Turbo Equalization for Doubly-Selective Fading Channels Using Nonlinear Kalman Filtering and Basis Expansion Models

Hyosung Kim and Jitendra K. Tugnait, *Fellow, IEEE*

***Abstract*—We present a turbo (iterative) equalization receiver with fixed-lag nonlinear Kalman filtering for coded data transmission over doubly-selective channels. The proposed receiver exploits the complex exponential basis expansion model (CE-BEM) for the overall channel variations, and an autoregressive (AR) model for the BEM coefficients. We extend an existing turbo equalization approach based on symbol-wise AR modeling of channels to channels based on BEM's. In the receiver an adaptive equalizer using nonlinear Kalman filters with delay is coupled with a soft-input soft-output (SISO) decoder to iteratively perform equalization and decoding. The adaptive equalizer jointly optimizes the estimates of the BEM coefficients and data symbols, thereby automatically accounting for correlation between data symbols and channel tap gains. An extrinsic information transfer (EXIT) chart analysis of the proposed approach is also presented. Simulation examples demonstrate that our CE-BEM-based approach significantly outperforms the existing symbol-wise AR model-based turbo equalizer.**

***Index Terms*—Basis expansion models, doubly-selective channels, turbo equalization, extended Kalman filter, channel estimation, iterative decoding.**

I. Introduction

DUE to multipath propagation and Doppler spread, wireless channels are characterized by frequency- and time-selectivity. Accurate modeling of time-variations of the channel plays a crucial role in channel estimation and data detection. Among various models for channel time-variations, the autoregressive (AR) process, particularly the first-order AR model, is regarded as a tractable formulation to describe a time-varying channel on a symbol-by-symbol basis [1], [5], [13]. In fast time-varying channel environments, however, channel prediction using an AR model may lead to high estimation variance resulting in erroneous symbol decisions [5]. Basis expansion models (BEM) depict evolutions of the channel over a period (block) of time, in which the time-varying channel taps are expressed as superpositions of time-varying basis functions in modeling Doppler effects, weighted by time-invariant coefficients [3].

Manuscript received June 18, 2009; revised June 24, 2009 and February 7, 2010; accepted April 20, 2010. The associate editor coordinating the review of this paper and approving it for publication was C. Xiao.

This work was supported by NSF under Grant ECCS-0823987.

The authors are with the Department of Electrical & Computer Engineering, 200 Broun Hall, Auburn University, Auburn, AL 36849, USA (e-mail: tugnajk@eng.auburn.edu, kimhyos@auburn.edu).

The material in this paper was presented in part at the 43rd Annual Conf. on Information Sciences & Systems, March 2009, Johns Hopkins University, Baltimore, MD.

Digital Object Identifier 10.1109/TWC.2010.06.090920

In [7], a subblock-wise tracking approach was proposed for doubly-selective channels using time-multiplexed (TM) training. It exploits the oversampled complex exponential BEM (CE-BEM) [17] for the overall channel variations of each (overlapping) block, and a first-order AR model to describe the evolutions of the BEM coefficients. Since the time-varying nature of the channel can be well captured in the CE-BEM by (known) Fourier basis functions, the time-variations of the (unknown) BEM coefficients are likely much slower than those of the channel, and thus more convenient to track in fast-fading environments [7]. The slow-varying BEM coefficients are updated via Kalman filtering at each training session; during information sessions, channel estimates are generated by the CE-BEM using the estimated BEM coefficients [7]. This approach achieves better performance in fast-fading environments, than using conventional symbol-wise AR models [7].

In this paper we extend the approach of [7] to coded modulation communication systems using turbo equalization receivers. Turbo (iterative) equalization is an iterative equalization and decoding approach used in place of the computationally prohibitive but optimal joint maximum likelihood (ML) or maximum a posteriori (MAP) equalization and decoding. Although originally proposed for parallel concatenated error correction codes [8], the turbo principle is shown to be applicable to the detection problem for coded systems with intersymbol interference (ISI) in [9]. By combining a MAP equalizer and a MAP decoder, and exchanging probabilistic information about data symbols iteratively, turbo equalization usually can achieve close-to-optimal performance but with much lower complexity [9]. In [10], a turbo-equalization-like system using linear equalizers based on soft interference cancellation and linear minimum mean-square error (MMSE) filtering is proposed as part of a multiuser detector for code division multiple access (CDMA) systems. Based on these works, a variety of soft-input soft-output (SISO) equalizers employing linear MMSE and decision feedback equalization (DFE) are proposed in [11] and [12]. For doubly-selective channels an adaptive SISO equalizer has been presented in [13], using an extended Kalman filter (EKF) to incorporate channel estimation into the equalization process. This adaptive soft nonlinear Kalman equalizer takes the soft decisions of data symbols from the SISO decoder as its *a priori* information, and performs equalization in each iteration. This approach jointly optimizes the estimates of the channel and data symbols in each iteration. This avoids the common draw-

back in separate channel estimation and equalization/detection approaches in that the correlation between channel estimates and data symbol decision is considered. The complexity of [13] is comparable to that of the turbo equalizers using linear filters [14], [15], [29], and is usually much lower than that of the ML/MAP based joint channel estimation and data detection schemes.

In this paper, based on the turbo approach proposed in [13] and the BEM-based approach of [7], we present an adaptive turbo equalizer with fixed-lag nonlinear Kalman filtering, based on the CE-BEM. The channel variations can be well captured by the CE-BEM since the time-variations of the BEM coefficients are likely much slower than those of real channels. The adaptive SISO equalizer takes the decision of data symbols provided by SISO decoder as its *a priori* information and the performance can be improved iteratively. The proposed adaptive equalizer jointly optimizes the estimates of channel BEM coefficients and data symbols in each iteration of the equalization process; thus, correlation between the estimates of the channel and data symbols is automatically considered. An extrinsic information transfer (EXIT) chart analysis of the proposed approach is also provided. Simulation examples demonstrate our CE-BEM based scheme has superior performance over the turbo equalizer in [13] that relies on the AR modeling of the channel*.

The rest of the paper is organized as follows. Sec. II introduces the coded modulation system and the channel model, including the AR and the CE-BEM representations. The turbo equalizer receiver structure is the subject of Sec. III. We then discuss the SISO nonlinear Kalman equalizer in Sec. IV. Simulation examples are presented in Sec. V. An EXIT chart analysis is presented in Sec. VI, and Sec. VII concludes the paper.

Notations: Superscripts $*$, T, H and $\dagger$ denote the complex conjugation, transpose, complex conjugate transpose, and Moore-Penrose pseudo-inverse, respectively. $\mathbf{I}_N$ is the $N \times N$ identity matrix, $\mathrm{tr}\,(\mathbf{A})$ is the trace of a square matrix $\mathbf{A}$, $\mathbf{0}_{M\times N}$ is the $M \times N$ null matrix and $\otimes$ denotes the Kronecker product. We use $\lceil\cdot\rceil$ for integer ceiling and $\lfloor\cdot\rfloor$ for integer floor. The symbol $E\{\cdot\}$ or $E[\cdot]$ denotes expectation, $\delta(\tau)$ is the Kronecker delta function, that is, $\delta(\tau) = 1$ for $\tau = 0$, and $\delta(\tau) = 0$ otherwise, and $\mathbf{x}^i$ (also x^i) denotes the ith component of vector $\mathbf{x}$.

II. System and Channel Models

A. Bit-Interleaved Coded Modulation (BICM) and Received Signal

We consider a BICM transmitter (as in [18]) as shown in Fig. 1. A sequence of independent data bits $\in \{1, 0\}$ are collected into blocks of length k_0 as $\mathbf{b}(n') = [b^1(n'), b^2(n'), \cdots, b^{k_0}(n')] \in \{1,0\}^{k_0}$ at time n'. The sequence $\mathbf{b}(n')$ is fed into a convolutional encoder with a code rate $R_c = k_0/n_0$. The coded output $\mathbf{c}(n') = [c^1(n'), c^2(n'), \cdots, c^{n_0}(n')] \in \{1,0\}^{n_0}$ is passed through a bit-wise random interleaver Π, generating the interleaved coded bit sequence $\mathbf{c}(n) = \Pi[\mathbf{c}(n')]$. The binary coded bits are then mapped to a data signal sequence $d(n)$ over a 2-dimensional signal constellation χ of cardinality $M = 2^m$ by an M-ary modulator with an one-to-one binary map $\mu : \{0,1\}^m \to \chi$. In this paper, we only consider the case of phase-shift keying (PSK) or quadrature amplitude modulation (QAM) with the average energy of the constellation χ constrained to be unity. That is, the signal $d(n)$ drawn from χ has mean $E[d(n)] = 0$ and variance $E[|d(n)|^2] = 1$. After modulation, we periodically insert short training sequences into the data symbol sequence. The training symbols $t(n)$, which are known to the receiver, are randomly drawn from the signal constellation χ with equal probabilities. The symbol $s(n)$ will be used throughout to denote both $d(n)$ and $t(n)$.

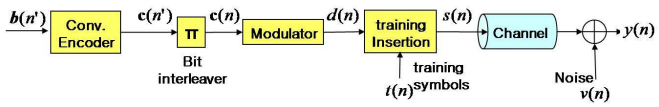


Fig. 1. Bit-interleaved coded modulation system model.

Further consider a doubly-selective (frequency- and time-selective) FIR linear channel with discrete-time impulse response $\{h(n;l)\}$ (channel response at time n to a unit input at time $n-l$). With $\{s(n)\}$ as the scalar input sequence, the symbol-rate noisy channel output is given by $(n = 0, 1, \ldots)$

$$y(n) = \sum_{l=0}^{L} h(n;l)\, s(n-l) + v(n) \tag{1}$$

where $v(n)$ is zero-mean white complex Gaussian noise with variance σ_v^2. We assume that $\{h(n;l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [2].

B. Channel Models

1) Autoregressive (AR) Model for Channel Variations: It is possible to accurately represent a WSSUS channel by a large order AR model; see [5], [13], [31] and references therein. Let

$$\tilde{\mathbf{h}}(n) := [h(n;0)\; h(n;1)\; \cdots\; h(n;L)]^T \tag{2}$$

where $\tilde{\mathbf{h}}(n)$ is $(L+1) \times 1$. Then a Pth order AR model, AR(P), for $\tilde{\mathbf{h}}(n)$ is given by

$$\tilde{\mathbf{h}}(n) = \sum_{i=1}^{P} \mathbf{A}_i \tilde{\mathbf{h}}(n-i) + \mathbf{G}_0 \tilde{\mathbf{w}}(n) \tag{3}$$

where $\mathbf{A}_i$'s are the $(L+1)\times(L+1)$ AR coefficient matrices, $\mathbf{G}_0$ is also $(L+1)\times(L+1)$ and the i.i.d. $(L+1)\times 1$ driving noise sequence $\tilde{\mathbf{w}}(n)$ is zero-mean with identity covariance matrix. Suppose that we know the correlation function $\mathbf{R}_h(\tau) = E\{\tilde{\mathbf{h}}(n+\tau)\tilde{\mathbf{h}}^H(n)\}$ for lags $\tau = 0, 1, \cdots, P$. The following Yule-Walker equation holds for (3) [19]:

$$\mathbf{R}_h(\tau) = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{R}_h(\tau - i) + \mathbf{G}_0 \mathbf{G}_0^H \delta(\tau). \tag{4}$$

Using (4) for $\tau = 1, 2, \cdots, P$, and the fact that $\mathbf{R}_h(-\tau) = \mathbf{R}_h^H(\tau)$, one can estimate $\mathbf{A}_i$'s. Using the estimated $\mathbf{A}_i$'s and (4) for $\tau = 0$ one can find $\mathbf{G}_0\mathbf{G}_0^H$, from which one can find

*It has been shown in [13] that their approach has better performance than the turbo approaches of [14], [15], [29]; hence we compare our approach only with [13].

(nonunique) $\mathbf{G}_0$ by computing its "square root" [20, p. 358]. As noted in [5, Sec. II-B], this procedure amounts to matching the first $P+1$ lags of the autocorrelation function of $\tilde{\mathbf{h}}(n)$. In [5], [13] only AR(1) or AR(2) models have been used whereas [31] discusses fitting large order (several tens) AR models. Later we will use AR models for some simulation comparisons where various channel tap gains are assumed to be mutually statistically independent. In this case we have an independent AR process for each channel tap gain. Given the channel correlation function, for AR model fitting, we followed [31] where it is noted that for "large" order AR model fitting, one may encounter numerical ill-conditioning requiring regularization of a certain correlation matrix inverse. It must be noted that in practice, one would not know $\mathbf{R}_h(\tau)$.

2) Complex Exponential Basis Expansion Model (CE-BEM): In contrast to the symbol-wise AR model, a BEM assigns temporal variations to basis functions [4]. Suppose that we collect the received signal over a time interval of T_r symbols. [Referring to Fig. 1 and Sec. II-A, T_r would be the interleaver size in symbols plus the number of inserted training symbols.] In the CE-BEM [3], [4], [17], the T_r symbols are divided into nonoverlapping blocks of T_B symbols and over the kth block, the channel impulse response is represented by (for $n = (k-1)T_B, (k-1)T_B+1, \ldots, kT_B-1$ and $l = 0, 1, \ldots, L$)

$$h(n;l) = \sum_{q=1}^{Q} h_q(l)\, e^{j\omega_q n}, \tag{5}$$

where the BEM coefficients $h_q(l)$'s remain invariant during each block, but are allowed to change at the next block, the Fourier basis functions $\left\{e^{j\omega_q n}\right\}$ are common for every block, one chooses ($q = 1, 2, \ldots, Q$ and $K \geq 1$ is an integer)

$$T_p := KT_B, \quad Q \geq 2\lceil f_d T_p T_s \rceil + 1, \tag{6}$$

$$\omega_q := \frac{2\pi}{T_p}\left[q - (Q+1)/2\right], \quad L := \lfloor \tau_d / T_s \rfloor, \tag{7}$$

τ_d and f_d are the delay spread and the Doppler spread, respectively, and T_s is the symbol duration. If the delay spread and the Doppler spread of the channel (or at least their upper bounds) are known, one can infer the basis functions of the CE-BEM [3]. Treating the basis functions as known parameters, estimation of a time-varying process is reduced to estimating the invariant coefficients over a block of T_B symbols. Note that the BEM period is $T_p = KT_B$ whereas the block size is T_B symbols. If $K > 1$ (e.g. $K = 2$ or $K = 3$), then the Doppler spectrum is said to be over-sampled [17] compared to the case $K = 1$ where the Doppler spectrum is said to be critically sampled. In [3], [4] only $K = 1$ (henceforth called critically-sampled CE-BEM) is considered whereas [17] considers $K \geq 2$ (henceforth called over-sampled CE-BEM).

Unlike the prior works [3], [4], [17], we will now allow the blocks of T_B symbols to overlap. By exploiting the invariance of the coefficients of the CE-BEM over each block, we consider two overlapping blocks (each of T_B symbols) that differ by just one symbol: the "past" block beginning at time n_0 and the "present" block beginning at time n_0+1. Since the two blocks overlap so significantly, one would expect the BEM coefficients to vary only "a little" from the past block to the present overlapping one. We propose to track the BEM coefficients (rather than the channel tap gains) symbol-by-symbol using a first-order AR model for their variations, where we will use (5) for all times n, not just the particular block of size T_B symbols, by allowing the coefficients $h_q(l)$'s to change with time.

Stack the channel coefficients in (5) into a $Q(L+1)\times 1$ vector

$$\mathbf{h} := \big[\; h_1(0) \quad h_2(0) \quad \cdots \quad h_Q(0) \quad h_1(1) \quad \cdots \quad h_Q(1) \quad \cdots \quad h_1(L) \quad h_2(L) \quad \cdots \quad h_Q(L) \;\big]^T. \tag{8}$$

We will allow $\mathbf{h}$ in (8) to change with "time" n, in which case it will be denoted by $\mathbf{h}(n)$. We assume that the channel BEM coefficients follow an AR model. One could fit a general AR(P) model with a high value of P (as in Sec. II-B1 for channel variations), but we seek a "simple" AR(1) model given by

$$\mathbf{h}(n) = \mathbf{A}_1 \mathbf{h}(n-1) + \mathbf{G}_0 \mathbf{w}(n) \tag{9}$$

where $\mathbf{A}_1$ is the time-invariant AR coefficient matrix and the driving noise vector $\mathbf{w}(n)$ is zero-mean white with identity covariance. Collecting all channel tap gains over one block, further define the $[(L+1)T_B]\times 1$ vector

$$\mathbf{g}(n) := \big[\; h(n;0) \quad h(n-1;0) \quad \cdots \quad h(n-T_B+1;0) \quad h(n;1) \quad h(n-1;1) \quad \cdots \quad h(n-T_B+1;1) \quad \cdots \quad h(n;L) \quad \cdots \quad h(n-T_B+1;L) \;\big]^T. \tag{10}$$

Define

$$\mathbf{b}_{ex}(n) := \left[e^{-j\omega_1 n} \quad e^{-j\omega_2 n} \quad \cdots \quad e^{-j\omega_Q n}\right]^T. \tag{11}$$

Using (11), we further define

$$\mathbf{B}(n) := \left[\mathbf{b}_{ex}(n) \quad \mathbf{b}_{ex}(n-1) \quad \cdots \quad \mathbf{b}_{ex}(n-T_B+1)\right]^H, \tag{12}$$

$$\mathbf{\Gamma} := \mathrm{diag}\left\{e^{j\omega_1}, e^{j\omega_2}, \ldots, e^{j\omega_Q}\right\},$$

where $\mathbf{B}(n)$ is $T_B \times Q$ and $\mathbf{\Gamma}$ is $Q \times Q$. Consider two overlapping blocks that differ by just one symbol: $\mathbf{g}(n)$ and $\mathbf{g}(n+1)$, with $\mathbf{h}(n)$ and $\mathbf{h}(n+1)$, respectively, as the corresponding BEM coefficients. It then follows that

$$\mathbf{g}(n) = \bar{\mathbf{B}}(n)\mathbf{h}(n), \quad \mathbf{g}(n+1) = \bar{\mathbf{B}}(n)\bar{\mathbf{\Gamma}}\mathbf{h}(n+1) \tag{13}$$

where ($\bar{\mathbf{B}}$ is $[(L+1)T_B]\times[(L+1)Q]$ and $\bar{\mathbf{\Gamma}}$ is $[(L+1)Q]\times[(L+1)Q]$)

$$\bar{\mathbf{B}}(n) := \mathrm{diag}\{\mathbf{B}(n),\ \mathbf{B}(n),\ \cdots,\ \mathbf{B}(n)\}, \tag{14}$$

$$\bar{\mathbf{\Gamma}} := \mathrm{diag}\{\mathbf{\Gamma},\ \mathbf{\Gamma},\ \cdots,\ \mathbf{\Gamma}\}.$$

If (9) holds, then using the Yule-Walker equation we have

$$\mathbf{A}_1 = E\left\{\mathbf{h}(n+1)\mathbf{h}^H(n)\right\}\left[E\left\{\mathbf{h}(n)\mathbf{h}^H(n)\right\}\right]^{-1} \tag{15}$$

where using (13) we have

$$E\left\{\mathbf{h}(n+1)\mathbf{h}^H(n)\right\} = \bar{\mathbf{\Gamma}}^{-1}\bar{\mathbf{B}}^{\dagger}(n) E\left\{\mathbf{g}(n+1)\mathbf{g}^H(n)\right\}\bar{\mathbf{B}}^{\dagger H}(n), \tag{16}$$

$$E\left\{\mathbf{h}(n)\mathbf{h}^H(n)\right\} = \bar{\mathbf{B}}^{\dagger}(n)E\left\{\mathbf{g}(n)\mathbf{g}^H(n)\right\}\bar{\mathbf{B}}^{\dagger H}(n), \quad (17)$$

and $E\left\{\mathbf{g}(n)\mathbf{g}^H(n)\right\}$ and $E\left\{\mathbf{g}(n+1)\mathbf{g}^H(n)\right\}$ can be calculated using (10) if we know the channel correlation function $\mathbf{R}_h(\tau)$ (as defined in Sec. II-B1). As in Sec. II-B1, this procedure results in matching the correlation function of $\mathbf{h}\,(n)$ at lags 0 and 1.

Typically $\mathbf{R}_h(\tau)$ will not be available. Therefore, to simplify we will assume that $\mathbf{A}_1 = \alpha\mathbf{I}$ (implying that all tap gains have the same Doppler spectrum), $\mathbf{G}_0 = \mathbf{I}_{Q(L+1)}$ and $E\{\mathbf{w}(n)\mathbf{w}^H(n)\} = \sigma_w^2\mathbf{I}_{Q(L+1)}$, leading to

$$\mathbf{h}\,(n) = \alpha\mathbf{h}\,(n-1) + \mathbf{w}\,(n)\,. \quad (18)$$

If the channel is stationary (WSSUS) and coefficients $h_q(l)$'s are independent (as assumed in [3]), then by (18) and Yule-Walker equations, we can estimate α as

$$\alpha = \left(\frac{E\left\{\mathbf{h}^H(n+1)\mathbf{h}(n)\right\}}{E\left\{\mathbf{h}^H(n)\mathbf{h}(n)\right\}}\right)^*$$

$$= \frac{\mathrm{tr}\left\{\bar{\mathbf{\Gamma}}^{-1}\bar{\mathbf{B}}^{\dagger}(n)E\left\{\mathbf{g}(n+1)\mathbf{g}^H(n)\right\}\bar{\mathbf{B}}^{\dagger H}(n)\right\}}{\mathrm{tr}\left\{\bar{\mathbf{B}}^{\dagger}(n)E\left\{\mathbf{g}(n)\mathbf{g}^H(n)\right\}\bar{\mathbf{B}}^{\dagger H}(n)\right\}}, \quad (19)$$

and for a uniform power delay profile, $\sigma_w^2 = E\{|h(n;l)|^2\}(1-|\alpha|^2)/Q$ where $\sigma_h^2 = E\{|h(n;l)|^2\}$. Note that (19) requires knowledge of $\mathbf{R}_h(\tau)$. In order to avoid this, one can somewhat arbitrarily pick a value of α such that $\alpha \approx 1$ but $\alpha < 1$; this has been done in, e.g. [32] (in a different but similar context). Besides, for tracking, one needs $\alpha < 1$ [32]. To gain more insight, let us consider a specific channel tap $h(n;l)$ following the Jakes' spectrum (also used in Sec. V in simulation examples). When $T_p = 200, T_B = 100, Q = 5$, and $f_dT_s = 0.01$, one gets α =0.99989 using (19). We compared it with $\mathbf{A}_1$ obtained via (15)-(17), yielding the normalized difference $\|\mathbf{A}_1 - \alpha\mathbf{I}\|_F/\|\mathbf{A}_1\|_F$ =0.0095 where $\|.\|_F$ denotes the Frobenius norm. [As we will see later in Sec. V (Fig. 8), this value of α is too close to one to permit tracking; we used $\alpha = 0.996$ in Sec. V.] Thus, for channel taps following the Jakes' spectrum, $\mathbf{A}_1 = \alpha\mathbf{I}$ is an excellent choice.

Under this formulation, we do not need a "strict" definition of the block size T_B. A key parameter now is the CE-BEM period T_p, not the block size T_B. Later we use (5) for all times n, not just the particular block of size T_B symbols, by allowing the coefficients $h_q(l)$'s to change with time (symbol-wise). Note that model (5) is periodic with period T_p whereas the channel is by no means periodic. So long as the effective "memory" of the Kalman filter used later is less than the model period T_p, there are no deleterious effects due to the use of (5) for all time.

III. Receiver Structure

A turbo equalization structure, as depicted in Fig. 2, is employed in the receiver, as in [13] except that [13] uses symbol-wise AR models. The adaptive SISO equalizer is embedded into the iterative decoding (ID) process of the BICM transmission system (BICM-ID) [18]. In each decoding iteration, the equalizer takes the training symbols and the soft decision information about data symbols supplied by the SISO decoder from the previous iteration as its *a priori*

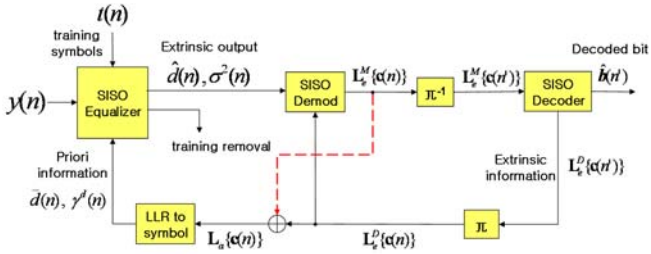


Fig. 2. Turbo-equalization receiver. Following [13], [23], [24] and contrary to the original turbo-principle, *a posteriori* LLR $\mathbf{L}_a\{\mathbf{c}(n)\} = \mathbf{L}_e^M\{\mathbf{c}(n)\} + \mathbf{L}_e^D\{\mathbf{c}(n)\}$ instead of the extrinsic LLR $\mathbf{L}_e^D\{\mathbf{c}(n)\}$ can be input to the LLR-to-symbol block. Inclusion of $\mathbf{L}_e^M\{\mathbf{c}(n)\}$ to create *a posteriori* LLR is shown via dashed line. For our proposed approach we follow [13], [23], [24]. SISO: soft-input soft-output.

information to perform joint adaptive channel estimation and equalization. The equalizer produces the soft-valued extrinsic estimate of the data symbols, which are independent of their *a priori* information. The output of the equalizer is an updated sequence of soft estimates $\hat{d}(n)$ and its error variance $\sigma^2(n)$. Using the adaptive SISO equalizer in Sec. IV-C, we have extrinsic information for the data symbols $d(n)$. The training symbols are removed at the SISO equalizer output and the iterative process that follows is only for data symbols. The SISO equalizer based on the CE-BEM is described in Sec. IV-C. The SISO demodulator follows [18] whereas the SISO decoder follows the MAP decoding algorithm ("BCJR") [22, Sec. 6.2].

The data symbol estimates $\hat{d}(n)$ and its error variance $\sigma^2(n)$ are passed to the SISO demodulator to generate extrinsic log-likelihood ratios (LLR's) $\mathbf{L}_e^M\{\mathbf{c}(n)\}$ for the coded bits $\mathbf{c}(n)$ given T_r received symbols $\{y(l),\ 0 \le l < T_r\}$, denoted by

$$\mathbf{L}_e^M\{\mathbf{c}(n)\} = \left[L_e^M\left\{c^i(n)\right\},\ i = 1, 2\cdots, n_0\right], \quad (20)$$

where T_r is the information block size after mapping the interleaved coded bits to the signal sequence,

$$L_e^M\left\{c^i(n)\right\} := \ln\frac{P\{c^i(n)=1 \mid y(l),\ 0\le l<T_r\}}{P\{c^i(n)=0 \mid y(l),\ 0\le l<T_r\}} - \underbrace{\ln\frac{P\{c^i(n)=1\}}{P\{c^i(n)=0\}}}_{=:L\{c^i(n)\}}, \quad (21)$$

and $L\{c^i(n)\}$ is the *a priori* LLR. In (21), $P\{c^i(n) = b \mid y(l),\ 0 \le l < T_r\}$, $b \in \{0,1\}$, is approximated as

$$P\{c^i(n) = b \mid y(l),\ 0 \le l < T_r\} \approx P\{c^i(n) = b \mid \hat{d}(n)\} \quad (22)$$

by replacing the data $\{y(l),\ 0 \le l < T_r\}$ with the soft estimate $\hat{d}(n)$. Since $P\{c^i(n) = b \mid \hat{d}(n)\} = P\{\hat{d}(n) \mid c^i(n) = b\}P\{c^i(n) = b\}/P\{\hat{d}(n)\}$, it follows from (21) and (22) that

$$L_e^M\left\{c^i(n)\right\} = \ln\frac{P\{\hat{d}(n) \mid c^i(n)=1\}}{P\{\hat{d}(n) \mid c^i(n)=0\}}. \quad (23)$$

The soft estimate $\hat{d}(n)$ of $d(n)$ follows from the fixed-lag SISO Kalman equalizer discussed later in Sec. IV (following [13]), and it and its variance are given by (50) and (51), respectively. We assume that $\hat{d}(n)$ is complex Gaussian distributed

with mean $d \in \mathcal{X}$ and variance $\sigma^2(n)$ and follow [18] to calculate (23). Let c_d^i denote the i-th code bit in the block of code bits $[c_d^1, c_d^2, \cdots, c_d^{n_0}]$ that is mapped to the symbol d; dropping the subscript d, we will use the notation (see Sec. II-A) $\mu\left([c^1, c^2, \cdots, c^{n_0}]\right) = d$. It then follows that

$$P\left\{d \mid c^i\right\} = \prod_{\substack{j=1,\\ j\neq i}}^{n_0} P\{c^j = c_d^j\}. \tag{24}$$

Furthermore, under the assumptions on $\hat{d}(n)$, we have

$$P\left\{\hat{d}(n) \mid d\right\} = \frac{1}{\pi\sigma^2(n)} \exp\left(-\frac{\mid \hat{d}(n) - d \mid^2}{\sigma^2(n)}\right). \tag{25}$$

Recall that in Sec. II-A we used $\mathcal{X}$ to denote the set of all possible data symbols. Let $\mathcal{X}(i,b) = \left\{\mu\left([c^1, c^2, \cdots, c^{n_0}]\right) \mid c^i = b\right\}$, with $b \in \{1,0\}$ and $i \in \{1,2,\cdots,n_0\}$, denote the collection of all data symbols whose corresponding i-th coded bit is "fixed" as b. Then using (23)-(25) one obtains

$$L_e^M\left\{c^i(n)\right\} = \ln \frac{\sum_{d\in\mathcal{X}(i,1)} \left[\exp\left(-\frac{|\hat{d}(n)-d|^2}{\sigma^2(n)}\right) \prod_{\substack{j=1,\\ j\neq i}}^{n_0} P\{c^j(n) = c_d^j\}\right]}{\sum_{d\in\mathcal{X}(i,0)} \left[\exp\left(-\frac{|\hat{d}(n)-d|^2}{\sigma^2(n)}\right) \prod_{\substack{j=1,\\ j\neq i}}^{n_0} P\{c^j(n) = c_d^j\}\right]}. \tag{26}$$

The output extrinsic bit LLR's of the SISO demodulator are bit-wise deinterleaved as $\mathbf{L}_e^M\left\{\mathbf{c}(n')\right\} = \Pi^{-1}[\mathbf{L}_e^M\left\{\mathbf{c}(n)\right\}]$, which are then input to the SISO convolutional decoder. In SISO decoder, the MAP decoding algorithm for convolution codes (see [22, Sec. 6.2]) is applied to update the LLR's of the coded bits $\{\mathbf{c}(n)\}$ as well as the LLR's of the information bits $\{\mathbf{b}(n)\}$, based on the code constraints. The decoder computes the extrinsic LLR for coded bits

$$\mathbf{L}_e^D\left\{\mathbf{c}(n')\right\} = \left[L_e^D\left\{c^i(n')\right\},\ \ i = 1, 2\cdots, n_0\right], \tag{27}$$

where

$$L_e^D\left\{c^i(n')\right\} := \ln \frac{P\{c^i(n') = 1 \mid \mathbf{L}_e^M\{\mathbf{c}(l)\},\ 0 \le l < T_r\}}{P\{c^i(n') = 0 \mid \mathbf{L}_e^M\{\mathbf{c}(l)\},\ 0 \le l < T_r\}} - \underbrace{L_e^M\{c^i(n')\}}_{=:L\{c^i(n')\}}. \tag{28}$$

The output bit LLR's of SISO decoder are bit-wise interleaved as $\mathbf{L}_e^D\left\{\mathbf{c}(n)\right\} = \Pi[\mathbf{L}_e^D\left\{\mathbf{c}(n')\right\}]$. The SISO demodulator performs symbol-by-symbol MAP demodulation using LLR's $\mathbf{L}_e^D\left\{\mathbf{c}(n)\right\}$ for the coded bits generated by SISO decoder in the previous iteration as its *a priori* information: $L\{c^i(n)\} = L_e^D\{c^i(n)\}$. We set $L_e^D\left\{c^i(n)\right\} = 0$ for the initial step (first iteration). The LLR for use in LLR-to-symbol block is computed via $\mathbf{L}_a\left\{\mathbf{c}(n)\right\} = \mathbf{L}_e^M\left\{\mathbf{c}(n)\right\} + \mathbf{L}_e^D\left\{\mathbf{c}(n)\right\}$, which is the the *a posteriori* LLR. [It is claimed in [23] that, unlike the original turbo-principle where one takes $\mathbf{L}_a\left\{\mathbf{c}(n)\right\} = \mathbf{L}_e^M\left\{\mathbf{c}(n)\right\}$, usage of the full SISO-decoder's soft information embodied in the *a posteriori* LLR $\mathbf{L}_a\left\{\mathbf{c}(n)\right\}$ enhances performance compared to using only $\mathbf{L}_e^D\left\{\mathbf{c}(n)\right\}$; [13] also uses this set-up. This has also been our experience in the simulations presented in this paper; therefore, we have followed this approach.] The bit probabilities (converted from the corresponding LLR $\mathbf{L}_a\left\{\mathbf{c}(n)\right\}$) at time n are used (following [18]) to compute the mean $\bar{d}(n)$ and variance $\gamma^d(n)$ for data symbols $d(n)$ as

$$\bar{d}(n) = E[d(n)] = \sum_{d\in\mathcal{X}} d\, P\left\{d(n) = d\right\} = \sum_{d\in\mathcal{X}} d \prod_{j=1}^{n_0} P_a\left\{c^j(n) = c_d^j\right\} \tag{29}$$

and

$$\gamma^d(n) = \mathrm{var}[d(n)] = \sum_{d\in\mathcal{X}} |d(n) - \bar{d}(n)|^2 P\left\{d(n) = d\right\} = 1 - \bar{d}^2(n) \tag{30}$$

where

$$P_a\left\{c^j(n) = 1\right\} = \frac{1}{1 + \exp\left(-L_a\{c^j(n)\}\right)},$$
$$P_a\left\{c^j(n) = 0\right\} = \frac{1}{1 + \exp\left(L_a\{c^j(n)\}\right)}. \tag{31}$$

Then $\bar{d}(n)$ and $\gamma^d(n)$ are fed back to the equalizer as *a priori* information, along with the training symbols.

IV. Adaptive Soft-Input Soft-Output Nonlinear Kalman Equalizer

Using a symbol-wise AR-model for channel variations, an adaptive SISO equalizer using fixed-lag EKF was presented in [13] for joint channel estimation and equalization where their correlation was (implicitly) considered. In this section, we present a CE-BEM model-based SISO nonlinear Kalman equalizer for turbo equalization.

A. State-Space Model using CE-BEM and a Priori Information

We will perform equalization with a delay $\delta > 0$. Define a parameter

$$\bar{\delta} := \max\left\{\delta + 1, L + 1\right\} \tag{32}$$

and the data vector

$$\mathbf{z}\left(n\right) := \left[s\left(n\right) \quad s\left(n-1\right) \quad \cdots \quad s\left(n - \bar{\delta} + 1\right)\right]^T. \tag{33}$$

Consider (18). In order to apply (extended) Kalman filtering to joint channel estimation and equalization, we stack $\mathbf{h}\left(n\right)$ and data vector $\mathbf{z}\left(n\right)$ together into a $J \times 1$ state vector $\mathbf{x}\left(n\right)$ at time n as

$$\mathbf{x}\left(n\right) := \left[\mathbf{z}^T\left(n\right) \quad \mathbf{h}^T\left(n\right)\right]^T, \quad J := \bar{\delta} + Q\left(L+1\right). \tag{34}$$

As in [13] (and others), we consider the symbol sequence $\{s(n)\}$ as a stochastic process so as to utilize the soft decisions on the data symbols generated in the iterative decoding process as its *a priori* information. We can express $s(n)$ as $s(n) = \bar{s}(n) + \tilde{s}(n)$ where $\bar{s}(n) = E[s(n)]$ and $\tilde{s}(n)$ is approximated as a zero-mean uncorrelated sequence such that $E[\tilde{s}(n)\tilde{s}^*(n+j)] = \gamma(n)\delta(j)$, assuming an ideal interleaver. Note that $\bar{s}(n)$ and $\gamma(n)$ are provided via the *a priori* information. We have $\bar{s}(n) = \bar{d}(n)$ and $\gamma(n) = \gamma^d(n)$ for a data symbol $d(n)$ (where $\bar{d}(n)$ and $\gamma^d(n)$ are specified in (29) and (30), respectively),

while $\bar{s}(n) = t(n)$ and $\gamma(n) = 0$ for a training symbol $t(n)$. Using $\mathbf{x}(n)$, the state equation turns out to be

$$\mathbf{x}(n) = \mathcal{T}\mathbf{x}(n-1) + \mathbf{e}_0\bar{s}(n) + \mathbf{u}(n), \tag{35}$$

where

$$\mathcal{T} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0}_{\bar{\delta}\times Q(L+1)} \\ \mathbf{0}_{Q(L+1)\times\bar{\delta}} & \mathbf{F} \end{bmatrix}_{J\times J}, \quad \mathbf{F} = \alpha\mathbf{I}_{Q(L+1)}, \tag{36}$$

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{0}_{1\times(\bar{\delta}-1)} & 0_{1\times 1} \\ \mathbf{I}_{(\bar{\delta}-1)} & \mathbf{0}_{(\bar{\delta}-1)\times 1} \end{bmatrix}_{\bar{\delta}\times\bar{\delta}}, \quad \mathbf{e}_0 = \begin{bmatrix} 1 & \mathbf{0}_{1\times(J-1)} \end{bmatrix}^T, \tag{37}$$

the vector

$$\mathbf{u}(n) := \begin{bmatrix} \mathbf{e}_{\bar{\delta}}^T\tilde{s}(n) & \mathbf{w}^T(n) \end{bmatrix}^T \tag{38}$$

is zero-mean uncorrelated process noise where $\mathbf{e}_{\bar{\delta}} = [1\ 0_{1\times(\bar{\delta}-1)}]^T$, $\mathbf{w}(n)$ is given in (9) and

$$\mathbf{Q}(n) := E[\mathbf{u}(n)\mathbf{u}^H(n)] = \tilde{\mathbf{Q}} + \gamma(n)\mathbf{e}_0\mathbf{e}_0^T,$$
$$\tilde{\mathbf{Q}} := \begin{bmatrix} \mathbf{0}_{\bar{\delta}\times\bar{\delta}} & \mathbf{0}_{\bar{\delta}\times Q(L+1)} \\ \mathbf{0}_{Q(L+1)\times\bar{\delta}} & \sigma_w^2\mathbf{I}_{Q(L+1)} \end{bmatrix}_{J\times J}. \tag{39}$$

The channel output $y(n)$ in (1) can be rewritten as

$$y(n) = f[\mathbf{x}(n)] + v(n), \tag{40}$$

where ($\mathbf{b}_{ex}(n)$ is as defined in (11)) $f[\mathbf{x}(n)]$ is as defined in (41). With (35) and (40) as the state and measurement equations, respectively, nonlinear Kalman filtering is applied to track $\mathbf{x}(n)$ for joint channel estimation and equalization.

B. Fixed-Lag Soft Input Extended Kalman Filtering

The EKF is applied to (35) and (40) to track the BEM coefficients and to decode data symbols jointly. The EKF is initialized with

$$\hat{\mathbf{x}}(-1 \mid -1) = \mathbf{0} \ \text{ and } \mathbf{P}(-1 \mid -1) = \tilde{\mathbf{Q}} \tag{42}$$

where $\hat{\mathbf{x}}(p \mid m)$ denotes the estimate of $\mathbf{x}(p)$ given the observations $\{\mathbf{y}(0), \mathbf{y}(1), \cdots, \mathbf{y}(m)\}$, and $\mathbf{P}(p \mid m)$ denotes the error covariance matrix of $\hat{\mathbf{x}}(p \mid m)$, defined as

$$\mathbf{P}(p \mid m) := E\{[\hat{\mathbf{x}}(p \mid m) - \mathbf{x}(p)][\hat{\mathbf{x}}(p \mid m) - \mathbf{x}(p)]^H\}. \tag{43}$$

Extended Kalman recursive filtering (for $n = 0, 1, 2, \cdots$) is applied as in [13] but with a different state and measurement equations, to generate $\hat{\mathbf{x}}(n \mid n)$ and $\mathbf{P}(n \mid n)$. The following steps are executed:

1) Time update:

$$\hat{\mathbf{x}}(n \mid n-1) = \mathcal{T}\hat{\mathbf{x}}(n-1 \mid n-1) + \mathbf{e}_0\bar{s}(n), \tag{44}$$
$$\mathbf{P}(n \mid n-1) = \mathcal{T}\mathbf{P}(n-1 \mid n-1)\mathcal{T}^T + \tilde{\mathbf{Q}} + \gamma(n)\mathbf{e}_0\mathbf{e}_0^T. \tag{45}$$

2) Kalman gain:

$$\mathbf{j}(n) = \left.\frac{\partial f[\mathbf{x}]}{\partial\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(n|n-1)} \ \ldots \text{ Jacobian matrix}$$
$$= \hat{\mathbf{x}}^T(n \mid n-1)\left(\mathbf{D} + \mathbf{D}^T\right), \tag{46}$$
$$\mathbf{k}(n) = \mathbf{P}(n \mid n-1)\mathbf{j}^H(n)/\left[\sigma_v^2 + \mathbf{j}(n)\mathbf{P}(n \mid n-1)\mathbf{j}^H(n)\right]. \tag{47}$$

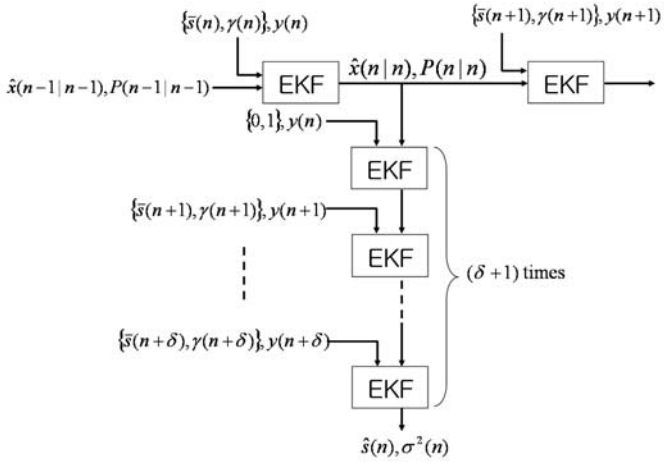


Fig. 3. Structure of adaptive SISO equalizer proposed in [13]. EKF: Extended Kalman Filter

3) Measurement update:

$$\hat{\mathbf{x}}(n \mid n) = \hat{\mathbf{x}}(n \mid n-1) + \mathbf{k}(n)\left(y(n) - f[\hat{\mathbf{x}}(n \mid n-1)]\right), \tag{48}$$
$$\mathbf{P}(n \mid n) = [\mathbf{I}_J - \mathbf{k}(n)\mathbf{j}(n)]\mathbf{P}(n \mid n-1). \tag{49}$$

The *a priori* information $\{\bar{s}(n), \gamma(n)\}$ is the soft input at time n acquired via (29) and (30), while δ-th element of the estimate $\hat{\mathbf{x}}(n+\delta \mid n+\delta)$ is the delayed *a posteriori* estimate of data symbol.

C. Structure of Adaptive Soft-Input Soft-Output Equalizer

The fixed-lag EKF takes soft inputs and generates a delayed *a posteriori* estimate for $s(n)$. In order to generate extrinsic estimate independent of the *a priori* information $\{\bar{s}(n), \gamma(n)\}$, a "comb" structure in conjunction with the EKF in Fig. 3 is used for SISO equalization, just as in [13]. At each time n, the vertical branch composed of $(\delta+1)$ EKF's produce the extrinsic estimate $\hat{s}(n)$, while the horizontal branch keeps updating the *a posteriori* estimate $\hat{\mathbf{x}}(n \mid n)$ and its error covariance $\mathbf{P}(n \mid n)$. The first vertical EKF has an input $\{0, 1\}$ in place of $\{\bar{s}(n), \gamma(n)\}$ to exclude the effect of the *a priori* information. Let $\hat{\mathbf{x}}_e(n+i \mid n+i)$ and $\mathbf{P}_e(n+i \mid n+i)$ denote the state estimate and its error covariance matrix, respectively, generated by the vertical filtering branch. Then the extrinsic estimate $\hat{s}(n)$ of $s(n)$ and its error variance $\sigma^2(n)$ are given by

$$\hat{s}(n) = \delta\text{th component of vector } \hat{\mathbf{x}}_e(n+\delta \mid n+\delta) \tag{50}$$
$$\sigma^2(n) = (\delta,\delta)\text{th component of matrix } \mathbf{P}_e(n+\delta \mid n+\delta). \tag{51}$$

Note that the extrinsic outputs $\hat{s}(n)$ and $\sigma^2(n)$ are computed for data symbol $d(n)$, not for training symbol $t(n)$, and then used in the later parts of the turbo-equalization receiver (see Fig. 2). Further details regarding generation of extrinsic estimates can be found in [13].

D. Computationally Complexity

The computational complexity of the approach of [13] is $\mathcal{O}((\delta+2)[\bar{\delta} + P(L+1)]^2)$ where δ is the equalization delay,

$$f[\mathbf{x}(n)] := \mathbf{x}^T(n) \underbrace{\left[\mathbf{I}_{(L+1)} \quad \mathbf{0}_{(L+1)\times(J-L-1)}\right]^T \left[\mathbf{I}_{(L+1)} \otimes \mathbf{b}_{ex}(n)\right]^H \left[\mathbf{0}_{[Q(L+1)]\times\bar{\delta}} \quad \mathbf{I}_{Q(L+1)}\right]}_{=:\mathbf{D}} \mathbf{x}(n). \tag{41}$$

$\bar{\delta}$ is given by (32) and an AR(P) channel model is used [13, Sec. IV-C]. Note that it is independent of the constellation size M. As we follow [13] with the difference that we use CE-BEM instead of AR modeling of the channel, the computational complexity of our proposed approach readily follows as $\mathcal{O}((\delta+2)[\bar{\delta}+Q(L+1)]^2) = \mathcal{O}((\delta+2)J^2)$ where Q is the number of basis functions in the CE-BEM. Therefore, the proposed approach and the approach of [13] have comparable computational complexity if one takes $P = Q$. As in [13], the computational complexity of our proposed approach is independent of the constellation size M. In the simulations presented in Sec. V, we have $\delta = 5$, $L = 2$ and $\bar{\delta} = 6$. For BEMs we take $Q = 5$ or $Q = 9$, therefore, corresponding values of the AR model order P in the approach of [13] were picked as 5 or 9 to attain comparable computational requirements for a fair performance comparison.

V. Simulation Examples

A random time- and frequency-selective Rayleigh fading channel is considered. We assume $h(n;l)$ is zero-mean, complex Gaussian WSS with variance σ_h^2. We take $L = 2$ (3 taps) in (1) (as in [13]), and $\sigma_h^2 = 1/(L+1)$ (i.e. uniform power delay profile). For different l's, $h(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [25] (with a correction in the appendix of [6]). We consider a communication system with carrier frequency of 2GHz, data rate of 40kBd (kilo-Bauds), therefore $T_s = 25\,\mu$s, and a varying Doppler spread f_d in the range of 40 to 400Hz, or the normalized Doppler spread f_dT_s from 0.001 to 0.01. The additive noise is zero-mean complex white Gaussian. The (receiver) SNR refers to the average energy per symbol over one-sided noise spectral density.

In the simulations, we use a 4-state convolutional code of rate $R_c = 1/2$ with octal generators $(5,7)$. The information block size is set to 3000 bits (T_i=3000) leading to a coded block size of 6000 bits, and the interleaver size is equal to the coded block size. In the modulator, the QPSK constellation with Gray mapping is used, which gives $M = 4$ and a block size of 3000 symbols. After modulation, training symbol sequences of length l_p are inserted in front of every l_s data symbols, leading to a sequence of length $T_r = 3750$ when $l_p = 5$ and $l_s = 20$ (20% training overhead).

For the CE-BEM, we consider BEM period $T_p = 200$ and 400 respectively, so that $Q = 5$ and 9, respectively, by (6). For the channel BEM coefficients, we take the AR-coefficient in (18) as $\alpha = 0.996$ for $T_p = 200$ and $\alpha = 0.998$ for $T_p = 400$. We compare our proposed BEM-based turbo equalization schemes (denoted by "TE-BEM(200)" for $T_p = 200$ and "TE-BEM(400)" for $T_p = 400$) with the AR(P) model-based scheme in [13] (denoted by "TE-AR5" for AR(5) model and "TE-AR9" for AR(9) model). The AR(P) model is as described in Sec. II-B1 and is fitted using [31] to Jakes' spectrum with f_dT_s=0.01 (the maximum anticipated normalized Doppler spread). We evaluate the performances of various schemes by considering their bit error rates (BER). The BER's are evaluated by employing the equalization delay $\delta = 5$ and using the decoded information symbol sequences at the turbo-equalization receiver output. All the simulation results are based on 1000 runs.

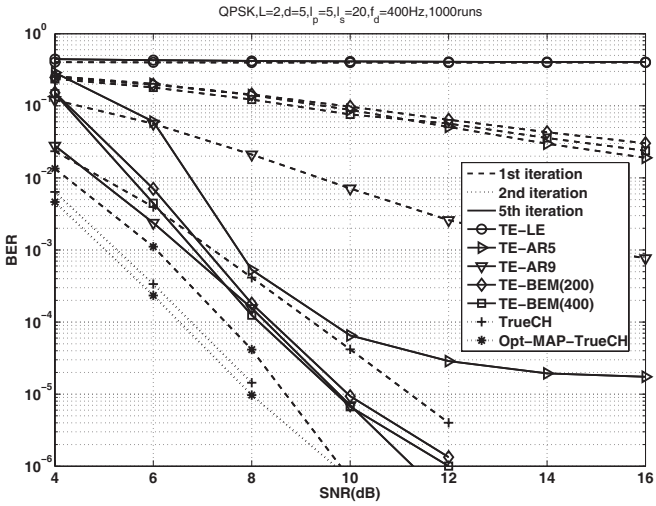


Fig. 4. BER vs SNR under $f_dT_s = 0.01$, $l_p = 5$, $l_s = 20$ (20% training overhead), for 1st and 5th iterations. Legend TE-LE refers to the turbo equalizer of [29] based on linear equalization. TE-BEM(T_p) is the proposed turbo equalizer using CE-BEM with BEM period T_p: TE-BEM(200) is based on $T_p = 200$ and $Q = 5$, and TE-BEM(400) is based on $T_p = 400$ and $Q = 9$. TE-ARP refers to the turbo equalizer of [13] using symbol-wise AR(P) channel model: TE-AR5 and TE-AR9 are based on AR(5) and AR(9) models, respectively. Legend TrueCH refers to the results of the turbo equalizer based on the fixed-lag Kalman filter with knowledge of the true channel. Legend Opt-MAP-TrueCH refers to the results of the turbo equalizer based on the optimum BCJR method with knowledge of the true channel.

In Fig. 4, the performances of the two schemes, under normalized Doppler spread $f_dT_s = 0.01$, are compared for different SNR's. In Fig. 5, the two schemes are compared over varying Doppler spreads f_d's, under SNR = 10dB; other settings of the simulation as for Fig. 4, including the fact that $Q = 5$ (for T_p=200) or $Q = 9$ (for T_p=400), regardless of the actual f_d. It is clear from these two figures that since the channel variations are well captured by the BEM coefficients, our proposed TE-BEM approach yields good performance even for "low" SNR's and over a wide range of Doppler spreads. Note that TE-BEM with larger block parameter $T_p = 400$ has a (slightly) better performance than with the smaller parameter $T_p = 200$; see the last paragraph in Sec. II-B2 for a possible explanation. The BER for TE-BEM varies only "slightly" with increasing normalized Doppler spread implying that its performance is not sensitive to the actual Doppler spread. Therefore, we do not have to know the exact Doppler spread of the channel – an upper bound on it is sufficient in practice. The performance of TE-AR5 is significantly worse than that of TE-BEM(200) (the two approaches have comparable computational complexity) in

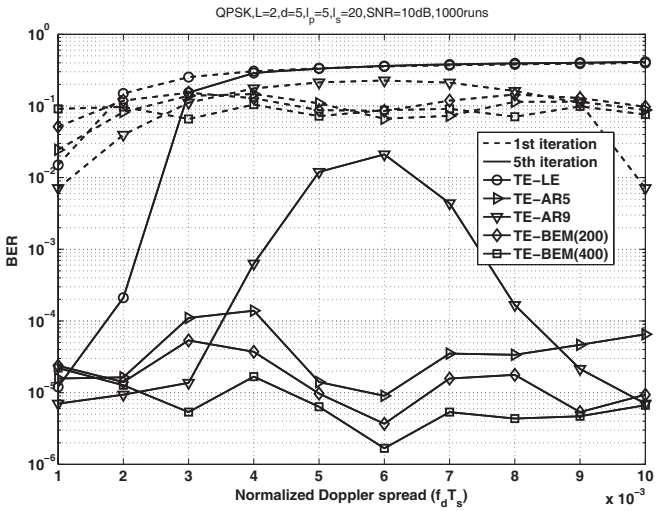


Fig. 5. BER vs f_dT_s under SNR = 10dB, $l_p = 5$, $l_s = 20$ (20% training overhead), for 1st and 5th iterations. TE-LE refers to the turbo equalizer of [29] based on linear equalization. TE-BEM(T_p) is the proposed turbo equalizer using CE-BEM with BEM period T_p: TE-BEM(200) is based on $T_p = 200$ and $Q = 5$, and TE-BEM(400) is based on $T_p = 400$ and $Q = 9$. TE-ARP refers to the turbo equalizer of [13] using symbol-wise AR(P) channel model: TE-AR5 and TE-AR9 are based on AR(5) and AR(9) models, respectively.

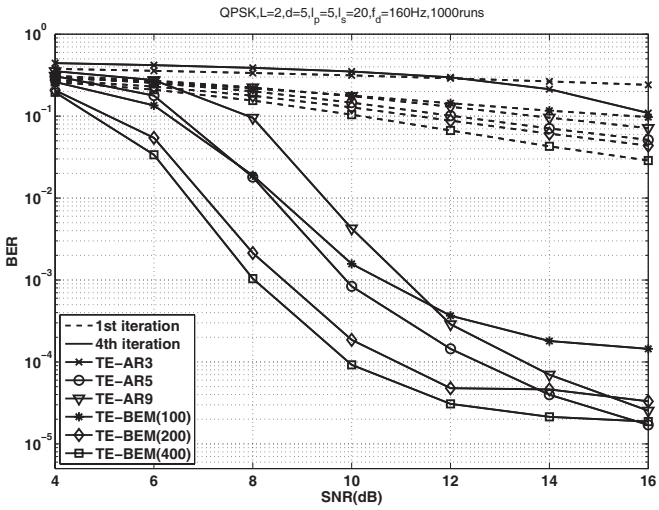


Fig. 6. BER vs SNR under $f_dT_s = 0.004$, $l_p = 5$, $l_s = 20$ (20% training overhead), for 1st and 4th iterations. TE-BEM(T_p) is the proposed turbo equalizer using CE-BEM with BEM period T_p: TE-BEM(100) is based on $T_p = 100$, $Q = 3$; TE-BEM(200) is based on $T_p = 200$, $Q = 5$; TE-BEM(400) is based on $T_p = 400$, $Q = 9$. TE-ARP refers to the turbo equalizer of [13] using symbol-wise AR(P) channel model: TE-AR3, TE-AR5 and TE-AR9 schemes are based on AR(3), AR(5) and AR(9) models, respectively.

Fig. 4 with increasing SNR for a fixed $f_dT_s = 0.01$, and is slightly worse in Fig. 5 for a fixed SNR of 10dB and varying Doppler spreads. On the other hand, while the performance of TE-AR9 is slightly better than that of TE-BEM(400) (the two approaches have comparable computational complexity) in Fig. 4 with increasing SNR for a fixed $f_dT_s = 0.01$, it is significantly worse in Fig. 5 for a fixed SNR of 10dB and varying Doppler spreads. While increasing the BEM period T_p improves performance, increasing the AR model order does not necessarily do so: we get inconsistent performance. A possible reason is that, as noted in [31], AR model fitting to a given correlation function can be numerically ill-conditioned for "large" model orders; it turned out to be so for AR(9) model and we followed the recommendations of [31] in choosing the regularization parameter for the matrix inversion involved. Such inconsistent behavior is also seen in Fig. 6 where we compare performance of various schemes (including TE-BEM(100) with $T_p = 100$ and $Q = 3$, and AR3 with order $P = 3$) for different SNR's under normalized Doppler spread $f_dT_s = 0.004$. It is seen that increasing the BEM period T_p improves performance but increasing the AR model order does not necessarily do so. Moreover, for the same computational complexity, BEM models outperform AR models.

In Figs. 4 and 5 the scheme TE-LE refers to the approach of [29] that uses the linear MMSE equalizer (e.g. [11]) coupled with modified RLS channel estimation. It is seen that this approach only works for normalized Doppler spread values of $\leq$0.002. In Fig. 4 we also present the performance of the turbo equalizer based on the fixed-lag Kalman filter with knowledge of the true channel (curves with plus sign marker and labeled "TrueCH") in order to illustrate the effectiveness of the proposed channel estimation approach; as there was little improvement beyond the second iteration, we only show the second iterative result with dotted curve labeled "TrueCH". It is seen that there is a slightly more than 2dB SNR penalty due to channel estimation. As has been noted in the literature, the Kalman filter based equalization is a sub-optimum equalizer compared to the trellis-based MAP (BCJR) equalizer [30]. In Fig. 4 we present the performance of the turbo equalizer based on the optimum BCJR method with knowledge of the true channel (curves with asterisk marker and labeled "Opt-MAP-TrueCH") in order to illustrate loss in performance due to suboptimality of the Kalman equalizer; as there was little improvement beyond the second iteration, we only show the second iterative result with dotted curve labeled "Opt-MAP-TrueCH". It is seen that while there is a large difference in performance initially (see 1st iteration results for "TrueCH" and "Opt-MAP-TrueCH" where both are dashed curves with plus sign and asterisk markers, respectively), just one turbo iteration yields very close performance (see the two dotted curves). That is, at least for this example, performance loss in using Kalman equalizer instead of the BCJR equalizer is quite negligible.

In Fig. 7, a smaller information block size ($T_i = 1000$) in the BICM transmitter is considered leading to a coded block size of 2000 bits and an interleaver length of 2000 bits also. Thus, we have a smaller interleaver size compared to 6000 bits in our earlier setting, designed to reduce the overall delay at turbo equalization receiver output. We compare the performance of TE-BEM with $T_p = 200$ and $Q = 5$ under different SNR's and normalized Doppler spread $f_dT_s = 0.01$ for two different interleaver lengths (equivalently different information block sizes). It is seen that a smaller interleaver length results in a "small" deterioration in BER (when five iterations are considered).

In Fig. 8 we show the BER performance of schemes TE-BEM(200) and TE-BEM(400) for different values of α. It is seen that while the performance is not sensitive to the choice

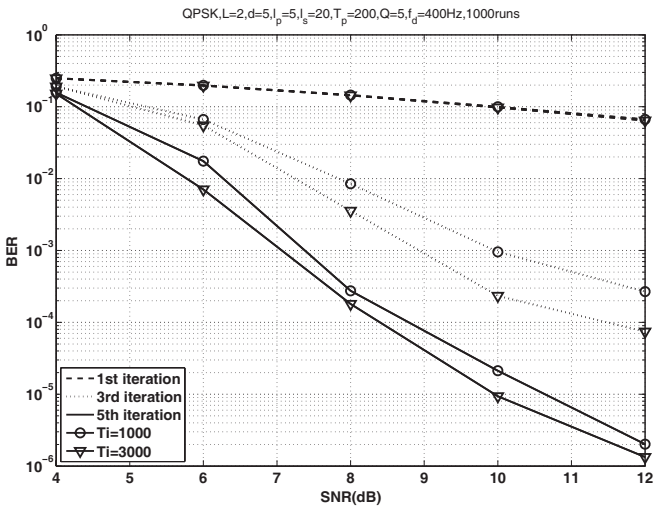
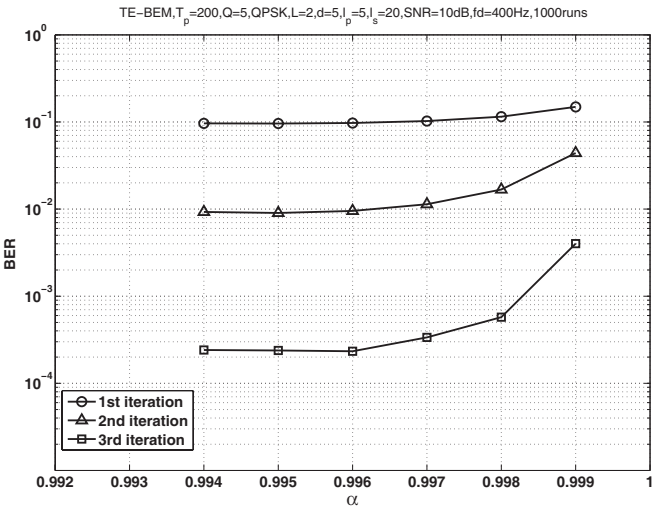
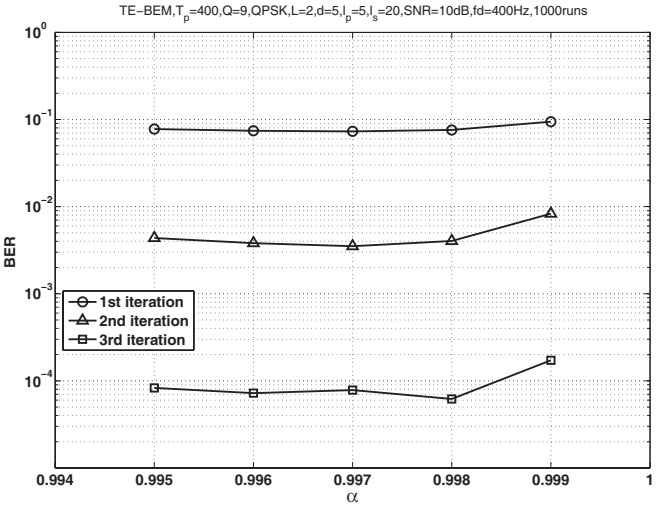


Fig. 7. BER vs SNR under different interleaver lengths, for the proposed turbo equalizer using CE-BEM with $T_p = 200$ and $Q = 5$ (TE-BEM(200) scheme), under $f_d T_s = 0.01$, $l_p = 5$, $l_s = 20$ (20% training overhead), for 1st, 3rd and 5th iterations. T_i = information block size in bits, interleaver length = $2T_i$.



(a) Proposed turbo equalizer using CE-BEM with $T_p = 200$ and $Q = 5$ (TE-BEM(200) scheme)



(b) Proposed turbo equalizer using CE-BEM with $T_p = 400$ and $Q = 9$ (TE-BEM(400) scheme)

Fig. 8. BER vs α for the proposed turbo equalizers TE-BEM(200) and TE-BEM(400) for the first three turbo iterations under $f_d T_s = 0.01$, SNR=10dB, $l_p = 5$, $l_s = 20$ (20% training overhead).

α over a relatively wide range of values, it does deteriorate as α approaches one. Note that $\alpha = 1$ in (18) implies time-invariance and $\alpha < 1$ permits tracking by discounting older values of the channel BEM coefficients – smaller the value of α higher this discounting effect but discrepancy with the value of α obtained from (19) also increases.

VI. EXIT Chart Analysis

The extrinsic information transfer (EXIT) chart is a useful semi-analytic tool [26]–[28] to analyze the exchange of mutual information between the equalizer and the decoder and to describe the convergence of the iterative receiver algorithm. The EXIT chart makes it possible to predict the system trajectory from extrinsic mutual information transfer functions without performing simulations on the complete iterative receiver. The (extrinsic) mutual information $I(L;c)$ between the equally likely $c \in \{+1,-1\}$ and the symmetric LLR L simplifies to [26], [27]

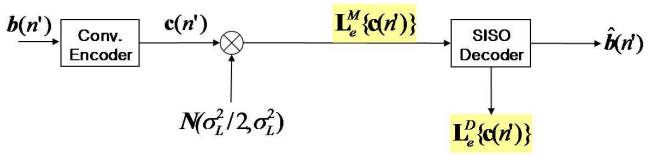
$$I(L;c) = 1 - E\left[\log_2(1+e^{-L}) \mid c=+1\right]. \qquad (52)$$

Under ergodicity, for a large sample of size T_r, we have [27]

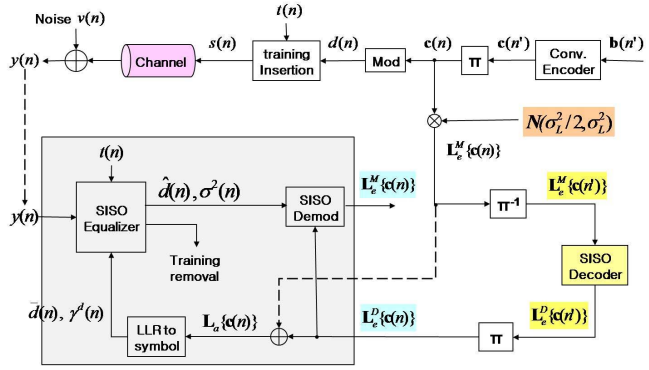
$$I(L;c) \approx 1 - T_r^{-1} \sum_{t=1}^{T_r} \log_2(1+e^{-c(t)L\{c(t)\}}). \qquad (53)$$

We observe the mutual information $I_e^M = I(L_e^M\{c(n)\}; c(n))$ at the equalizer output and $I_e^D = I(L_e^D\{c(n')\}; c(n'))$ at the decoder output. The EXIT chart combines the equalizer transfer function and the decoder transfer function. Since the output LLR's from the equalizer are input to the decoder and vice versa, both transfer functions are drawn in the same plot with the axes being flipped for the decoder transfer function. The system trajectory of the turbo equalization receiver forms a "zigzag-path" between the two transfer functions where each equalization (or decoding) task is represented as a vertical (or horizontal) arrow.

The simulation setup to generate the extrinsic information transfer function is shown in Fig. 9. Following [26] (and others), $L_e^M\{c^i(n')\}$ (input to the SISO decoder) is modeled as independent and identically distributed (i.i.d.) Gaussian with mean $c^i(n')\sigma_L^2/2$ and variance σ_L^2; then mutual information I_e^M and I_e^D at the input and output, respectively, of the decoder are functions of a single parameter σ_L. For a range of values of σ_L and randomly generated $L_e^M\{c^i(n')\}$, we can estimate I_e^M and I_e^D (the same for all channel models) via simulations using (53). The interleaved random extrinsic LLR's $L_e^D\{c^i(n)\}$ are input to the "LLR to symbol" block in Fig. VI together with the corresponding *a priori* LLR $L_e^M\{c^i(n)\}$, the input LLR's of the decoder, in order to obtain the *a posteriori* LLR's $L_a\{c^i(n)\}$. Then we can estimate input-output mutual information I_e^D and I_e^M of the equalizer (dependent upon the channel model) using the input LLR's $L_e^D\{c^i(n)\}$ (not $L_a\{c^i(n)\}$) and output LLR's $L_e^M\{c^i(n)\}$, respectively, of



(a) Decoder



(b) Equalizer

Fig. 9. Simulation setup for generating extrinsic information transfer functions. $N(m, \sigma^2)$ denotes a Gaussian distribution with mean m and variance σ^2. In (b) the equalizer block is shown "blocked" and shaded in the left-bottom side while the right-side generates the entitiy needed to generate *a posteriori* LLR $\mathbf{L}_a\{\mathbf{c}(n)\}$ (see also Fig. 2).

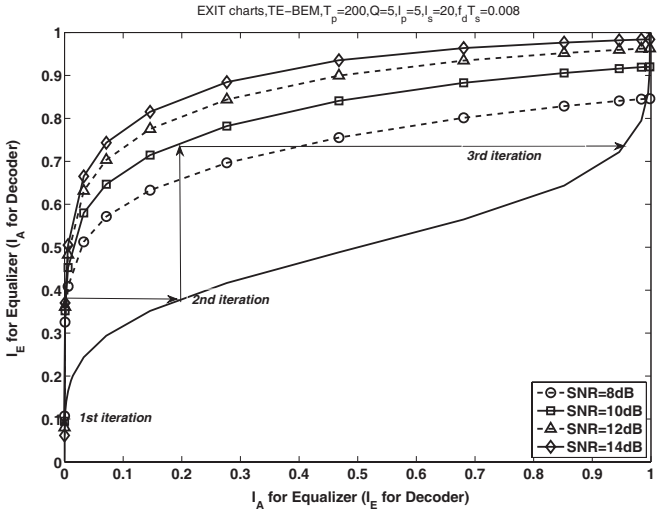
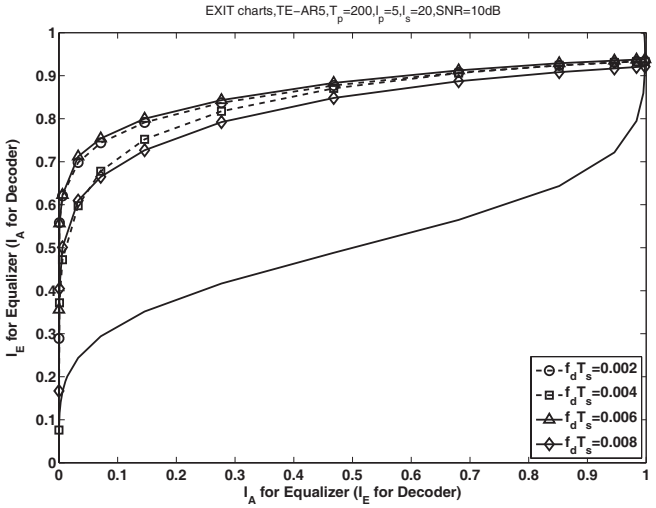


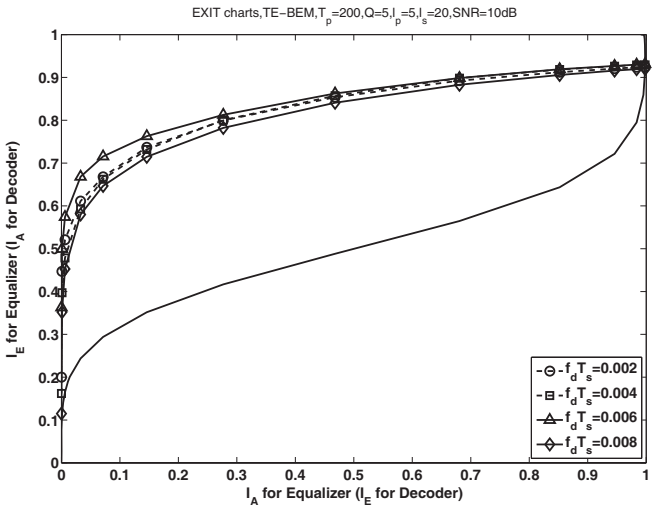
Fig. 10. EXIT charts for the proposed turbo equalizer using CE-BEM with $T_p = 200$ and $Q = 5$ (TE-BEM(200) scheme) under $f_dT_s = 0.008$ for different SNR's; $l_p = 5$, $l_s = 20$ (20% training overhead). The "bottom" solid black curve is the decoder transfer function; the other curves are the equalizer transfer functions for different SNR's.

a given SISO equalizer. For a given equalizer we plot curves (transfer function) with input I_e^D along horizontal axis and output I_e^M along the vertical axis; the axes are "flipped" for the decoder. The iteration process between equalizer and decoder can be visualized by using a trajectory trace where each vertical trace represents equalization task and each horizontal trace represents decoding task and the trajectory starts at the (0,0) point (see Fig. 10 for instance).

Using the set-up and parameters of Fig. 4 but with the information block size set to 30000 bits (the coded block size $T_r = 60000$), the normally distributed LLR's were generated



(a) Turbo equalizer of [13] using symbol-wise AR(5) channel model



(b) Proposed turbo equalizer using CE-BEM with $T_p = 200$ and $Q = 5$

Fig. 11. EXIT charts under $SNR = 10dB$ for different f_dT_s's; $l_p = 5$, $l_s = 20$ (20% training overhead). The "bottom" solid black curve is the decoder transfer function; the other curves are the equalizer transfer functions for different channel models and Doppler spreads. Part (a) shows results for the approach of [13] (TE-AR5 scheme) based on symbol-wise AR(5) model. Part (b) shows results for the proposed turbo equalizer using CE-BEM with $T_p = 200$ and $Q = 5$ (TE-BEM(200) scheme).

with values of $\sigma_L^2 \in \left[10^{-2}, 10^2\right]$, and then I_e^M and I_e^D were calculated. We analyze the EXIT charts of our CE-BEM based approach with $T_p = 200$ and $Q = 5$ (TE-BEM(200) scheme) and the symbol-wise AR-model-based approach in [13] using AR(5) model (TE-AR5 scheme). In Fig. 10, EXIT charts for TE-BEM(200) are shown under a fixed normalized Doppler spread $f_dT_s = 0.008$ for different SNR's. In Fig. 10 we show the trajectory trace for SNR=10 dB where the first iteration is not visible as it is "cramped" in the lower left corner. Note that, as SNR increases, the output mutual information of the SISO equalizer increases. In Fig. 11, EXIT charts for TE-BEM(200) and TE-AR5 schemes are depicted under SNR = 10dB for different normalized Doppler spreads. Table I compares the BER's obtained via full Monte Carlo

TABLE I
COMPARISON BETWEEN ACTUAL BER (VIA SIMULATIONS) AND PREDICTED BER (VIA EXIT CHARTS)

		Via Monte Carlo runs (predicted by EXIT charts): SNR=10dB	
f_d (f_dT_s)	iteration	TE-BEM(200)	TE-AR5
$80Hz$ (0.002)	1st	1.2×10^{-1} (1.4×10^{-1})	0.8×10^{-1} (1.2×10^{-1})
	2nd	2.3×10^{-2} (2.4×10^{-2})	8.5×10^{-3} (1.5×10^{-3})
	3rd	1.9×10^{-3} (1.5×10^{-5})	6.3×10^{-4} ($< 10^{-5}$)
$160Hz$ (0.004)	1st	1.3×10^{-1} (1.5×10^{-1})	1.5×10^{-1} (1.7×10^{-1})
	2nd	2.7×10^{-2} (5.5×10^{-2})	4.8×10^{-2} (9.0×10^{-2})
	3rd	2.1×10^{-3} (2.0×10^{-4})	7.0×10^{-3} (2.4×10^{-3})
$240Hz$ (0.006)	1st	8.5×10^{-2} (1.1×10^{-1})	6.6×10^{-2} (8.5×10^{-2})
	2nd	6.9×10^{-3} (2.2×10^{-3})	3.3×10^{-3} (4.0×10^{-4})
	3rd	1.9×10^{-4} ($< 10^{-5}$)	7.1×10^{-5} ($< 10^{-5}$)
$320Hz$ (0.008)	1st	1.4×10^{-1} (1.7×10^{-1})	1.1×10^{-1} (1.6×10^{-1})
	2nd	3.9×10^{-2} (7.0×10^{-2})	1.8×10^{-2} (4.5×10^{-2})
	3rd	2.9×10^{-3} (1.3×10^{-3})	7.2×10^{-4} (2.5×10^{-4})

simulations (as in Sec. V, Fig. 5, with $l_p = 5$, $l_s = 20$ and SNR=10 dB) and predicted by EXIT chart analysis (shown in parentheses). It is seen that while the two sets of BER's are "close," there are discrepancies. One reason for this is that while EXIT charts are based on the assumption of infinite interleaver length, simulation results are based on finite length interleaver. Furthermore, drawing of trajectory traces is subject to "manual" errors.

VII. CONCLUSIONS

We extended the single-user turbo equalization approach of [13] based on symbol-wise AR modeling of channels to channels based on CE-BEMs where the adaptive equalizer using nonlinear Kalman filters is coupled with an SISO decoder to iteratively perform equalization and decoding using soft information feedback. The proposed adaptive equalizer jointly optimizes the estimation of BEM channel coefficients and data symbol decoding in each iteration with the assistance of *a priori* information for the data symbols given by the SISO decoder. Unlike [13], an EXIT chart analysis of the proposed approach was also provided. Simulation examples demonstrated that our CE-BEM-based approach had significantly superior performance over the symbol-wise AR model-based turbo equalizer of [13] for comparable computational complexity.

REFERENCES

[1] H. S. Wang and P.-C. Chang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 353-357, May 1996.

[2] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd edition. Upper Saddle River, NJ: Prentice-Hall, 2002.

[3] X. Ma, G. B. Giannakis, and S. Ohno, "Optimal training for block transmissions over doubly selective channels," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1351-1366, May 2003.

[4] G. B. Giannakis and C. Tepedelenlioğlu, "Basis expansion models and diversity techniques for blind identification and equalization of time-varying channels," *Proc. IEEE*, vol. 86, no. 10, pp. 1969-1986, Oct. 1998.

[5] C. Komninakis, C. Fragouli, A. H. Sayed, and R. D. Wesel, "Multi-input multi-output fading channel tracking and equalization using Kalman estimation," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1065-1076, May 2002.

[6] T. Zemen and F. Mecklenbräuker, "Time-variant channel estimation using discrete prolate spheroidal sequences," *IEEE Trans. Signal Process.*, vol. 53, no. 9, pp. 3597-3607, Sep. 2005.

[7] S. He and J. K. Tugnait, "Doubly-selective channel estimation using exponential basis models and subblock tracking," in *Proc. IEEE GLOBECOM '07*, pp. 2847-2851, Washington, DC, Nov. 26-30, 2007.

[8] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *Proc. IEEE Intern. Conf. Commun.*, pp. 1064-1070, Geneva, Switzerland, May 1993.

[9] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux,"Iterative correction of intersymbol interference: turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507-511, Sep.-Oct. 1995.

[10] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 1046-1061, July 1999.

[11] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Signal Process.*, vol. 50, pp. 673-683, Mar. 2002.

[12] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: principles and new results," *IEEE Trans. Commun.*, vol. 50, pp. 754-767, May 2002.

[13] X. Li and T. F. Wong, "Turbo equalization with nonlinear Kalman filtering for time-varying frequency-selective fading channels," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 691-700, Feb. 2007.

[14] M. Nissilä and S. Pasupathy, "Adaptive Bayesian and EM-based detectors for frequency-selective fading channels," *IEEE Trans. Commun.*, vol. 51, nov. 8, pp. 1325-1336, June 2003.

[15] S. Song, A. Singer, and K. Sung, "Soft input channel estimation for turbo equalization," *IEEE Trans. Signal Process.*, vol. 52, pp. 2885-2894, Oct. 2004.

[16] R. Otnes and M. Tüchler, "Iterative channel estimation for turbo equalization for time-varying frequency-selective channels," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 1918-1923, Dec. 2004.

[17] G. Leus, "On the estimation of rapidly time-varying channels," in *Proc. European Signal Proc. Conf.*, pp. 2227-2230, Vienna, Austria, Sep. 2004.

[18] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," in *Proc. IEEE Intern. Conf. Commun.*, pp. 858-864, June 1999.

[19] S. M. Kay, *Modern Spectral Analysis: Theory and Application*. Englewoods Cliffs, NJ: Prentice Hall, 1988.

[20] P. Stoica and R. Moses, *Spectral Analysis of Signals*. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[21] Z. Liu, X. Ma, and G. B. Giannakis, "Space-time coding and Kalman filtering for time-selective fading channels," *IEEE Trans. Commun.*, vol. 50, pp. 183-186, Feb. 2002.

[22] X. Wang and H. V. Poor, *Wireless Communication Systems: Advanced Techniques for Signal Reception*. Upper Saddle River, NJ: Prentice Hall, 2004.

[23] F. Vogelbruch and S. Haar, "Low complexity turbo equalization based on soft feedback interference cancelation," *IEEE Commun. Letters*, vol. 9, no. 7, pp. 586-588, July 2005.

[24] R.R. Lopes and J.R. Barry, "The soft-feedback equalizer for turbo equalization of highly dispersive channels," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 783-788, May 2006.

[25] Y. R. Zheng and C. Xiao, "Simulation models with correct statistical properties for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 51, no. 6, pp. 920-928, June 2003.

[26] S. ten Brink, "Convergence behavior of iterative decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727-1737, Oct. 2001.

[27] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. Conference on Information Science and Systems*, CISS 2002, Mar. 2002.
[28] R. Otnes and M. Tüchler, "EXIT chart analysis applied to adaptive turbo equalization," in *Proc. IEEE Nordic Signal Proc. Symp.*, Hurtigruta, Tromsö-Trondheim, Norway, Oct. 2002.
[29] R. Otnes and M. Tüchler, "Iterative channel estimation for turbo equalization of time-varying frequency-selective channels," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 1918-1923, Nov. 2004.
[30] R. Koetter, A. C. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Process. Mag.*, vol. 21, pp. 67-80, 2004.
[31] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 1650-1662, July 2005.
[32] K. Huber and S. Haykin, "Improved Bayesian MIMO channel tracking for wireless communications: incorporating a dynamical model," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 2468-2476, Sep. 2006.

Hyosung Kim received the B.A. degree in Electrical Engineering from Soongsil University, Seoul, Korea, in 1996 and the M.S. degree in Computer Science from Korea National Defense University in Seoul, Korea, in 2004. He is currently working toward the Ph.D. degree in Electrical Engineering at Auburn University, Auburn, AL. Since 2007, he has been a Graduate Research Assistant at the Department of Electrical and Computer Engineering, Auburn University. His research interests include channel estimation and equalization, multiuser detection, wireless security, and statistical and adaptive signal processing and analysis.

Jitendra K. Tugnait (M'79–SM'93–F'94) was born in Jabalpur, India on December 3, 1950. He received the B.Sc.(Hons.) degree in electronics and electrical communication engineering from the Punjab Engineering College, Chandigarh, India in 1971, the M.S. and the E.E. degrees from Syracuse University, Syracuse, NY and the Ph.D. degree from the University of Illinois, Urbana-Champaign in 1973, 1974, and 1978, respectively, all in electrical engineering.

From 1978 to 1982 he was an Assistant Professor of Electrical and Computer Engineering at the University of Iowa, Iowa City, IA. He was with the Long Range Research Division of the Exxon Production Research Company, Houston, TX, from June 1982 to Sept. 1989. He joined the Department of Electrical & Computer Engineering, Auburn University, Auburn, AL, in September 1989 as a Professor. He currently holds the title of James B. Davis Professor. His current research interests are in statistical signal processing, wireless and wireline digital communications, multiple sensor multiple target tracking and stochastic systems analysis.

Dr. Tugnait is a past Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE SIGNAL PROCESSING LETTERS. He is currently an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.