Fast Noncoherent Decoding of Block Transmissions over Doubly Dispersive Channels

Sung-Jun Hwang and Philip Schniter
The Ohio State University, Dept. of Electrical and Computer Engineering,
2015 Neil Avenue, Columbus, OH 43210.
E-mail: {hwangsu, schniter}@ece.osu.edu

***Abstract*—We propose a scheme for noncoherent iterative (i.e., turbo) reception of coded block transmissions over unknown time and frequency selective, or doubly dispersive, channels. Starting with a noncoherent metric that leverages a basis expansion model (BEM) for the channel's time-variation, we propose an efficient noncoherent soft equalization strategy that combines sub-optimal tree search with a fast noncoherent metric update. Though the complexity of our scheme is only linear in the block length and quadratic in the number of BEM parameters, numerical experiments show that it attains a performance relatively close to that of the turbo receiver with perfectly known channel.**[1]

I. Introduction

In this paper, we consider the problem of decoding a data sequence transmitted over a time- and frequency-selective channel, otherwise known as a doubly selective or doubly dispersive (DD) channel, whose realizations are unknown but whose statistics are known. In particular, we are interested in the case of coded transmissions with possibly long codewords (as with LDPC or turbo codes). While the maximum a posteriori (MAP) bit detector is known to minimize the bit error rate (BER) [1], it is too complex to implement for the codes and channels of interest. A near-optimal but significantly cheaper strategy follows from the turbo principle [2], which suggests to iterate between separate soft equalization and decoding steps. In this case, the equalizer's role becomes that of producing posterior bit probabilities from the received samples and any extrinsic information previously supplied by the decoder.

The calculation of posterior bit information in the presence of an unknown DD channel is not a trivial task, however. (See [3] for a recent overview). In the most common approach to the problem, the channel is modeled as a first-order Gauss-Markov process and trellis-based methods are used with either forward-backward or fixed-lag MAP processing. Fitting a realistic time-varying channel into this framework generally requires the use of approximations which degrade performance and/or limit the range of applicability (e.g., to slowly varying channels).

We consider a different approach to soft noncoherent equalization which uses a basis expansion model (BEM) [4], [5] for channel variations. The use of a BEM yields an efficient channel parameterization which, as we will show, translates directly into an efficient soft noncoherent equalization algorithm. In addition, the flexibility of the BEM approach makes it directly applicable to, e.g., frequency-domain channel models (as arise with OFDM) and/or sparse channel models. (See, e.g., [6].)

The soft noncoherent equalizer we propose for DD channels builds on recent ideas from the multiple-input multiple-output (MIMO) literature (e.g., [7], [8]), such as the use of suboptimal tree search to find the dominant contributions to a noncoherent metric. Our principal contribution is the derivation of a fast algorithm for the sequential update of the BEM-based noncoherent metric. In particular, the proposed algorithm yields a complexity that scales linearly in the block length and quadratically in the number of BEM parameters. Numerical experiments show that the proposed technique maintains performance relatively close to that of turbo reception under a perfectly known channel.

II. System Model

At the transmitter, information bits are rate-R coded, interleaved, and mapped to 2^Q-ary QAM symbols. Groups of N_s information symbols are then combined with pilot and guard symbols to form transmission blocks of length $N \geq N_s$. (Details on the pilots and guards will be given later.) The j^{th} block is composed of the symbols $\{s_n^{(j)}\}_{n=0}^{N-1}$ which correspond to the coded bits $\{x_k^{(j)}\}_{k=0}^{N_sQ-1}$. In particular, symbol $s_n^{(j)}$ is mapped from the coded bits $\underline{x}_n^{(j)} := [x_{nQ}^{(j)}, \ldots, x_{nQ+Q-1}^{(j)}]^T$.

A linear time-varying noisy channel gives the n^{th} sample of the j^{th} received block the form

$$r_n^{(j)} = \sum_{l=0}^{N_h-1} h_{n,l}^{(j)} s_{n-l}^{(j)} + v_n^{(j)}, \qquad (1)$$

where $h_{n,l}^{(j)}$ is the channel coefficient at time-n and delay-l, N_h is the discrete delay spread, and $\{v_n^{(j)}\}$ is zero-mean circular white Gaussian noise (CWGN) with covariance σ^2. The channel is assumed to be Rayleigh fading and wide-sense stationary uncorrelated scattering (WSSUS) [9], so that $\{h_{n,l}^{(j)}\}$ are zero-mean circular Gaussian with $\mathrm{E}\{h_{n,l}^{(j)} h_{n-m,l-\ell}^{(j)*}\} = \rho_m \sigma_l^2 \delta_\ell$. Here, ρ_m is the temporal autocorrelation, σ_l^2 is the delay-power profile, and δ_ℓ is the Kronecker delta sequence.

The receiver consists of a soft noncoherent equalizer and a soft decoder, connected in the turbo configuration of Fig. 1. The equalizer uses the observations $\{r_n^{(j)}\}$, as well as any *a priori* information provided by the decoder, to generate soft

[1]This work was supported by the National Science Foundation under grant CCR-0237037 and the Office of Naval Research under grant N00014-07-1-0209.

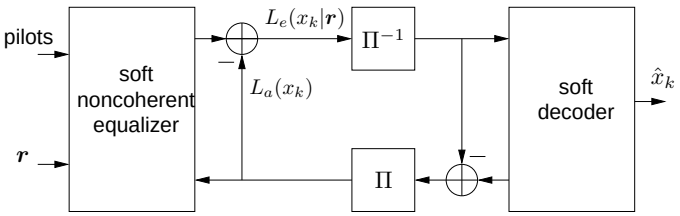

Fig. 1. Iterative noncoherent receiver structure.

information on the coded bits, leveraging its knowledge of pilot symbols and statistical channel structure. The decoder uses the soft equalizer outputs to infer the information bits, leveraging its knowledge of the code structure. Also, the decoder may refine the soft information on the coded bits for subsequent use by the equalizer.

A key feature of our equalizer is its use of a N_b-term basis expansion model (BEM) for channel variation over the block:

$$h_{n,l}^{(j)} \approx \sum_{p=0}^{N_b-1} b_{n,p}\theta_{p,l}^{(j)} \quad \text{for } n \in \{0,\dots,N-1\}. \tag{2}$$

Here, N_b and $\{b_{n,p}\}$ are design parameters whereas $\{\theta_{p,l}^{(j)}\}$ are unknown channel coefficients. While an error-free approximation is possible with $N_b = N$, significant reduction in receiver complexity is possible with $N_b \ll N$. Under approximation (2), the received sample $r_n^{(j)}$ from (1) becomes

$$r_n^{(j)} \;=\; \boldsymbol{b}_n^H \sum_{l=0}^{N_h-1} s_{n-l}^{(j)}\boldsymbol{\theta}_l^{(j)} + v_n^{(j)}, \tag{3}$$

for $\boldsymbol{b}_n := [b_{n,0},\dots,b_{n,N_b-1}]^H$ and $\boldsymbol{\theta}_l^{(j)} := [\theta_{0,l}^{(j)},\dots,\theta_{N_b-1,l}^{(j)}]^T$. The up-to-time-$n$ observations $\boldsymbol{r}_n^{(j)} := [r_0^{(j)},\dots,r_n^{(j)}]^T$ then become

$$\boldsymbol{r}_n^{(j)} \;=\; \boldsymbol{A}_n^{(j)}\boldsymbol{\theta}^{(j)} + \boldsymbol{v}_n^{(j)} \quad \text{for } n \in \{0,\dots,N-1\}, \tag{4}$$

where $\boldsymbol{\theta}^{(j)} := [\boldsymbol{\theta}_0^{(j)T},\dots,\boldsymbol{\theta}_{N_h-1}^{(j)T}]^T$, $\boldsymbol{v}_n^{(j)} := [v_0^{(j)},\dots,v_n^{(j)}]^T$, and

$$\boldsymbol{A}_n^{(j)} := \begin{bmatrix} s_0^{(j)}\boldsymbol{b}_0^H & \cdots & s_{-N_h+1}^{(j)}\boldsymbol{b}_0^H \\ \vdots & & \vdots \\ s_n^{(j)}\boldsymbol{b}_n^H & \cdots & s_{n-N_h+1}^{(j)}\boldsymbol{b}_n^H \end{bmatrix}. \tag{5}$$

Similar to $\boldsymbol{r}_n^{(j)}$, we can define $\boldsymbol{s}_n^{(j)} := [s_0^{(j)},\dots,s_n^{(j)}]^T$ and $\boldsymbol{x}_n^{(j)} = [\underline{x}_0^{(j)T},\dots,\underline{x}_n^{(j)T}]^T$. For brevity, the full-block quantities $\boldsymbol{r}_{N-1}^{(j)}$, $\boldsymbol{A}_{N-1}^{(j)}$, $\boldsymbol{s}_{N-1}^{(j)}$, $\boldsymbol{v}_{N-1}^{(j)}$, and $\boldsymbol{x}_{N-1}^{(j)}$ will be abbreviated by $\boldsymbol{r}^{(j)}$, $\boldsymbol{A}^{(j)}$, $\boldsymbol{s}^{(j)}$, $\boldsymbol{v}^{(j)}$, and $\boldsymbol{x}^{(j)}$, respectively.

III. Noncoherent Soft Equalizer

In this section, we describe the proposed noncoherent soft equalizer, where the soft information takes the form of log-likelihood ratios (LLRs) on coded bits. Given the observation $\boldsymbol{r}^{(j)}$, and any *a priori* LLRs made available by the decoder, the soft equalizer generates LLRs for each of the coded bits in $\boldsymbol{x}^{(j)}$. The equalizer is "noncoherent" in that it treats the channel realization $\boldsymbol{\theta}^{(j)}$ as unknown. However, it is assumed to know the distributions of $\boldsymbol{\theta}^{(j)}$ and $\boldsymbol{v}^{(j)}$, which in our case are $\mathcal{CN}(\boldsymbol{0},\boldsymbol{R}_\theta)$ and $\mathcal{CN}(\boldsymbol{0},\sigma^2\boldsymbol{I})$, respectively. We will also assume that $s_n^{(j)}|_{n<0} = 0$, which in practice means that the transmission blocks are separated by guard intervals of length $\geq N_h - 1$. In addition, we make the mild assumption that $\boldsymbol{R}_\theta$ is full rank. Because the algorithm does not depend on the block index j, we suppress the block notation "$^{(j)}$" in the sequel.

A. LLR Approximation

The log-likelihood ratio (LLR) of coded bit x_k given $\boldsymbol{r}$,

$$L(x_k|\boldsymbol{r}) \;:=\; \ln\frac{\Pr[x_k=1|\boldsymbol{r}]}{\Pr[x_k=0|\boldsymbol{r}]}, \quad k \in \{0,\dots,N_sQ-1\}, \tag{6}$$

can be written in the form [7]

$$L(x_k|\boldsymbol{r}) \;=\; \ln\frac{\sum_{\boldsymbol{x}:x_k=1} p(\boldsymbol{r}|\boldsymbol{x})\cdot\exp\boldsymbol{l}^T\boldsymbol{x}}{\sum_{\boldsymbol{x}:x_k=0} p(\boldsymbol{r}|\boldsymbol{x})\cdot\exp\boldsymbol{l}^T\boldsymbol{x}}, \tag{7}$$

where $\boldsymbol{l} := [L_a(x_0),\dots,L_a(x_{N_sQ})]^T$ and $L_a(x_k) := \ln P[x_k=1]/P[x_k=0]$ denotes the *a priori* LLR of x_k. The use of the metric

$$\mu(\boldsymbol{x}) \;:=\; \ln p(\boldsymbol{r}|\boldsymbol{x}) + \boldsymbol{l}^T\boldsymbol{x} \tag{8}$$

allows the "extrinsic" LLR $L_e(x_k|\boldsymbol{r}) := L(x_k|\boldsymbol{r}) - L_a(x_k)$ to be written

$$L_e(x_k|\boldsymbol{r}) \;=\; \ln\frac{\sum_{\boldsymbol{x}:x_k=1}\exp\mu(\boldsymbol{x})}{\sum_{\boldsymbol{x}:x_k=0}\exp\mu(\boldsymbol{x})} - L_a(x_k). \tag{9}$$

Computing $L_e(x_k|\boldsymbol{r})$ from (9) requires 2^{N_sQ} evaluations of $\mu(\boldsymbol{x})$, and hence is impractical. However, as suggested in [7], the extrinsic LLR $L_e(x_k|\boldsymbol{r})$ can be approximated as

$$L_e(x_k|\boldsymbol{r}) \;\approx\; \max_{\boldsymbol{x}\in\mathcal{L}\cap\{\boldsymbol{x}:x_k=1\}}\mu(\boldsymbol{x}) \;-\; \max_{\boldsymbol{x}\in\mathcal{L}\cap\{\boldsymbol{x}:x_k=0\}}\mu(\boldsymbol{x}) - L_a(x_k), \tag{10}$$

using the "max-log" approximation $\sum_{\boldsymbol{x}:x_k=1}\exp\mu(\boldsymbol{x}) \approx \max_{\boldsymbol{x}:x_k=1}\mu(\boldsymbol{x})$ and subsequently restricting the maximization search space to the "most important" sequences $\mathcal{L}$.

B. LLR Evaluation via Tree Search

The Rayleigh fading model (4) implies that

$$\boldsymbol{r}|\boldsymbol{x} \;\sim\; \mathcal{CN}(\boldsymbol{0},\boldsymbol{A}\boldsymbol{R}_\theta\boldsymbol{A}^H + \sigma^2\boldsymbol{I}), \tag{11}$$

where $\boldsymbol{A}$ depends on the coded bits $\boldsymbol{x}$ through the corresponding symbols $\boldsymbol{s}$. Thus, the use of $\boldsymbol{\Phi} := \boldsymbol{A}\boldsymbol{R}_\theta\boldsymbol{A}^H + \sigma^2\boldsymbol{I}_N$ yields

$$\ln p(\boldsymbol{r}|\boldsymbol{x}) \;=\; -\boldsymbol{r}^H\boldsymbol{\Phi}^{-1}\boldsymbol{r} - \ln(\pi^N\det\boldsymbol{\Phi}), \tag{12}$$

allowing the metric to be written as

$$\mu(\boldsymbol{x}) \;=\; -\boldsymbol{r}^H\boldsymbol{\Phi}^{-1}\boldsymbol{r} - \ln(\pi^N\det\boldsymbol{\Phi}) + \boldsymbol{l}^T\boldsymbol{x}. \tag{13}$$

Because direct evaluation of (13) requires $\mathcal{O}(N^3)$ operations, we recognize two principle challenges in evaluating (10):

1) Efficient selection of the "most important" sequences $\mathcal{L}$,
2) Fast calculation of $\mu(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{L}$.

As suggested in [8], both challenges can be met by evaluating the partial metric

$$\mu(\boldsymbol{x}_n) \;:=\; \ln p(\boldsymbol{r}_n|\boldsymbol{x}_n) + \boldsymbol{l}_n^T \boldsymbol{x}_n \tag{14}$$

sequentially (i.e., as $\mu(\boldsymbol{x}_0), \mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_{N-1})$) using M possibilities of each partial bit vector $\boldsymbol{x}_n$, where now $\boldsymbol{l}_n := [\underline{l}_0, \ldots, \underline{l}_n]^T$ and $\underline{l}_i := [L(x_{iQ}), \ldots, L(x_{iQ+Q-1})]^T$. To choose the M possibilities of $\boldsymbol{x}_n$, all one-symbol extensions of the M "most important" partial bit vectors $\boldsymbol{x}_{n-1}$ are examined, and only the M extensions which maximize the partial metric $\mu(\boldsymbol{x}_n)$ are kept. In other words, the M-algorithm[2] [10] is applied to compute $\{\mu(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{L}'}$ for $\mathcal{L}' \approx \mathcal{L}$. Note that $\mathcal{L}' \cap \{\boldsymbol{x} : x_k = 1\}$ or $\mathcal{L}' \cap \{\boldsymbol{x} : x_k = 0\}$ might be empty for some k, in which case $L_e(x_k|\boldsymbol{r})$ would be infinite. To prevent this situation, the LLRs are clipped to a finite value. The choice of the clipping threshold is discussed in [8].

In the sequel, we show that the metric $\mu(\boldsymbol{x}_n)$ can be updated from $\mu(\boldsymbol{x}_{n-1})$ using only $\mathcal{O}(N_b^2 N_h^2)$ operations, so that $\{\mu(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{L}'}$can be evaluated using only $\mathcal{O}(NM2^Q N_b^2 N_h^2)$ operations. The soft equalizer complexity is thus *linear* in the block length N and *quadratic* in the number of channel parameters $N_b N_h$.

C. Fast Metric Update

Writing the partial metric (14) in the form of (13) yields

$$\mu(\boldsymbol{x}_n) \;=\; -\boldsymbol{r_n}^H \boldsymbol{\Phi}_n^{-1} \boldsymbol{r_n} - \ln(\pi^{n+1} \det \boldsymbol{\Phi}_n) + \boldsymbol{l}_n^T \boldsymbol{x}_n, \tag{15}$$

where $\boldsymbol{\Phi}_n := \boldsymbol{A}_n \boldsymbol{R}_\theta \boldsymbol{A}_n^H + \sigma^2 \boldsymbol{I}_{n+1}$. In the Appendix, we derive the following fast sequential algorithm to compute $\mu(\boldsymbol{x}_{N-1})$, which can be shown to require $N(N_b^2 N_h^2 + 7N_b N_h + 8)$ multiplications.

set $\{\mu(\boldsymbol{x}_{-1}),\ \boldsymbol{\Sigma}_{-1}^{-1},\ \hat{\boldsymbol{\theta}}_{-1}\} := \{\ln \sigma^{-2},\ \sigma^{-2}\boldsymbol{R}_\theta,\ \boldsymbol{0}\}$;

for $n = 0, 1, 2, \ldots, N-1$,

$$\boldsymbol{a}_n = [s_n \boldsymbol{b}_n^H, \cdots, s_{n-N_h+1} \boldsymbol{b}_n^H]^H; \tag{16}$$

$$\boldsymbol{d}_n = \boldsymbol{\Sigma}_{n-1}^{-1} \boldsymbol{a}_n; \tag{17}$$

$$\alpha_n = (1 + \boldsymbol{a}_n^H \boldsymbol{d}_n)^{-1}; \tag{18}$$

$$\boldsymbol{\Sigma}_n^{-1} = \boldsymbol{\Sigma}_{n-1}^{-1} - \alpha_n \boldsymbol{d}_n \boldsymbol{d}_n^H; \tag{19}$$

$$\mu(\boldsymbol{x}_n) = \mu(\boldsymbol{x}_{n-1}) - \frac{\alpha_n}{\sigma^2} |r_n - \boldsymbol{a}_n^H \hat{\boldsymbol{\theta}}_{n-1}|^2 - \ln(\sigma^2 \pi/\alpha_n) + \underline{l}_n^T \underline{x}_n; \tag{20}$$

$$\hat{\boldsymbol{\theta}}_n = (\boldsymbol{I} - \alpha_n \boldsymbol{d}_n \boldsymbol{a}_n^H)\hat{\boldsymbol{\theta}}_{n-1} + (1 - \alpha_n \boldsymbol{d}_n^H \boldsymbol{a}_n) r_n \boldsymbol{d}_n; \tag{21}$$

end

D. On Pilots and Guards

While a single pilot symbol per block is sufficient to resolve the inherent channel/data phase ambiguity, we have found that the inclusion of several pilots is beneficial to the performance of the (suboptimal) tree search proposed in Section III-B. In particular, if pilots can be incorporated into the first few metrics (i.e., $\mu(\boldsymbol{x}_n)$ for small n), then M-algorithm path-pruning becomes more robust. We investigate these issues numerically in Section IV.

As mentioned earlier, the use of block zero-padding with guard length $\geq N_h - 1$ prevents inter-block interference, thereby justifying the use of decoupled block equalization.

Note that a simple modification of the M-algorithm suffices to handle the case of arbitrary pilot/guard symbols: When the M-algorithm encounters a known symbol, each surviving path is given a single (rather than 2^Q-ary) extension.

IV. Numerical Results

For the numerical experiments, Jakes method was employed to generate realizations of a WSSUS Rayleigh fading channel with uniform delay-power profile $\sigma_l^2 = 1/N_h$ and temporal autocorrelation $\rho_m = J_0(2\pi f_d T_s m)$. Here, $f_d T_s$ denotes the normalized single-sided Doppler spread and $J_0(\cdot)$ the 0^{th}-order Bessel function of the first kind. The values $f_d T_s = 0.002$ and $N_h = 3$ were assumed throughout.

The transmitter employed rate-$R = \frac{1}{2}$ irregular low density parity check (LDPC) codes with average column-weight 3, generated via the publicly available software [11]. The coded bits were mapped to QPSK symbols (i.e., $Q = 2$) and partitioned into data blocks of length N_s, each of which was merged with N_p leading pilots and $N_h - 1$ trailing zeros to form a transmission block of length $N = N_s + N_p + N_h - 1$. So that each codeword spanned $J = 64$ data blocks, $(JQN_s, RJQN_s)$-LDPC codes were employed. The block length $N = 64$ was used throughout with $N_p = 6$ pilots per block (unless otherwise noted).

The soft noncoherent equalizer used the Karhunen Lóeve (KL) BEM [5] with $N_b = 3$ to model channel variation. In other words, $b_{n,p} = [\boldsymbol{V}]_{n,p}$ for $\boldsymbol{V}$ constructed column-wise from the N_b principal eigenvectors of $\boldsymbol{R}_\theta$. The M-algorithm used the search parameter $M = 64$, where the LLR magnitudes were clipped to 2.3. The publicly available LDPC decoder from [11] was used with a maximum of 60 "inner" iterations, and equalization/decoding were iterated using a maximum of 16 "outer" (or "turbo") iterations. We specify the *maximum* number of iterations because the receiver breaks out of both the inner and outer loops as soon as the LDPC syndrome check indicates error-free decoding.

A. Effect of Equalizer Parameters and Pilots

Figure 2 shows coded BER under different choices of the M-algorithm search parameter: $M \in \{16, 32, 64, 128\}$. The figure shows that performance increases with M, although gains from the use of $M > 64$ are quite small (e.g., ≈ 0.1 dB).

Figure 3 shows coded BER versus the maximum number of outer (i.e., turbo) iterations. There, performance is seen to increase until about 12 iterations, after which it saturates. Note that receiver complexity does not increase linearly with the number of outer (or inner) iterations, because in the vast majority of cases the iterations terminate early.

Figure 4 shows the effect of N_p, the number of pilots per block, on coded BER. As predicted in Section III-D,

[2]Other types of tree search could also be applied. However, unlike most other search algorithms, the M-algorithm yields a complexity that is invariant to channel realization and SNR.

performance increases with N_p until about $N_p = 6$, after which it saturates. We reason that, at the saturation point, the improvement in channel estimation error is balanced by the penalty on E_b/N_o.

B. Performance Comparison

In Fig. 5, the proposed soft noncoherent equalizer was compared to two genie-aided bounds, to a *hard* noncoherent equalization scheme, and to a soft *coherent* equalization scheme aided by soft channel estimates.

For the first genie-aided bound, perfect channel knowledge was assumed. Note that, with perfect channel knowledge, the proposed soft noncoherent scheme reduces to the soft coherent scheme of [8] (which also evaluates max-log LLRs using the M-algorithm). For the second genie-aided bound, a 100% pilot-block (i.e., $N_p = 62$) was used to generate a MMSE channel estimate that was subsequently used by the soft coherent equalizer [8]. Figure 5 shows that the proposed equalizer[3] performs about 2 dB from the perfectly-known channel bound and about 1.6 dB from the 100%-pilot bound.

The reference hard noncoherent equalization scheme used the list-Viterbi algorithm (LVA) with per-survivor Kalman-filter generated channel estimates, similar to [13]. To convert the hard LVA bit estimates into the LLRs needed for soft decoding, we considered a binary symmetric channel whose cross-over probability was matched to the experimentally measured uncoded BER (at each E_b/N_o). Since the LVA does not provide a means to incorporate soft decoder outputs, outer iteration was not used. We used $N_p = 9$ (after observing poor performance with fewer pilots) and an LVA list size of 64. Figures 3 and 5 show that the proposed soft noncoherent equalizer outperforms the hard noncoherent equalizer by ≈ 1 dB *without* outer iteration, and ≈ 2 dB with outer iteration.

As a reference soft channel estimator, we considered the soft Kalman approach of [14]. Fig. 5 shows that the proposed equalizer exhibits ≈ 2.4 dB gain over the combination of soft channel estimation and soft coherent decoding.

APPENDIX

We first derive a fast recursion for $\mu_1(\boldsymbol{x}_n) := \boldsymbol{r_n}^H \boldsymbol{\Phi}_n^{-1} \boldsymbol{r_n}$. Rewriting $\boldsymbol{\Phi}_n$ with the aid of $\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{A}_{n-1} \\ \boldsymbol{a}_n^H \end{bmatrix}$, where $\boldsymbol{a}_n^H$ denotes the n^{th} row of $\boldsymbol{A}$, we have

$$\boldsymbol{\Phi}_n^{-1} = \begin{bmatrix} \boldsymbol{\Phi}_{n-1} & \boldsymbol{A}_{n-1}\boldsymbol{R}_\theta \boldsymbol{a}_n \\ \boldsymbol{a}_n^H \boldsymbol{R}_\theta \boldsymbol{A}_{n-1}^H & \boldsymbol{a}_n^H \boldsymbol{R}_\theta \boldsymbol{a}_n + \sigma^2 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{P}_n & \boldsymbol{p}_n \\ \boldsymbol{p}_n^H & p_n \end{bmatrix}, \quad (22)$$

for the block-inverse quantities

$$\boldsymbol{P}_n := \boldsymbol{\Phi}_{n-1}^{-1} + p_n^{-1} \boldsymbol{p}_n \boldsymbol{p}_n^H \quad (23)$$

$$\boldsymbol{p}_n := -\boldsymbol{\Phi}_{n-1}^{-1} \boldsymbol{A}_{n-1} \boldsymbol{R}_\theta \boldsymbol{a}_n p_n \quad (24)$$

$$p_n^{-1} := \sigma^2 + \boldsymbol{a}_n^H \left(\boldsymbol{R}_\theta - \boldsymbol{R}_\theta \boldsymbol{A}_{n-1}^H \boldsymbol{\Phi}_{n-1}^{-1} \boldsymbol{A}_{n-1} \boldsymbol{R}_\theta \right) \boldsymbol{a}_n. \quad (25)$$

[3]It is interesting to note that the asymptotic slope of the BER curves suggests that the algorithm captures the full diversity of the noncoherent DD channel [12]. In particular, with codewords spanning $JN = 4096$ symbols and a channel coherence time of $(\boldsymbol{f}_\mathrm{d} T_s)^{-1} = 500$ symbols, the codeword experiences 8 coherence intervals for each of the $N_h = 3$ channel taps, or 24 degrees of channel freedom in total. The asymptotic BER slopes have a slope of about 24.

Writing $\mu_1(\boldsymbol{x}_n)$ using (22) and $\boldsymbol{r}_n = \left[\begin{smallmatrix} \boldsymbol{r}_{n-1} \\ r_n \end{smallmatrix} \right]$, we get

$$\mu_1(\boldsymbol{x}_n) = \boldsymbol{r}_{n-1}^H \boldsymbol{P}_n \boldsymbol{r}_{n-1} + 2\Re\{\boldsymbol{r}_{n-1}^H \boldsymbol{p}_n r_n\} + p_n |r_n|^2. \quad (26)$$

The MMSE estimate of $\boldsymbol{\theta}$ from $\boldsymbol{r}_{n-1}$ conditioned on $\boldsymbol{s}_{n-1}$:

$$\hat{\boldsymbol{\theta}}_{n-1} = \mathrm{E}\{\boldsymbol{\theta} \boldsymbol{r}_{n-1}^H | \boldsymbol{s}_{n-1}\} \mathrm{E}\{\boldsymbol{r}_{n-1} \boldsymbol{r}_{n-1}^H | \boldsymbol{s}_{n-1}\}^{-1} \boldsymbol{r}_{n-1} \quad (27)$$

$$= \boldsymbol{R}_\theta \boldsymbol{A}_{n-1}^H \boldsymbol{\Phi}_{n-1}^{-1} \boldsymbol{r}_{n-1}, \quad (28)$$

allows us to write $\boldsymbol{r}_{n-1}^H \boldsymbol{p}_n = -\hat{\boldsymbol{\theta}}_{n-1} \boldsymbol{a}_n p_n$, which can be combined with (23)-(25) to express (26) as

$$\mu_1(\boldsymbol{x}_n) = \boldsymbol{r}_{n-1}^H \boldsymbol{\Phi}_{n-1}^{-1} \boldsymbol{r}_{n-1} + p_n \hat{\boldsymbol{\theta}}_{n-1}^H \boldsymbol{a}_n \boldsymbol{a}_n^H \hat{\boldsymbol{\theta}}_{n-1} + 2p_n \Re\{\hat{\boldsymbol{\theta}}_{n-1}^H \boldsymbol{a}_n r_n\} + p_n |r_n|^2 \quad (29)$$

$$= \mu_1(\boldsymbol{x}_{n-1}) + p_n \cdot |r_n - \boldsymbol{a}_n^H \hat{\boldsymbol{\theta}}_{n-1}|^2. \quad (30)$$

For a fast update of $\mu_1(\boldsymbol{x}_n)$, we need fast updates of p_n and $\hat{\boldsymbol{\theta}}_n$. For p_n, the matrix inversion lemma (MIL) implies

$$\boldsymbol{R}_\theta - \boldsymbol{R}_\theta \boldsymbol{A}_{n-1}^H \boldsymbol{\Phi}_{n-1}^{-1} \boldsymbol{A}_{n-1} \boldsymbol{R}_\theta = \sigma^2 (\underbrace{\boldsymbol{A}_{n-1}^H \boldsymbol{A}_{n-1} + \sigma^2 \boldsymbol{R}_\theta^{-1}}_{:= \boldsymbol{\Sigma}_{n-1}})^{-1} \quad (31)$$

from which (25) gives

$$p_n^{-1} = \sigma^2 (1 + \boldsymbol{a}_n^H \boldsymbol{\Sigma}_{n-1}^{-1} \boldsymbol{a}_n). \quad (32)$$

Because $\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_{n-1} + \boldsymbol{a}_n \boldsymbol{a}_n^H$, a second application of the MIL yields $\boldsymbol{\Sigma}_n^{-1} = \boldsymbol{\Sigma}_{n-1}^{-1} - \alpha_n \boldsymbol{d}_n \boldsymbol{d}_n^H$ for $\boldsymbol{d}_n := \boldsymbol{\Sigma}_{n-1}^{-1} \boldsymbol{a}_n$ and $\alpha_n := (1 + \boldsymbol{a}_n^H \boldsymbol{d}_n)^{-1} = p_n \sigma^2$. Together, this gives a fast update of $p_n = \alpha_n / \sigma^2$.

For $\hat{\boldsymbol{\theta}}_n$, a third application of the MIL gives $\boldsymbol{\Phi}_n^{-1} = \sigma^{-2}(\boldsymbol{I} - \boldsymbol{A}_n \boldsymbol{\Sigma}_n^{-1} \boldsymbol{A}_n^H)$, which applied to (28) yields

$$\hat{\boldsymbol{\theta}}_n = \sigma^{-2} \boldsymbol{R}_\theta \big(\boldsymbol{\Sigma}_n \boldsymbol{\Sigma}_n^{-1} \boldsymbol{A}_n^H - \boldsymbol{A}_n^H \boldsymbol{A}_n \boldsymbol{\Sigma}_n^{-1} \boldsymbol{A}_n^H \big) \boldsymbol{r}_n \quad (33)$$

$$= \sigma^{-2} \boldsymbol{R}_\theta \big(\boldsymbol{\Sigma}_n - \boldsymbol{A}_n^H \boldsymbol{A}_n \big) \boldsymbol{\Sigma}_n^{-1} \boldsymbol{A}_n^H \boldsymbol{r}_n \quad (34)$$

$$= \boldsymbol{\Sigma}_n^{-1} \boldsymbol{A}_n^H \boldsymbol{r}_n \quad (35)$$

$$= \big(\boldsymbol{\Sigma}_{n-1}^{-1} - \alpha_n \boldsymbol{d}_n \boldsymbol{d}_n^H \big) \big(\boldsymbol{A}_{n-1}^H \boldsymbol{r}_{n-1} + \boldsymbol{a}_n r_n \big) \quad (36)$$

$$= \big(\boldsymbol{I}_{N_h N_b} - \alpha_n \boldsymbol{d}_n \boldsymbol{a}_n^H \big) \hat{\boldsymbol{\theta}}_{n-1} + \big(1 - \alpha_n \boldsymbol{d}_n^H \boldsymbol{a}_n \big) r_n \boldsymbol{d}_n \quad (37)$$

giving a fast update for $\hat{\boldsymbol{\theta}}_n$.

A fast recursion for $\mu_2(\boldsymbol{x}_n) := \ln\big(\pi^{n+1} \det \boldsymbol{\Phi}_n\big)$ is:

$$\mu_2(\boldsymbol{x}_n) = \ln \Big\{ (\sigma^2 \pi)^{n+1} \det\big(\sigma^{-2} \boldsymbol{R}_\theta^{\frac{1}{2}} \boldsymbol{A}_n^H \boldsymbol{A}_n \boldsymbol{R}_\theta^{\frac{1}{2}} + \boldsymbol{I}_{N_h N_b} \big) \Big\} \quad (38)$$

$$= \ln \big\{ (\sigma^2 \pi)^{n+1} \det\big(\sigma^{-2} \boldsymbol{R}_\theta^{\frac{1}{2}} \boldsymbol{a}_n \boldsymbol{a}_n^H \boldsymbol{R}_\theta^{\frac{1}{2}} + \sigma^{-2} \boldsymbol{R}_\theta^{\frac{1}{2}} \boldsymbol{A}_{n-1}^H \boldsymbol{A}_{n-1} \boldsymbol{R}_\theta^{\frac{1}{2}} + \boldsymbol{I}_{N_h N_b} \big) \big\} \quad (39)$$

$$= \ln \big\{ \sigma^2 \pi \big(1 + \sigma^{-2} \boldsymbol{a}_n^H \boldsymbol{R}_\theta^{\frac{1}{2}} \times \big(\sigma^{-2} \boldsymbol{R}_\theta^{\frac{1}{2}} \boldsymbol{A}_{n-1}^H \boldsymbol{A}_{n-1} \boldsymbol{R}_\theta^{\frac{1}{2}} + \boldsymbol{I}_{N_h N_b} \big)^{-1} \boldsymbol{R}_\theta^{\frac{1}{2}} \boldsymbol{a}_n \big) \big\} + \ln \big\{ (\sigma^2 \pi)^n \det\big(\sigma^{-2} \boldsymbol{R}_\theta^{\frac{1}{2}} \boldsymbol{A}_{n-1}^H \boldsymbol{A}_{n-1} \boldsymbol{R}_\theta^{\frac{1}{2}} + \boldsymbol{I}_{N_h N_b} \big) \big\} \quad (40)$$

$$= \ln \big\{ \sigma^2 \pi (1 + \boldsymbol{a}_n^H (\boldsymbol{A}_{n-1}^H \boldsymbol{A}_{n-1} + \sigma^2 \boldsymbol{R}_\theta^{-1})^{-1} \boldsymbol{a}_n) \big\} + \ln \big\{ \pi^n \det(\boldsymbol{A}_{n-1} \boldsymbol{R}_\theta \boldsymbol{A}_{n-1}^H + \sigma^2 \boldsymbol{I}_n) \big\} \quad (41)$$

$$= \ln(\sigma^2 \pi / \alpha_n) + \mu_2(\boldsymbol{x}_{n-1}), \quad (42)$$

where (41) used $\det(\boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{B}) = (1 + \boldsymbol{x}^H \boldsymbol{B}^{-1} \boldsymbol{x}) \det(\boldsymbol{B})$.

The updates for $\mu_1(\boldsymbol{x}_n)$, $\mu_2(\boldsymbol{x}_n)$, $\hat{\boldsymbol{\theta}}_n$, and $\boldsymbol{\Sigma}_n^{-1}$, derived here are combined to form (16)-(21).

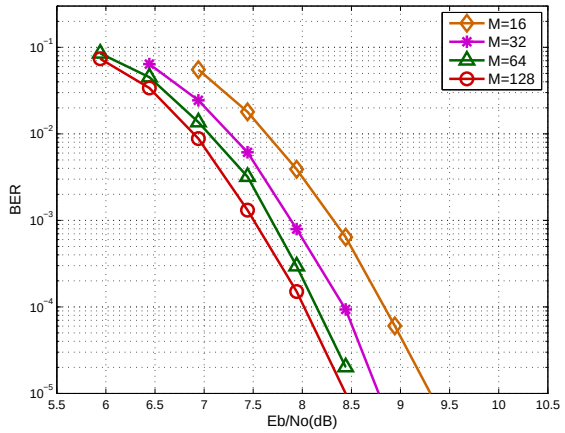

Fig. 2. Coded BER vs. E_b/N_o for M-algorithm parameter $M \in \{16, 32, 64, 128\}$.

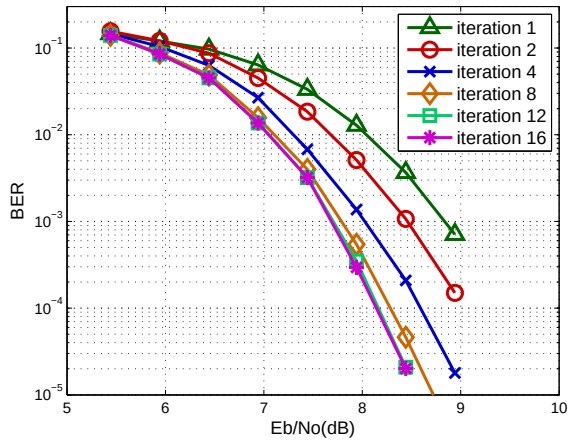

Fig. 3. Coded BER vs. E_b/N_o for different numbers of outer iterations.

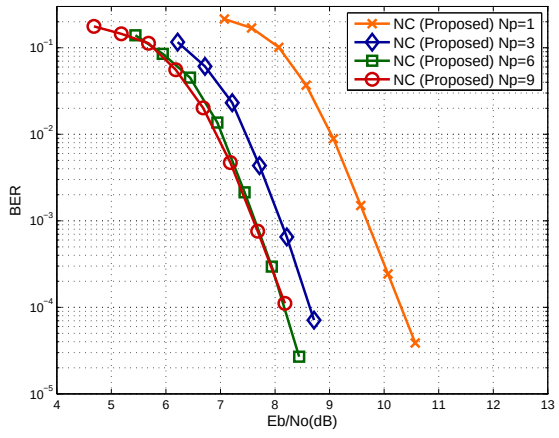

Fig. 4. Coded BER vs. E_b/N_o for different numbers of outer iterations.

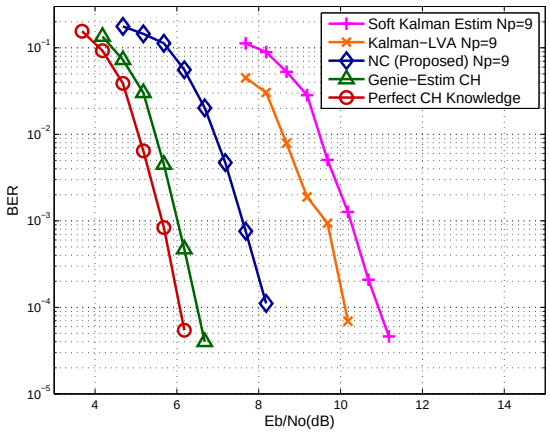

Fig. 5. Coded BER vs. E_b/N_o for proposed soft noncoherent equalizer, hard noncoherent equalizer based on Kalman-LVA, soft coherent equalization plus soft Kalman channel estimate, and soft coherent equalization with perfect channel estimate (genie-aided) and with MMSE estimate using 100% pilots (genie-aided).

REFERENCES

[1] B. D. Hart and S. Pasupathy, "Innovations-based MAP detection for time-varying frequency-selective channels," *IEEE Trans. on Vehicular Technology*, vol. 48, pp. 1507–1519, Sept. 2000.

[2] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *European Trans. on Telecommunications*, vol. 6, pp. 507–511, Sept.-Oct. 1995.

[3] M. Nissilä and S. Pasupathy, "Adaptive Bayesian and EM-based detectors for frequency-selective fading channels," *IEEE Trans. on Communications*, vol. 51, pp. 1325–1336, Aug. 2003.

[4] M. K. Tsatsanis and G. B. Giannakis, "Modeling and equalization of rapidly fading channels," *Internat. Journal of Adaptive Control & Signal Processing*, vol. 10, pp. 159–176, Mar. 1996.

[5] D. K. Borah and B. D. Hart, "Frequency-selective fading channel estimation with a polynomial time-varying channel model," *IEEE Trans. on Communications*, vol. 47, pp. 862–873, June 1999.

[6] S.-J. Hwang and P. Schniter, "Efficient communication over highly spread underwater acoustic channels," in *Proc. ACM International Workshop on Underwater Networks (WUWNet)*, Sept. 2007.

[7] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Communications*, vol. 51, pp. 389–399, Mar. 2003.

[8] Y. L. C. de Jong and T. J. Willink, "Iterative tree search detection for MIMO wireless systems," *IEEE Trans. on Communications*, vol. 53, pp. 930–935, June 2005.

[9] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 4th ed., 2001.

[10] J. B. Anderson and S. Mohan, "Sequential decoding algorithms: A survey and cost analysis," *IEEE Trans. on Communications*, vol. 32, pp. 169–172, 1984.

[11] I. Kozintsev, "Matlab programs for encoding and decoding of LDPC codes over GF(2^m)." http://www.kozintsev.net/soft.html.

[12] S.-J. Hwang and P. Schniter, "Maximum-diversity affine precoding for the noncoherent doubly dispersive channel," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communication*, pp. 1–5, June 2007.

[13] H. Chen, K. Buckley, and R. Perry, "Time-recursive maximum likelihood based sequence estimation for unknown ISI channels," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, pp. 1005–1009, 2000.

[14] S. Song, A. C. Singer, and K.-M. Sung, "Soft input channel estimation for turbo equalization," *IEEE Trans. on Signal Processing*, vol. 52, pp. 2885–2894, Oct. 2004.