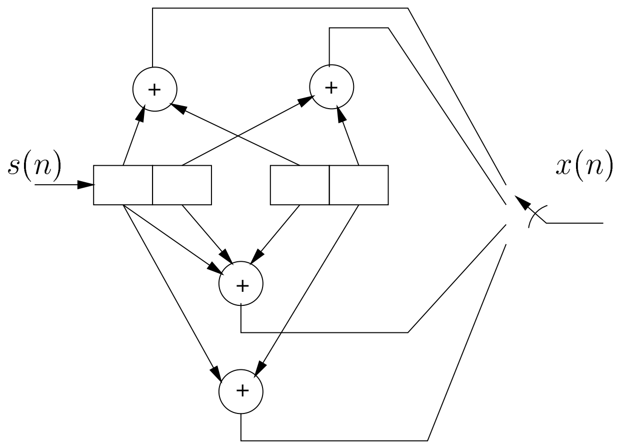

Figure 3: A shift register to produce a rate 1/2 convolutional code (Code C_3) [12].

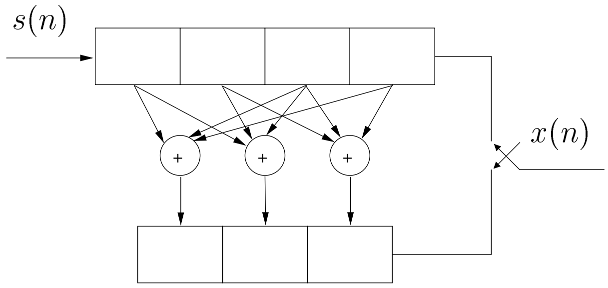

Figure 4: A shift register to produce a linear block code (code B_1) [12].

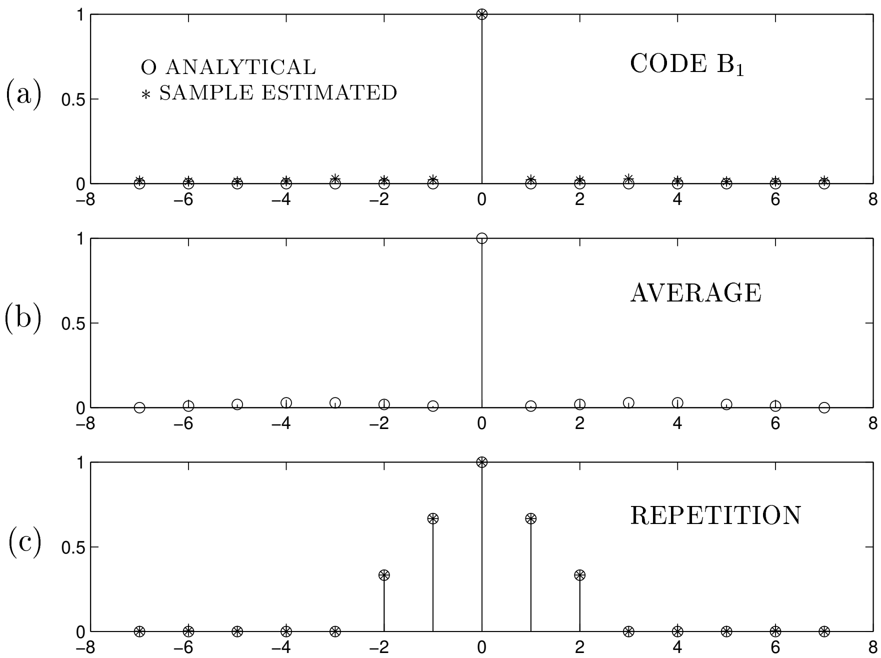

Figure 5: Autocorrelation of linearly block encoded sequences, (a) code B_1, (b) average of 2730 $(7,4)$-systematic codes, (c) a $(3,1)$ repetition code. $N_s = 2000$ for sample estimation.

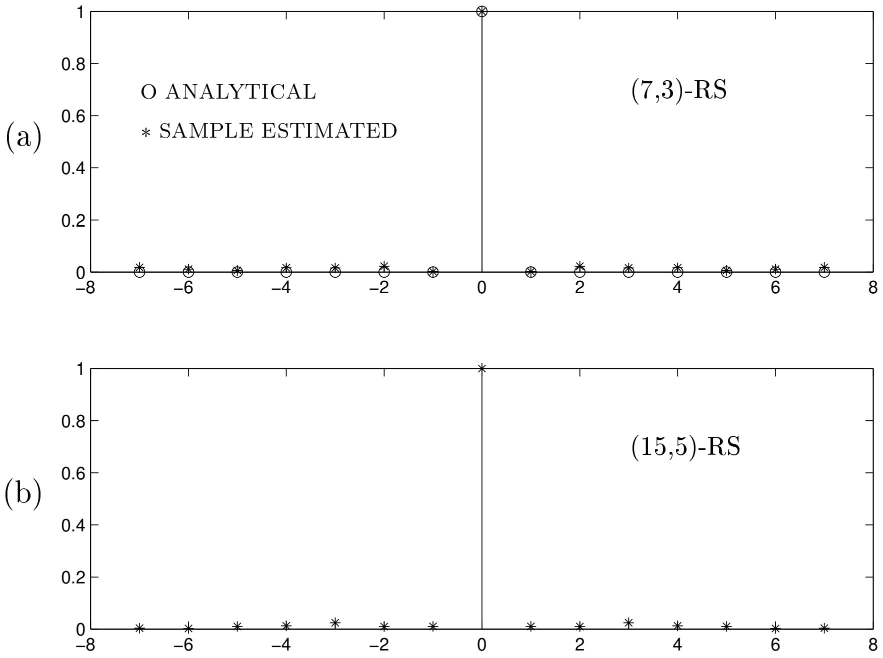

Figure 6: Autocorrelation function of (a) a $(7, 3)$ RS code, (b) a $(15, 5)$ RS code. $N_s =$ 2000 for sample estimation.

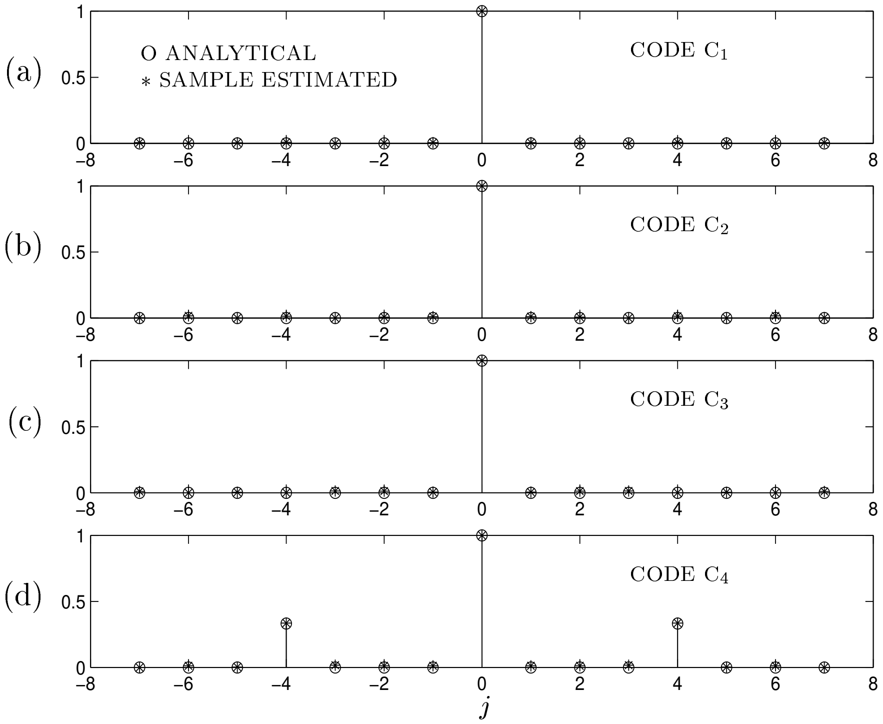

Figure 7: Autocorrelation for codes (a) C_1, (b) C_2, (c) C_3 and (d) C_4. $N_s = 2000$ for sample estimates.

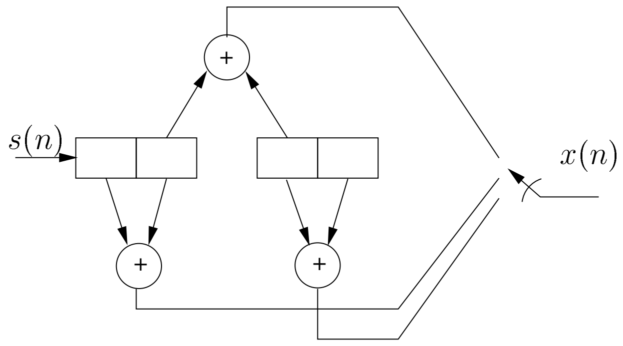

Figure 8: A shift register to produce a rate 2/3 non-white convolutional code (Code C_4).

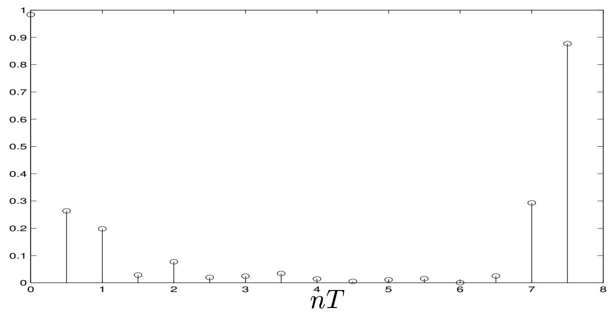

Figure 9: Magnitude of the channel impulse response.

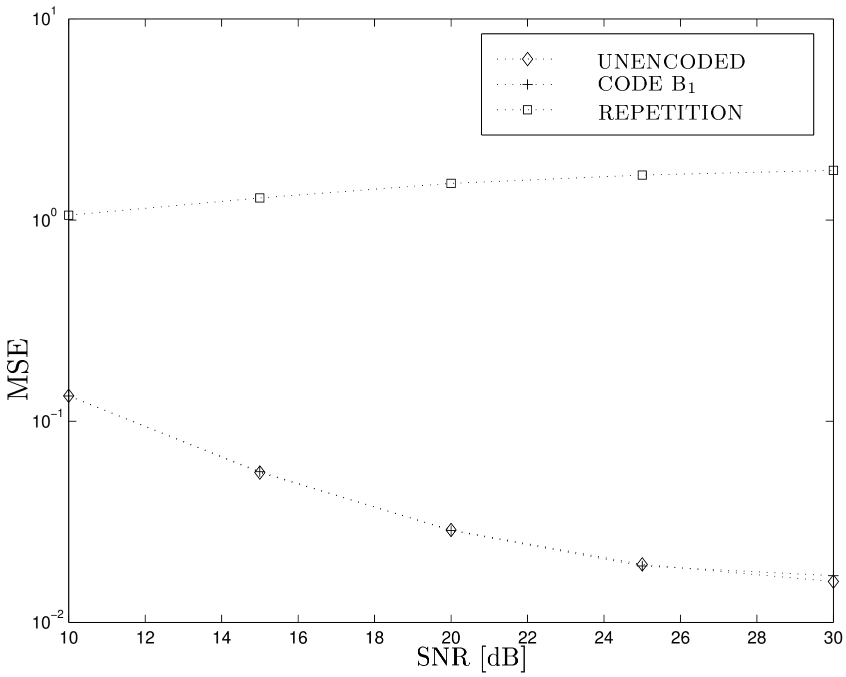

Figure 10: MSE of the PEF-based blind equalizer for an unencoded sequence, block code B_1 and repetition code, sample size 1000.

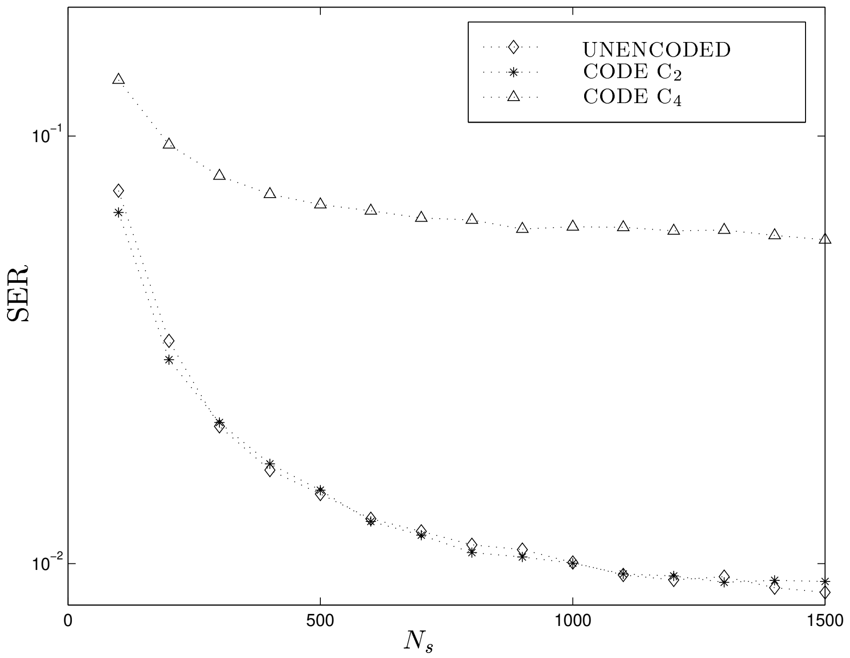

Figure 11: SER convergence of the PEF-based blind equalizer with the convolutional codes, SNR of 10 dB.

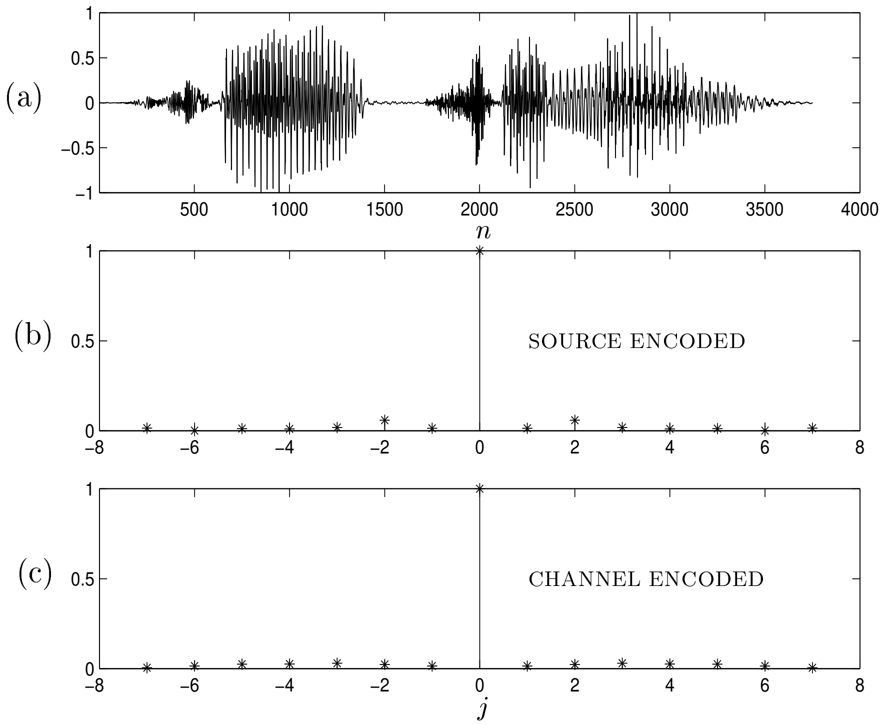

Figure 12: (a) The original speech signal, (b) $\hat{r}_x(j)$ of the GSM speech encoder output and (c) the further C_3 encoded sequence.

[2] G. Bilardi, R. Padovani, and G. L. Pierobon. Spectral analysis of functions of Markov chains with applications. *IEEE Tr. Comm.*, 31(7):853–860, July 1983.

[3] G. L. Cariolaro and G. P. Tronca. Spectra of block coded digital signals. *IEEE Tr. Comm.*, 22(10):1555–1563, October 1974.

[4] T. J. Endres, S. D. Halford, C. R. Johnson, and G. B. Giannakis. Blind adaptive channel equalization using fractionally-spaced receivers: A comparison study. In *Proc. 30th Conf. Inform. Sci. Syst.*, Princeton, NJ, March 1996.

[5] European Telecommunications Standards Institute. Digital cellular telecommunications system; Enhanced Full Rate (EFR) speech transcoding (GSM 06.60), March 1997.

[6] P. Galko and S. Pasupathy. The mean power spectral density of markov chain driven signals. *IEEE Tr. Inf. Theory*, 27(6):746–754, November 1981.

[7] W. A. Gardner. Exploitation of spectral redundancy in cyclostationary signals. *IEEE Signal Processing Magazine*, 8(2):4–36, April 1991.

[8] G. B. Giannakis and S. D. Halford. Blind fractionally spaced equalization of noisy FIR channels: Direct and adaptive solutions. *IEEE Tr. Signal Proc.*, 45(9):2277–2292, September 1997.

[9] S. Haykin, editor. *Communication Systems.* John Wiley & Sons Inc., 3rd edition, 1994.

[10] J. Mannerkoski and V. Koivunen. Are blind equalizers assuming white data applicable to block encoded sequences? In *European Signal Processing Conference, to appear*, Tampere, Finland, September 2000.

[11] E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue. Subspace methods for the blind identification of multichannel FIR filters. *IEEE Tr. Signal Proc.*, 43(2):516–525, February 1995.

[12] J. G. Proakis. *Digital Communications.* McGraw-Hill, 3rd edition, 1995.

[13] S. Roman. *Graduate Texts in Mathematics — Coding and Information Theory.* Springer-Verlag, Harrisonburg, VA, 1992.

[14] J. Shen, and Z. Ding. Direct blind MMSE channel equalization based on second-order statistics. *IEEE Tr. Signal Proc.*, 48(4):1015–1022, April 2000.

[15] D. T. Slock. Blind fractionally-spaced equalization, perfect reconstruction filter banks and multichannel linear prediction. In *IEEE ICASSP*, pages IV585–IV588, Adelaide, Australia, April 1994.

List of Figures

List of Tables

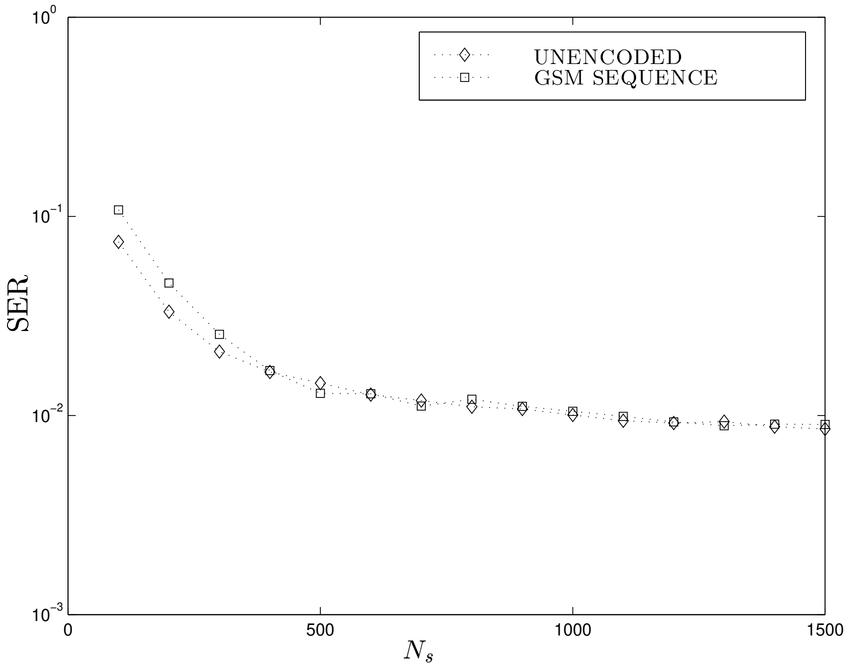

Figure 13: SER convergence with the GSM coder output combined with code C_3, SNR of 10 dB.