On Phase Correct Blind Deconvolution exploiting Channel Coding

Ansgar Scherb, Volker Kühn and Karl-Dirk Kammeyer
Department of Communications Engineering
University of Bremen, Otto-Hahn-Allee, D-28359 Bremen, Germany
Email:{scherb, kuehn, kammeyer}@ant.uni-bremen.de

***Abstract*— In this paper we will derive an algorithm, which estimates the channel blindly exploiting the statistical dependencies of the transmitted signal caused by channel coding. An additional feature of this algorithm is that in contrast to most blind deconvolution algorithms phase correct estimates can be obtained. The error performance of the proposed algorithm depends on the characteristics of the channel code. If the code has appropriate properties, which is true for some convolutional codes as well as for several block codes, e.g. especially low-density parity check codes (LDPC), the proposed algorithm performs similarly or slightly better in comparison to higher order statics based algorithms.**

I. Introduction

Beside subspace algorithms and methods based on the approximation of the maximum likelihood criterion, most blind deconvolution algorithms are based on the assumption of statistically independent sources. The channel causes a superposition of source signals weighted by channel coefficients such that this property is distorted. Thus, a channel equalizer has to restore the statistical independence at the channel output. Due to the fact that a measure for the statistical independence can be obtained from higher orders statistics (HOS), these methods are often called HOS based algorithms [1],[2]. A good survey of blind channel estimation can be found in [3],[4].

Focussing on communications the data are usually channel encoded before transmission in order to protect them against bit errors. The encoding causes statistical dependencies. Therefore, the assumption of independent source signals does not hold exactly in this case. The impact of encoding on HOS based blind channel identification was examined in [5].

However, the key idea of this paper is to exploit the statistical dependencies caused by channel encoding to identify the channel blindly. As we will see, if the code has special properties as shown in Section III, in contrast to HOS based methods phase correct estimation is also feasible. This paper is organized as follows. In Section II the transmission model is presented. In Section III the concept of "logic strings" is introduced in order to describe the statistical dependencies caused by channel coding. In Section IV we will derive an algorithm to blindly estimate the channel exploiting channel coding. The performance of the algorithm is evaluated on the basis of simulation results in Section V and the paper is concluded in Section VI.

II. System Model

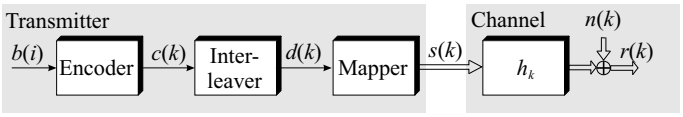

Fig. 1. System Model

Consider an information data sequence of binary bits $\mathbf{b} = [b(1), \cdots, b(I)]^T$ of length I, where $b(i) \in \{0, 1\}$. As shown in Fig. 1 the bits are encoded in order to protect them against errors. The encoded data of length K are denoted by $\mathbf{c} = [c(1), \cdots, c(K)]^T$, where $K > I$, $c(k) \in \{0, 1\}$ and $\mathcal{C}$ is the set of all valid code words.

The relation between origin and encoded data is given by

$$c(k) = g_{k,1}b(1) \oplus g_{k,2}b(2) \oplus \cdots \oplus g_{k,I}b(I), \tag{1}$$

where $g_{k,i} \in \{0, 1\}$ and $\oplus$ is the XOR operator. Therefore, a vector matrix notation is given by

$$\mathbf{c} = (\mathbf{G}\mathbf{b})_{\mathrm{mod}_2}, \tag{2}$$

where $g_{k,i}$ is the k-th row and i-th column element of $\mathbf{G}$.

The encoded data $\mathbf{c}$ are interleaved by

$$\mathbf{d} = \mathbf{P}\mathbf{c}, \tag{3}$$

where $\mathbf{P} \in \{0, 1\}^{K \times K}$ contains by definition exactly one non-zero element in each row and column. The row index of the non-zero element in the k-th column will be denoted by $\pi(k)$, such that $c(k) = d(\pi(k))$ holds.

The encoded bits are mapped onto BPSK symbols $s(k)$ by assigning $0 \Rightarrow 1$ and $1 \Rightarrow -1$.

The transmission is characterized by the discrete time channel impulse response (CIR) h_κ of order H including pulse shaping and receive filter. The receiver input can be expressed as

$$r(k) = \sum_{\kappa=0}^{H} h_\kappa s(k - \kappa) + n(k). \tag{4}$$

III. Properties of encoded Sequences

A. The Concept of Logic Strings

In order to exploit the statistical dependencies caused by the channel coding, we make use of a concept termed as

logic strings[1] which is defined as follows:

Definition 1: A set of encoded bits is called *logic string*, when the XOR-conjunction of its elements always gives a zero for each arbitrary original data sequence.

In order to explain this definition, let us assume that a rule exists drawing N logic strings of length M according to an encoding scheme. We denote by $\mathcal{A}$ the set of cardinality N containing all time indices corresponding to logic strings of this type. Let $[\tau_{1,n}, \tau_{2,n}, \cdots, \tau_{M,n}] \in \mathcal{A}$ be the time indices corresponding to the n-th valid logic string. Then,

$$c(\tau_{1,n}) \oplus c(\tau_{2,n}) \oplus \cdots \oplus c(\tau_{M,n}) = 0 \tag{5}$$

holds for $n = 1, \cdots, N$.

After interleaving the string $d(\pi_{1,n}) \oplus d(\pi_{2,n}) \oplus d(\pi_{M,n}) = 0$ is equivalent to (5), where $\pi_{n,m} = \pi(\tau_{n,m})$ and the set of valid logic strings is determined by $\mathcal{B} = \{[\pi(\tau_{1,n}), \pi(\tau_{2,n}), \cdots, \pi(\tau_{M,n})] \quad | \quad [\tau_{1,n}, \tau_{2,n}, \cdots, \tau_{M,n}] \in \mathcal{A}\}$. After mapping the encoded bits onto signal space, the XOR operator in (5) can be replaced by multiplications such that

$$s(\pi_{1,n})s(\pi_{2,n}) \cdots s(\pi_{M,n}) = 1 \tag{6}$$

holds. Furthermore, the M-th order moments for any $k_1, k_2, \cdots, k_M \notin \mathcal{B}$ vanish, i.e.

$$E\{s(k_1)s(k_2) \cdots s(k_M)\} = 0. \tag{7}$$

B. Asymmetry

The goal of phase correct channel estimation can be attained, if the encoding scheme is non-symmetric, where the term "asymmetric" is defined as follows:

Definition 2: A code is called *asymmetric* if the negation of each valid code word is not a valid code word, i.e.

$$\mathbf{c} \in \mathcal{C} \Rightarrow \overline{\mathbf{c}} \notin \mathcal{C}. \tag{8}$$

Note that if the code incorporates a logic string of odd length M, the code is always non-symmetric. The requirements will become more clear in the example given below.

C. Example: Convolution Code

As example we examine the half rate $(7, 5)$-code. The encoding rule for odd indexed c can be expressed as

$$c(2n-1) = b(n) \oplus b(n+1) \oplus b(n+2) \quad ; \quad n = 1, \cdots, N \tag{9}$$

and for even indexed c as

$$c(2n) = b(n) \oplus b(n+2) \quad ; \quad n = 1, \cdots, N \tag{10}$$

[1]In terms of coding theory, the concept of logic strings is closely related to the representation of the parity check matrix and the null space corresponding to the code space, respectively. An important difference is that the length of a set of logic strings is fixed, whereas e.g. the number of ones in each column vector of a syndrome matrix may vary.

A logic string of length 5 for this encoder is given by

$$\begin{aligned} & c(2n-1) \oplus c(2n) \oplus c(2n+2) \oplus c(2n+3) \oplus c(2n+4) \\ = \; & b(n) \oplus b(n+1) \oplus b(n+2) \\ & \oplus b(n) \oplus b(n+2) \\ & \oplus b(n+1) \oplus b(n+3) \\ & \oplus b(n+2) \oplus b(n+3) \oplus b(n+4) \\ & \oplus b(n+2) \oplus b(n+4) \\ = \; & 0. \end{aligned} \tag{11}$$

A rule displaying all time indices $\tau_{m,n}$ corresponding to the set of N logic strings of length $M = 5$ is given by

$$\tau_{m,n} = \begin{cases} 2n+m-2 & m = 1, 2 \\ 2n+m-1 & m = 3, 4, 5 \end{cases}. \tag{12}$$

There may exist other logic strings of higher length, which can be found by a heuristic search. However, this is the shortest string for this code. Due to the odd number of elements of the logic strings defined in (12), this code is also non-symmetric.

IV. Algorithm

On the basis of the results of Section III we are now able to derive a simple blind channel estimator for the l-th channel coefficient exploiting the properties of (6) and (7). As we will see, an estimate of the l-th channel gain can be obtained by the expectation of the product of the received data according to an arbitrary logic string:

$$\check{h}_l = E\left\{ r(\pi_{1,n} + l) \prod_{m=2}^{P+1} r(\pi_{m,n}) \prod_{m=P+2}^{M} r^*(\pi_{m,n}) \right\}, \tag{13}$$

where $P = (M-1)/2$. Note that apart from the first factor the number P of the complex conjugated factors is equal to the number of non complex conjugated factors. This relation holds only, if the length of the concerning logic string is odd. Remember that in this case the code is always asymmetric and therefore we can achieve a phase correct channel estimation. As we will see, the complex conjugation of P factors in (13) yields a phase correct estimator.

After replacing $r(k)$ in (13) by the r.h.s of (4) we obtain

$$\begin{aligned} \check{h}_l \; = \; & E\left\{ \left(\sum_{\kappa_1=0}^{H} h_{\kappa_1} s(\pi_{1,n} + l - \kappa_1) + n(\pi_{1,n} + l) \right) \right. \\ & \prod_{m=2}^{P+1} \left(\sum_{\kappa_m=0}^{H} h_{\kappa_m} s(\pi_{m,n} - \kappa_m) + n(\pi_{m,n}) \right) \\ & \left. \prod_{m=P+2}^{M} \left(\sum_{\kappa_m=0}^{H} h^*_{\kappa_m} s^*(\pi_{m,n} - \kappa_m) + n^*(\pi_{m,n}) \right) \right\}. \end{aligned} \tag{14}$$

In order to obtain a more illustrative expression, we want to exchange products and sums in (14). Since the noise is i.i.d, all elements incorporating noise vanish. Therefore, in the sequel noise will be neglected.

Let $\mathcal{K} = \{(\kappa_1, \cdots, \kappa_M) | 0 \leq \kappa_m \leq H\}$ be the set of M-tuples with cardinality $(H+1)^M$ consisting of all combinations

of indices which occur in (14). Now, (14) can be rewritten as (15). Assuming that no logic string is accidentally[2] caught in $v(\kappa_1,\cdots,\kappa_M)$ in (15), due to (7) only the left term assigned to the logic string n remains such that

$$\check{h}_l = h_l|h_0|^{M-1}. \tag{16}$$

Hence, the estimate $\check{h}_l$ consists of the true channel coefficient h_l, which is weighted by a real positive factor $|h_0|^{M-1}$. Unfortunately, in real applications the expectation in (15) is not available. Thus, it must be approximated by averaging over N logic strings.

$$\begin{aligned}\check{h}_l &= \frac{1}{N}\sum_{n=1}^{N} r(\pi_{1,n}+l)\prod_{m=2}^{P+1} r(\pi_{m,n})\prod_{m=P+2}^{M} r^*(\pi_{m,n}) &(17)\\ &= h_l|h_0|^{M-1}\frac{1}{N}\sum_{n=1}^{N}\prod_{m=1}^{M} s(\pi_{m,n}) &(18)\\ &+ \sum_{(\kappa_1,\cdots,\kappa_M)\in\mathcal{K}\setminus(l,0,\cdots,0)} u(\kappa_1,\cdots,\kappa_M)\hat{v}(\kappa_1,\cdots,\kappa_M),\end{aligned}$$

where

$$\begin{aligned}&\hat{v}(\kappa_1,\cdots,\kappa_M)\\ &= \frac{1}{N}\sum_{n=1}^{N}\left(s(\pi_{m,n}+l-\kappa_1)\prod_{m=2}^{M} s(\pi_{m,n}-\kappa_m)\right) &(19)\end{aligned}$$

In the face of a finite number N, the term $\hat{v}(\kappa_1,\cdots,\kappa_M)$ does not become exactly zero for $(\kappa_1,\cdots,\kappa_M)\in\mathcal{K}\setminus(l,0,\cdots,0)$. Therefore, an error remains weighted by $u(\kappa_1,\cdots,\kappa_M)$. The error power depends on the characteristics of the channel and how it weights the useful part in relation to the remaining error. However, in the case of ill conditioned channel characteristics ($|h_0|\ll|h_\kappa|:\kappa\neq 0$) the performance of the estimator might become even worse. In order to combat this effect, we suggest to insert a linear equalizer $\mathbf{e}=[e_L,\cdots,e_0]^H$ of order L at the receiver.

On the basis of the channel estimate the filter coefficients can be adjusted. Otherwise, an appropriate linear filter $\mathbf{e}$ may assist reducing the impact of the error in $\hat{v}(\kappa_1,\cdots,\kappa_M)$. Therefore, we suggest an iterative two-step algorithm, where channel estimation and filter adaptation are repeated alternately until the algorithm converges.

Sampling $L+1$ data at the receiver, we obtain

$$\mathbf{r}(k) = \mathbf{H}\mathbf{s}(k)+\mathbf{n}(k) \tag{20}$$

[2]In order to limit the effort of decoding, the encoder usually shows a periodical structure. It can be shown that in these case also the logic strings may have a periodical structure. Due to the fact that the channel causes a weighted superposition of encoded data, it is not unlikely that beside the logic string of interest a parasitic logic string is caught. In order to avoid this effect interleaving the encoded data is mandatory for our approach.

where $\mathbf{s}(k)=[s(k-L-H+k_0),\cdots,s(k+k_0)]$ with k_0 is an arbitrary delay in the interval $H<k_0<L$, $\mathbf{n}(k)=[n(k-L),\cdots,n(k)]$ and $n(k)$ is additive white gaussian noise of power $E\{|n(k)|^2\}=\sigma_n^2$. The $(L+1\times L+H+1)$-matrix $\mathbf{H}$ is defined by

$$[\mathbf{H}]_{\nu,\mu} = \begin{cases} h_{H+\nu-\mu} & : \quad 0\le\nu-\mu\le H \\ 0 & : \quad \text{else}\end{cases}. \tag{21}$$

Let us denote the output of the filter by

$$\begin{aligned} y(k) &= \mathbf{e}^H\mathbf{r}(k)\\ &= \mathbf{e}^H\mathbf{H}\mathbf{s}(k)+\mathbf{e}^H\mathbf{n}(k)\\ &= \mathbf{w}^H\mathbf{s}(k)+\eta(k), &(22)\end{aligned}$$

where $\mathbf{w}=[w_{L+H},\cdots,w_0]^H$ is the joint impulse response of channel and filter and $\eta(k)=\mathbf{e}^H\mathbf{n}(k)$.

The channel estimate can be improved using the outputs of the filter by replacing the $M-1$ last factors in (17) by the filter outputs we obtain (23), where $\mathcal{L}=\{(\kappa_1,\cdots,\kappa_M)\}$ with $0\le\kappa_1\le H$ and $0\le\kappa_m\le H+L$ for $m=2,\cdots,M$ is the set of M-tuples with cardinality $(H+1)(H+L+1)^{M-1}$ and

$$\tilde{u}(\kappa_1,\cdots,\kappa_M) = h_{\kappa_1-l}\prod_{m=2}^{P+1} w_{\kappa_m}\prod_{m=P+2}^{M} w^*_{\kappa_m}. \tag{24}$$

From (23) and (24) it can be seen, that if $\mathbf{e}$ nearly represents the ideal channel equalizer, the factors $\tilde{u}(\kappa_1,\cdots,\kappa_M)$ for $(\kappa_1,\cdots,\kappa_M)\in\mathcal{L}\setminus(l,0,\cdots,0)$ weighting the remaining errors become very small.

On the basis of the channel estimate the filter's coefficients can be adjusted, e.g. by the MMSE approach [6] with

$$\mathbf{e}' = \mathbf{\Phi}_{rr}^{-1}\tilde{\mathbf{h}}, \tag{25}$$

where $\tilde{\mathbf{h}}=[0,\cdots,0,\hat{h}(0),\cdots,\hat{h}(H),0,\cdots,0]^T$ is a vector of length L with k_0 zeros at the beginning and $\mathbf{\Phi}_{rr}=E\{\mathbf{r}(k)\mathbf{r}^H(k)\}$ is the covariance matrix of the receive signal. In order to get close to the absolute of the true CIR and to avoid a bit overflow, $\mathbf{e}'$ should be normalized by

$$\mathbf{e} = \frac{\mathbf{e}'}{\sqrt{\mathbf{e}'^H\mathbf{\Phi}_{rr}\mathbf{e}'}}. \tag{26}$$

In the initial step $\mathbf{e}$ should contain at least one non-zero entry at index $H<k_0<L$ in order to ensure a non zero entry of the joint impulse response $\mathbf{w}$ at index k. As initialization, $\mathbf{e}_0=\boldsymbol{\delta}_{k_0}$ seems to be a good choice, where $\boldsymbol{\delta}_{k_0}$ is an unit vector and the k_0-th element is 1 . The algorithm is summarized in Tab. I.

$$\check{h}_l = h_l|h_0|^{M-1}E\left\{\prod_{m=1}^{M} s(\pi_{m,n})\right\} + \sum_{(\kappa_1,\cdots,\kappa_M)\in\mathcal{K}\setminus(l,0,\cdots,0)} \underbrace{h_{\kappa_1+l}\prod_{m=2}^{P+1} h_{\kappa_m}\prod_{m=P+2}^{M} h^*_{\kappa_m}}_{u(\kappa_1,\cdots,\kappa_M)} \underbrace{E\left\{s(\pi_{m,n}+l-\kappa_1)\prod_{m=2}^{M} s(\pi_{m,n}-\kappa_m)\right\}}_{v(\kappa_1,\cdots,\kappa_M)} \tag{15}$$

TABLE I

PHASE CORRECT BLIND DECONVOLUTION EXPLOITING CHANNEL CODING (BDCC)

1	:	Initialize $\mathbf{e}_0 = \boldsymbol{\delta}_{k_0}$ and the iteration counter $\iota = 0$.
2	:	**repeat**
3	:	Estimate the channel by (23).
4	:	Update the linear equalizer by (25) and (26).
5	:	Set $\iota = \iota + 1$.
6	:	**end**

V. SIMULATION RESULTS

The estimation performance is evaluated over normalized mean squared error (NMSE) between the true channel and the estimated channel, which is defined as

$$\text{NMSE} = \frac{\sum_{l=0}^{H} |h_l - \hat{h}_l|^2}{\sum_{l=0}^{H} |h_l|^2}. \tag{27}$$

The data are encoded by a (75)-convolutional (conv75) encoder with rate $1/2$, which comprises $K/2-1$ logic strings of length $M = 5$, and by a single parity check code with rate $2/3$ (spcc), which comprises $K/3$ logic strings of length $M = 3$. As reference, the super exponential algorithm (SEA) [2] is used, which is a typical HOS-based method. As in our approach, SEA will also consist of several iterations, where jointly a linear filter is adapted and the channel is estimated.

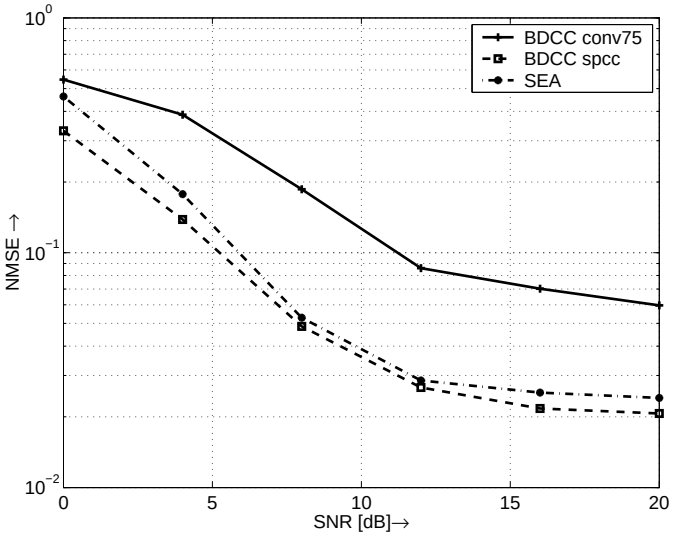

Fig. 2. NMSE vs. SNR

In simulations we used a 3-tap Rayleigh fading channel with constant coefficients for a block of $K = 400$ symbols (block fading channel). The order of the linear filter was $L = 10$ and the algorithms was stopped after 5 iterations. The results are averaged over 5000 channel realizations.

Fig. 2 shows the NMSE performance versus signal to noise ratio (SNR). It can be seen, that the performance of the blind deconvolution exploiting channel coding (BDCC) depends strongly on the length M of the particular logic strings. The reason may be on the one hand that the number of terms, which are accumulated in (23) is $(H+1)^M$ and consequently the performance is very sensitive with respect to M. On the other hand also the impact of noise on the estimation must be taken into account. For small M the BDCC performs similarly or slightly better than SEA.

VI. CONCLUSION

In this paper we have derived an iterative algorithm, which blindly identifies the channel exploiting the statistical dependencies of the transmitted signal caused by channel coding. The algorithm jointly updates channel estimates and adapts a linear equalizer. For asymmetric channel coding, the algorithm delivers phase correct estimates. It was shown by simulations, that the performance of the proposed algorithm depends on the properties of the code and the length of its logic strings, respectively. In the case of short logic strings the algorithm performs slightly better than the HOS based methods and additionally delivers phase correct estimates.

REFERENCES

[1] B. Jelonnek D. Boss and K.D. Kammeyer. Eigenvector Algorithm for Blind MA System Identification. *Elsevier Signal Processing*, 66(1):1–26, April 1998.

[2] O. Shalvi and E. Weinstein. Super-exponential methods for blind deconvolution. *IEEE Trans. on Information Theory*, 39:504–519, March 1993.

[3] L. Tong and S. Perreau. Multichannel Blind Identification: From Subspace to Maximum Likelihood Methods. *IEEE Proceedings*, 86(10):1951–1968, October 1998.

[4] C.H. Chen C.Y. Chi, C.Y. Chen and C.C. Feng. Batch Processing Algorithms for Blind Equalization using Higher Order Statistics. *IEEE Signal Processing Magazine*, 20(1):25–49, January 2003.

[5] J. Mannerkoski and V. Koivunen. Autocorrelation properties of channel encoded sequences- Applicability to blind equalization. *IEEE Trans. on Signal Processing*, 48:3501–3507, December 2000.

[6] S.M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall PTR, 1993.

$$\begin{aligned}
\hat{h}_l &= \frac{1}{N}\sum_{n=1}^{N} r(\pi_{1,n}+l) \prod_{m=2}^{P+1} \mathbf{e}^H \mathbf{r}(\pi_{m,n}) \prod_{m=P+2}^{M} \mathbf{r}^H(\pi_{m,n})\mathbf{e} \\
&= h_l |w_0|^{M-1} \frac{1}{N}\sum_{n=1}^{N}\prod_{m=1}^{M} s(\pi_{m,n}) + \sum_{(\kappa_1,\cdots,\kappa_M)\in\mathcal{L}\setminus(l,0,\cdots,0)} \tilde{u}(\kappa_1,\cdots,\kappa_M)\hat{v}(\kappa_1,\cdots,\kappa_M) \\
&\approx h_l |w_0|^{M-1},
\end{aligned} \tag{23}$$