RECURSIVE LEAST-SQUARES DECISION-DIRECTED TRACKING OF DOUBLY-SELECTIVE CHANNELS USING EXPONENTIAL BASIS MODELS

Hyosung Kim and Jitendra K. Tugnait

Department of Electrical & Computer Engineering, Auburn University, Auburn, AL 36849

ABSTRACT

We present a decision-directed tracking approach to doubly-selective channel estimation exploiting the complex exponential basis model (CE-BEM). The time-varying nature of the channel is well captured by the CE-BEM while the time-variations of the (unknown) BEM coefficients are likely much slower than those of the channel. We track the BEM coefficients via the exponentially-weighted recursive least-squares (RLS) algorithm, aided by symbol decisions from a decision-feedback equalizer (DFE). Simulation examples demonstrate its superior performance over an existing subblock-wise channel tracking scheme.

Index Terms— Doubly-selective channels, adaptive channel estimation, recursive least-squares, basis expansion models

1. INTRODUCTION

Recently basis expansion models (BEM) have been investigated to represent doubly-selective channels in wireless applications [1, 9, 13]. In contrast to the symbol-wise AR models that describe the channel variations on a symbol-by-symbol basis, a BEM depicts evolution of the channel over a period (block) of time. Using time-multiplexed (TM) training, in [2], a subblock-wise tracking approach based on a CE-BEM for the channel and an AR model for the BEM coefficients was proposed. This approach outperforms the symbol-wise AR model-based approach in fast-fading environments. In [3], decision-directed tracking of CE-BEM-based doubly-selective channels was proposed based on the subblock tracking approach of [2]. The approaches of [2, 3] assume that each BEM coefficient follows a first-order AR process, which is not necessarily true for a “real-world” channel, and this assumption possibly incurs modeling error in estimation. To circumvent this problem, an adaptive channel estimation scheme with no a priori model for the BEM coefficients was proposed in [4], where two finite-memory adaptive filtering algorithms, the exponentially-weighted and the sliding-window recursive least-squares (RLS) algorithm, are considered for subblock-wise channel tracking.

In this paper, we propose an RLS decision-directed approach to track the channel and to detect information symbols. It is based on the exponentially-weighted RLS (EW-RLS) algorithm of [4] and the decision-directed tracking proposed in [3]. Decision-directed channel tracking using a polynomial BEM has been investigated in [5], where the BEM coefficients are updated via the RLS algorithm with a sliding window. Decision-directed channel estimation using Kalman filtering and polynomial or CE-BEM for OFDM systems has been explored in [6, 8]. The contributions [5, 6, 8] consider block-by-block updating and/or a priori stochastic models for BEM coefficients, whereas our decision-directed scheme updates the BEM coefficients much more frequently and without using any “arbitrary” model for variations of the BEM coefficients.

Notation: Superscripts $s$, $T$, and $H$ denote the complex conjugation, transpose, and complex conjugate transpose respectively. $I_N$ is the $N \times N$ identity matrix, $\Theta_{M \times N}$ is the $M \times N$ null matrix and $\otimes$ denotes the Kronecker product. $\delta(\tau)$ is the Kronecker delta.

2. SYSTEM MODEL

Consider a doubly-selective, single-input multi-output (SIMO), FIR linear channel with $N$ outputs and discrete-time response \( \{h(n; l)\} \) ($N$-column vector channel response at time instance $n$ to a unit input at time instance $n - l$). With \( \{s(n)\} \) as the scalar information sequence, the symbol-rate noisy $N$-column channel output vector is given by ($n = 0, 1, \ldots$)

\[
y(n) = \sum_{l=0}^{L} h(n; l) s(n - l) + v(n)
\]

where $v(n)$ is zero-mean, white noise, uncorrelated with $s(n)$, with $E\{v(n + \tau) v^H(n)\} = \sigma_v^2 I_N \delta(\tau)$. In TM training schemes, $s(n)$ can be either a training or an information symbol.

In CE-BEM [1, 9, 10], over the $k'$-th block consisting of an observation window of $T_B$ symbols, the channel is represented as ($n = (k' - 1)T_B, (k' - 1)T_B + 1, \ldots, k'T_B - 1$ and $l = 0, 1, \ldots, L$)

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

where one chooses ($q = 1, 2, \ldots, Q$ and $K \geq 1$ is an integer)

\[
T := k'T_B, \quad Q \geq 2 \left[ f_d T_s \right] + 1,
\]

\[
\omega_q := \frac{2\pi}{T} \left[ q - (Q + 1)/2 \right], \quad L := \left[ \tau_d/T_s \right],
\]

\[
T_d \quad \text{and} \quad f_d \quad \text{are respectively the delay spread and the Doppler spread, and} \quad T_s \quad \text{is the symbol duration. The BEM coefficients} \quad h_q^{(l)} \quad \text{remain invariant during this block, but are allowed to change at the next block; the functions} \quad \{e^{j\omega_q n}\}
\]

This work was supported by NSF under Grant ECCS-0823987.
(q = 1, 2, · · · , Q) are common for every block. If K ≥ 2, the Doppler spectrum is over-sampled (therefore the representation (2) is called over-sampled CE-BEM) [10], compared with the critically sampled case K = 1 [1, 9].

3. DECISION-DIRECTED TRACKING

Consider two overlapping blocks (each consisting of T_B symbols) that differ by only m_s (1 ≤ m_s < T_B) symbols: the “past” block beginning at time n_0, and the “present” block beginning at time n_0 + m_s. Thanks to the significant overlapping of the two blocks, one can expect the BEM coefficients to vary only a little from the past block to the current overlapping one. Therefore, rather than estimate h_lq anew at every non-overlapping block as in [9], we propose to update the BEM coefficients every m_s ("step-size") symbols using TM training and detected symbols. [In the following s(n) denotes the symbol estimate and ˘s(n) denotes the detected symbol.]

3.1. State-Space Modeling using CE-BEM

Stack the BEM coefficients in (2) vectors

$$h^{(i)} := \begin{bmatrix} h_1 \dot{h} \cdots h_Q \end{bmatrix}^T$$

of size NQ and M := NQ (L + 1) respectively. The coefficient vectors in (5) and (6) of the p-th overlapping block will be denoted by h_lq(p) and h(p). Again, we emphasize that the p-th block and the (p + 1)-st block differ by just m_s symbols. For pm_s ≤ n < (p + 1) m_s, by (1), (2), (5) and (6), the received signal at time n can be written as

$$y(n) = S^T(n) [I_{L+1} \otimes \mathcal{E}(n)]^H h(p) + v(n)$$

where

$$\mathcal{E}(n) := \begin{bmatrix} e^{-j \omega_1 n} & e^{-j \omega_2 n} & \cdots & e^{-j \omega_Q n} \end{bmatrix}^T \otimes I_N,$$

$$S(n) := \begin{bmatrix} s(n) & s(n-1) & \cdots & s(n-L) \end{bmatrix}^T \otimes I_N.$$.

Further defining

$$C_i(p) := S^T(pm_s + i)[I_{L+1} \otimes \mathcal{E}(pm_s + i)]^H,$$

$$C(p) := \begin{bmatrix} C_1(p) & C_2(p) & \cdots & C_{m_s-1}(p) \end{bmatrix}^T,$$

we have

$$y_{ms}(p) = C(p) h(p) + v_{ms}(p)$$

where

$$y_{ms}(p) := \begin{bmatrix} y^T(pm_s) & y^T(pm_s + 1) & \cdots & y^T((p + 1)m_s - 1) \end{bmatrix}^T$$

and v_{ms}(p) is defined likewise. Using TM training or detected symbols in C(p), our objective is to devise an RLS scheme for estimating h(p).

3.2. Exponentially-Weighted RLS Tracking

Based on (9), we can apply exponentially-weighted regularized RLS (EW-RLS) algorithm [11, Chapter 12] to track an unknown h(p). Choose h to minimize the cost function

$$\lambda^{p+1} \beta \|h\|^2 + \sum_{i=0}^{p} \lambda^{p-i} \|y_{ms}(i) - C(i)h\|^2$$

(10)

where \( \beta > 0 \) is a regularization parameter and \( 0 < \lambda < 1 \) is the forgetting factor. Mimicking [11, Algorithm 12.3.1] (see also [4]), EW-RLS tracking comprises the following steps:

1) Initialization: \( \hat{h}(-1) = 0_{M 	imes 1} \) and \( P(-1) = \beta^{-1} I_M \)
2) RLS recursion: For \( p = 0, 1, \cdots \)

$$\begin{align*}
\mathbf{\Gamma}(p) & = \lambda I_{N_{m_s}} + C(p)P(p-1)C^H(p), \\
\mathbf{G}(p) & = P(p-1)C^H(p)\mathbf{\Gamma}^{-1}(p), \\
\hat{h}(p) & = \hat{h}(p-1) + \mathbf{G}(p)\left[y_{ms}(p) - C(p)\hat{h}(p-1)\right], \\
P(p) & = \lambda^{-1}(I_M - G(p)C(p))P(p-1),
\end{align*}$$

where \( \hat{h}(p) \) denotes the estimate of h(p) given the observations \( \{y_{ms}(0), y_{ms}(1), \cdots, y_{ms}(p)\} \).

After RLS recursion for every p, we can generate the channel by the estimated \( \hat{h}(p) \) via the CE-BEM (2).

3.3. Channel Prediction

We employ a DFE [12] with equalization delay \( d > 0 \) to equalize the estimated channel at the receiver. Its output symbol decisions are used as a pseudo-training. We need to “predict” the channel up to time \( n \) to obtain the detected symbol \( \hat{s}(n-d) \) at the DFE. We use the “current” BEM coefficient vector estimate in the CE-BEM (2) to predict the channel \( \hat{h}(n;l) \) for values of \( n \) as needed, i.e., the channel estimates in our receiver are given by

$$\hat{h}(n;l) = \mathcal{E}^H(n)\hat{h}^{(i)}(p),$$

(11)

for \( n = pm_s, pm_s + 1, \cdots, (p + 1)m_s + d - 1 \) where the definition of \( \hat{h}^{(i)}(p) \) is similar to (5) and \( \hat{h}^{(i)}(p) \) is based on observations up to time \( n = (p + 1)m_s - 1 \).

3.4. Minimum Mean-Square Error (MMSE)-DFE

Using the estimated channel, the symbol decisions are made by an FIR MMSE-DFE [12]. Given the lengths of the feedforward (FF) and feedback (FB) filters as \( l_f \) and \( l_b \), respectively, the estimate of the information symbol \( \tilde{s}(n-d) \) is obtained by combining the outputs of FF and FB filters and can be written as

$$\tilde{s}(n-d) = \sum_{m=0}^{l_f-1} f_m^{l_f}(n)y(n-m) - \sum_{k=1}^{l_b} b_k(n)\tilde{s}(n-d-k)$$

(12)

where \( N \times 1 f_m^{l_f}(n) \)'s and scalar \( b_k(n) \)'s are the taps of FF and FB time-varying filters at time \( n \), and \( \hat{s}(n-d-k) \) is the hard decision of \( \tilde{s}(n-d-k) \). The estimate \( \tilde{s}(n-d) \) is also fed into the quantizer to obtain the symbol decision \( \hat{s}(n-d) \).
Stack the inputs of the FF filter at time $n$ as

$$y_f(n) := [y^T(n) \ y^T(n-1) \ \cdots \ y^T(n-l_f+1)]^T$$

and also define $v_f(n)$ likewise. By (1), we have

$$y_f(n) = H(n) s_f(n) + v_f(n)$$

where $H(n) := \begin{bmatrix} h(n;0) & \cdots & h(n;L) \\ \vdots & \vdots & \vdots \\ h(n-l_f+1;0) & \cdots & h(n-l_f+1;L) \end{bmatrix}$

$$s_f(n) := [s(n) \ s(n-1) \ \cdots \ s(n-l_f-L+1)]^T.$$ 

Further define

$$s_b(n) := [\hat{s}(n-d) \ \hat{s}(n-d-1) \ \cdots \ \hat{s}(n-d-l_b)]^T.$$ 

Since $\{s(n)\}$ is i.i.d. with variance $\sigma^2_s$, from (13) we have

$$R_{ss}(n) := E\{s_b(n)s_b^H(n)\} = \sigma^2_s I_{l_b+1},$$

$$R_{sy}(n) := E\{s_b(n)y_f^H(n)\} = \sigma^2_s \Phi H^H(n),$$

$$R_{yy}(n) := E\{y_f(n)y_f^H(n)\} = \sigma^2_s H(n) H^H(n) + \sigma^2_s I_{N_lf}$$

where $\Phi := [0_{(l_b+1)\times d} I_{l_b+1} 0_{(l_b+1)\times (l_f+L-d-l_b-1)}].$

Let $f(n)$ and $b(n)$ denote the vectors of time-varying taps of FF and FB filters,

$$f(n) := [f_1^T(n) \ f_2^T(n) \ \cdots \ f_{l_f-1}^T(n)]^T,$$

$$b(n) := [1 \ b_1(n) \ b_2(n) \ \cdots \ b_{l_b}(n)]^T.$$ 

Assuming the decisions $\{\hat{s}(n)\}$ are correct and equal to $\{s(n)\}$, the FF and the FB time-varying filters of the MMSE-DFE are given by [12]

$$b_{\text{MMSE}}(n) = R_{yy}^{-1} e_0 e_0^T R_{ss}^{-1} e_0,$$  \hspace{1cm} (14)

$$f_{\text{MMSE}}(n) = R_{sy}^{-1}(n) R_{yy}^H(n) b_{\text{MMSE}}(n),$$  \hspace{1cm} (15)

where $e_0 := [1 \ 0 \ \cdots \ 0]^T$,

$$R_{\delta} := R_{ss}(n) - R_{sy}(n) R_{yy}^{-1}(n) R_{sy}^H(n)$$

$$= \Phi \left[ \frac{1}{\sigma^2_s} H(n) H^H(n) + \frac{1}{\sigma^2_s} I_{N_lf} \right]^{-1} \Phi^H.$$ 

Using (14) and (15) in (12), we have the symbol estimate $\{\hat{s}(n-d)\}$). Since the “true” channel response $\{h(n;l)\}$ is not available at the receiver, we use the channel estimates $\{\hat{h}(n;l)\}$ obtained by (11) to design the MMSE-DFE. In order to compensate for the channel estimation errors in (13), for the simulations presented in Sec. 4 we increased the variance of $v(n)$ in (13) from $\sigma^2_v$ to $\sigma^2_v + 0.01 \sigma^2_s$.

4. SIMULATION EXAMPLES

We assume $h(n;l)$ is zero-mean, complex Gaussian, and spatially white with $E\{h(n;l) h^H(n;l)\} = \sigma^2_h I_N$. We take $L = 2$ (3 taps) in (1), and $\sigma^2_s = 1/(L+1)$. For different $l's$, $h(n;l)'s$ are mutually independent and satisfy Jakes’ model. To this end, we simulate each single tap following [13]. We consider a communication system with carrier frequency of 2GHz, data rate of 40kb/s (kilo-Bauds), therefore $T_s = 25 \mu s$, and a Doppler spread $f_d = 400$ Hz ($f_dT_s = 0.01$). 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.

[Simulation results and graphs are depicted, showing the performance of the MMSE-DFE with varying SNR and data rate.]

Fig. 1. NCMSE vs SNR, under $f_dT_s = 0.01$, $m_b = 100$ or 40 with QPSK information symbols.

We evaluate the performances of various schemes by considering their normalized channel mean square error (NCMSE) and bit error rate (BER). For $T_N$ received symbols, the NCMSE is defined as

$$\text{NCMSE} := \frac{\sum_{i=1}^{M_f} \sum_{n=0}^{T_N-1} \sum_{l=0}^{L} \left\| \hat{h}^{(i)}(n;l) - h^{(i)}(n;l) \right\|^2}{\text{\sum}_{l=1}^{L} \sum_{n=0}^{T_N-1} \sum_{l=0}^{L} \left\| h^{(i)}(n;l) \right\|^2}$$

where $h^{(i)}(n;l)$ is the true channel and $\hat{h}^{(i)}(n;l)$ is the estimated channel at the $i$-th Monte Carlo run, among total $M_f$ runs. In each run, an “initialization” training mode of 200 BPSK symbols is followed by a decision-directed mode of 4000 QPSK symbols ($T_N = 4000$). All the simulation results are based on 500 runs, and we consider the performances during the decision-directed mode only. In the decision-directed mode, training sessions are also periodically sent to facilitate the EW-RLS tracking. The TM training scheme of [9], which is optimal for channels satisfying critically-sampled CE-BEM representations, is adopted. In [9] the block of $T_B$ symbols is segmented into subblocks of equal length of $m_b$ symbols consisting of an information session of $m_d$ symbols and a succeeding training session of $m_t$ symbols ($m_b = m_d + m_t$). The training session contains an impulse guarded by zeros (silent periods), which for a channel with $L + 1$ taps has the structure $[0_{1 \times L} \ \gamma \ \ 0_{1 \times L_t}]$, where $\gamma > 0$; therefore, $m_t = 2L + 1 = 5$ symbols are devoted for training and the remaining $m_d = m_b - m_t$ are available for information symbols. We assume that each information symbol has unit power, while in every
training session, we set \( \gamma = \sqrt{2L + 1} = \sqrt{5} \) so that the average power per symbol in the training sessions is equal to that in the information sessions.

![Figure 2](image-url)  
**Fig. 2.** BER vs SNR, under \( f_dT_s = 0.01, m_b = 100 \) or 40 with QPSK information symbols.

We compare the following three schemes:

1. Subblock-wise EW-RLS algorithm of [4] with \( \beta = 1 \).

2. The proposed decision-directed tracking scheme with step size \( m_s \) in the EW-RLS tracking with \( \beta = 1 \). We take the forgetting factor \( \lambda = 0.92, 0.96 \) and 0.98 for \( m_s = 2, 4, 2 \) respectively. In the figures, our approach is denoted by “DD”.

3. Perfect symbol decisions are used as training for RLS channel tracking with step size \( m_s = 2 \) and \( \beta = 1 \).

For each of the above schemes, an MMSE-DFE described in Sec. 3.4 and [3] is employed at the receiver with \( l_f = 8, \) \( b_s = 2 \), and the delay \( d = 5 \) symbols.

In Figs. 1 and 2, the performances of each scheme are compared \( f_dT_s = 0.01 \) and different SNR’s, for the sub-block size \( m_b = 40 \) or 100. For the sub-block-wise scheme of [4], frequent training sessions are required in order to track the rapid channel variations; SB with larger subblock \( m_b = 100 \) and hence less training does not work. Due to the error propagation triggered by incorrect symbol decisions in DFE, our RLS decision-directed tracking does not perform well when the SNR is low. As the SNR increases, the proposed scheme performs “closer” to the performance of the perfect decision-directed channel tracking scheme. Since the BEM coefficients are updated every \( m_s \) symbols in our approach, a “finer” channel tracking can be obtained by reducing step size \( m_s \) although the computational complexity increases. In Fig. 3 the three schemes are evaluated with multiple receive antennas \( (N = 2 \) or 3), under \( f_dT_s = 0.01 \) and different SNR’s, for the larger subblock size \( m_b = 100 \). We take the step size \( m_s = 2 \) for the decision-directed and perfect decision-directed schemes. The subblock-wise approach of [4] is still inadequate, but our RLS decision-directed tracking scheme shows significant enhancement with higher \( N \).

5. REFERENCES


