Adaptive Estimation and Equalization of Doubly-selective Fading Channels using Basis Expansion Models

by

Hyosung Kim

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 13, 2010

Keywords: basis expansion model, adaptive channel estimation, doubly-selective channels

Approved by

Jitendra K. Tugnait, Chair, James B. Davis Professor of Electrical and Computer Engineering
Stanley J. Reeves, Professor of Electrical and Computer Engineering
Shiwen Mao, Assistant Professor of Electrical and Computer Engineering

Abstract

Wireless channels, due to multipath propagation and Doppler spread, are characterized by frequency- and time-selectivity, so-called doubly-selective wireless channels. In this dissertation, we concentrate on adaptive channel estimation and equalization for communications systems over doubly selective channels, exploiting basis expansion models (BEM). Since the time-varying nature of the channel is well captured in the complex exponential basis expansion model (CE-BEM) by the known exponential 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.

First, a subblock-wise channel estimation based on CE-BEM is considered, where we track the BEM coefficients using time-multiplexed (TM) periodic training symbols. Assuming the BEM coefficients follow a first-order AR model, Kalman filtering is used to track the BEM coefficients. This first-order AR assumption, however, is not necessarily true and possibly incurs significant modeling errors in estimation. We then seek adaptive channel estimation schemes with no *a priori* model for the BEM coefficients using recursive least-square (RLS) algorithm with finite memory.

Next, taking the performance of BEM-based approach into account, we investigate an adaptive soft-in soft-out turbo equalization for coded communication systems, exploiting CE-BEM for the overall channel variations and 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 CE-BEM.

Based on the subblock-wise approach, we also propose a decision-directed tracking based on BEM, where we track the BEM coefficients using the information symbol decisions of a decision feedback equalizer (DFE) as virtual training. The time gap between symbol decisions and required channel estimates, arising from the decision-directed tracking, is bridged by CE-BEM-based channel prediction using the estimated BEM coefficients. We also adopt an exponentially-weighted (EW) RLS algorithm for our BEM-based decision-directed tracking scheme. Decision-directed tracking requires fewer training symbols compared to the training-based tracking, for the same performance.

The contribution of the proposed BEM-based channel estimation and equalization schemes is that we track the BEM coefficients in CE-BEM, not the channel taps directly, based on

the subblock-wise approach and then generate the time-varying channels via CE-BEM. Simulation examples illustrate the superior performance of our approach over several existing doubly-selective channel estimators.

Finally, we extend all the proposed channel estimation and equalization approaches to multiple-input multiple-output (MIMO) systems.

Table of Contents

List of Figures

List of Tables

$\approx$	approximately equal to
$\otimes$	Kronecker product
$\mathbf{0}_M$	$M \times 1$ all zeros vector
$\mathbf{0}_{M \times N}$	$M \times N$ all zeros matrix
a	lower case letters for scalars
$\lceil a \rceil$	integer ceiling of a
$\lfloor a \rfloor$	integer floor of a
$\lvert a \rvert$	magnitude or absolute value of a
$\mathbf{a}$	lower case letters in bold face for column vectors
$\lVert a \rVert$	Euclidean norm of a
$\lVert a \rVert_F$	Frobenius norm of a
$\mathbf{A}$	upper case letters in bold face for matrices
$\mathbf{A}^*$	complex conjugate of $\mathbf{A}$
$\mathbf{A}^\dagger$	Moore-Penrose pseudo-inverse of $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of $\mathbf{A}$
$\mathbf{A}^H$	complex conjugate transpose of $\mathbf{A}$
$\mathbf{A}^T$	transpose of $\mathbf{A}$
$\arg\max_x f(x)$	value of x for which $f(x)$ attains its maximum
$\arg\min_x f(x)$	value of x for which $f(x)$ attains its minimum
$\delta(\cdot)$	Kronecker delta function, defined as $\delta(n) = \begin{cases} 1, & \text{if } n = 0 \\ 0, & \text{otherwise} \end{cases}$
$diag\{a_1, \cdots, a_N\}$	$N \times N$ diagonal matrix with diagonal elements $a_1, \cdots, a_N$
$E\{\cdot\}$	expectation operator
$\mathbf{I}_N$	$N \times N$ identity matrix
$\max(\cdot)$	maximum value operator
$\min(\cdot)$	minimum value operator
$\mathcal{O}(\cdot)$	big **O** notation
$tr\{\mathbf{A}\}$	trace of a square matrix $\mathbf{A}$

List of Abbreviations

AM	amplitude modulation
APP	*a posteriori* probability
AR	auto-regressive
AWGN	additive white Gaussian noise
BEM	basis expansion model
BER	bit error rate
BICM	bit-interleaved coded modulation
BPSK	binary phase-shift keying
CDMA	code division multiple access
CE-BEM	complex exponential basis expansion model
CSI	channel state information
DFE	decision feedback equalizer
DFT	discrete Fourier transform
DKL-BEM	discrete Karhuen-Loève basis expansion model
DML	deterministic maximum likelihood
DPS	discrete prolate spheroidal
DPS-BEM	discrete prolate spheroidal basis expansion model
EKF	extended Kalman filter
EXIT	extrinsic information transfer
EW-RLS	exponentially-weighted recursive least square
FB	feed-back
FF	feed-forward
FIR	finite impulse response
FM	frequency modulation
ID	iterative decoding
i.i.d.	independent and identically distributed
ISI	inter-symbol interference
KD	Kalman detector
KF	Kalman filter
LLR	log-likelihood ratio
LMS	least mean squares

LS	least squares
MAP	maximum *a posteriori*
MIMO	multiple-input multiple-output
ML	maximum likelihood
MSE	mean square error
MMSE	minimum mean square error
NCMSE	normalized channel mean square error
OFDM	orthogonal frequency division multiplexing
OP-BEM	orthogonal polynomial basis expansion model
pdf	probability density function
PSAM	pilot symbol aided modulation
QAM	quadrature amplitude modulation
PSK	phase-shift keying
QPSK	quadrature phase-shift keying
RLS	recursive least-square
SfiSfo	soft-in soft-out
SIMO	single-input multiple-output
SISO	single-input single-output
SNR	signal-to-noise ratio
SW-RLS	sliding-window recursive least square
TE	turbo equalization
TM	time-multiplexed
WSS	wide-sense stationary
WSSUS	wide-sense stationary uncorrelated scattering

Chapter 1
Introduction

Wireless channels are challenging communication media with limited bandwidth, relatively low capacity per unit bandwidth, random amplitude and phase fluctuations, and inter-symbol interference (ISI). Due to multipath propagation and Doppler spread, wireless channels are characterized by frequency- and time-selectivity [60]. To design a physical link with data rates approaching the fundamental information capacity limits of the wireless channel, accurate knowledge of the channel state information (CSI) becomes a prerequisite for many physical layer approaches. At the receiver end, equalizers are usually used to compensate for the signal distortion. One may design an equalizer based on a channel estimate or directly using the received signal. Accurate modeling of the temporal evolution of the channel plays a crucial role for estimation and tracking.

Among various models, the first-order autoregressive (AR) process is often regarded as a tractable model to describe a time-varying channel, where the channel is assumed to be Markovian; that is, for the current channel sample, the effect of channel samples other than the immediately preceding one is negligible [16]. This Markovian assumption has been justified for Rayleigh fading channels in [16] by considering the mutual information between successive channel samples.

The AR models, however, have their own drawback. When time-multiplexed (TM) training is used, channel tracking may not perform well during information-symbol transmissions (information sessions) since the information data are unknown. During information sessions, channel estimates can only be obtained based on the results from training sessions [76]. This channel prediction strategy, apparently, is not appropriate to a fast-varying channel, where the AR model may lead to high estimation variance resulting in erroneous symbol detections [6]. Potential solutions lie in exploiting the detected symbols for channel tracking: In [64], a Kalman filter is used in a decision-feedback mode during information sessions; per survivor processing (PSP) is used in [43], which embeds data-aided channel estimation into the Viterbi algorithm; in [68], joint channel estimation and data detection is implemented via extended Kalman filtering. Although channel tracking can be improved by such means during information sessions, the phenomenon of error propagation can be pronounced for fast-fading channels. More accurate channel modeling becomes necessary to track fast-fading channels.

Recently, basis expansion models (BEM) have been widely investigated to represent doubly-selective channels in wireless applications [8,11,35,63,70], for which, the time-varying channel taps are expressed as superpositions of time-varying basis functions in modeling Doppler effects, weighted by time-invariant coefficients. Candidate basis functions include complex exponential (Fourier) functions [11, 70], polynomials [8], wavelets [35], and discrete prolate spheroidal sequences [63], etc. In contrast to AR models that describe temporal variation on a symbol-by-symbol update basis, a BEM depicts the evolution of the channel over a period (block) of time. Since the time-varying nature of the channel can be well captured in a BEM by the known basis functions, the BEM coefficients should evolve much more slowly than the channel taps, and hence are more convenient to track in a fast-fading environment. We will discuss the adaptive channel estimation and equalization approaches using BEM's.

1.1 Previous Work and Contributions

In this section, we summarize the previous research work on BEM-based channel estimation, equalization, and related areas.

Doubly-selective channel estimation using a complex exponential basis expansion model (CE-BEM) and time-multiplexed training is considered in [30, 70], where CE-BEM based on Fourier basis functions is applied to represent the time-variant channel. However, since the Fourier basis expansion has the major drawback that the rectangular window associated with the discrete Fourier transform (DFT) introduces spectral leakage [20], the bit error rate (BER) suffers an error floor [17, 62]. In [62, 63], the linear minimum mean-square-error (MMSE) channel estimation using discrete prolate spheroidal (DPS) sequences is considered. It is shown that DPS-BEM-based approaches outperform CE-BEM-based approaches for doubly-selective channel estimation and data detection.

Although the CE-BEM is more convenient in theoretical analysis, its modeling error is noticeable due to the spectral leakage. To mitigate this leakage, the over-sampled CE-BEM has been considered in [14] where the Doppler spectrum is said to be oversampled compared to the case in [11, 70]. In this dissertation, we will use the oversampled CE-BEM for doubly-selective channel estimation.

To acquire the channel state information at the receiver, training symbols are usually periodically inserted during transmission, which is known as pilot symbol aided modulation (PSAM) [22]. Optimization of the PSAM for CE-BEM based doubly-selective channel models has been considered in [62,70] where the time-multiplexed (TM) training sequence is designed to minimize the channel estimation mean-square-error (MSE). In the case of CE-BEM with

independent basis expansion coefficients, minimizing the channel estimation MSE is also shown in [30, 70] to be equivalent to maximizing a lower bound on the estimated channel-based average capacity. No such considerations are to be found in [62, 63] where the doubly-selective channel is represented by DPS-BEM. We adopt the TM training scheme in [30, 70] for the proposed subblock-wise tracking approach.

Decision-directed channel tracking using a polynomial BEM has been investigated in [8], where the BEM coefficients are updated every block via the RLS algorithm within a sliding window. Channel estimation using Kalman filtering and polynomial or CE-BEM's for OFDM systems has been explored in [26,41,44], among which decision-directed tracking is considered in [26, 41]. All these contributions consider block-by-block updating unlike our contribution where we exploit subblock-wise updating. The distinction is as follows: One block comprises several subblocks. For parameter identifiability, one needs the number of subblocks at least as large as the number of basis functions used for channel modeling. Data in just one subblock do not satisfy the parameter identifiability requirements. However, unlike the block-wise schemes where the receiver has to collect the whole block of data in order to generate channel estimates and perform equalization, in the proposed subblock-wise approaches the receiver is able to accomplish the two tasks after every subblock.

In our subblock-wise approaches, we first assume that the BEM coefficients (rather than the time-varying channel taps) follow a first-order AR model, and then Kalman filtering is used to track the coefficients. This first-order AR assumption, however, is not necessarily true for a real-world channel, and possibly incurs modeling error in estimation. We then seek adaptive channel estimation schemes with no *a priori* models for the BEM coefficients. Two adaptive filtering algorithms with finite memory are considered: the exponentially-weighted recursive least-squares (RLS) algorithm and the sliding-window RLS algorithm.

We also consider BEM-based approach to coded modulation communication systems using turbo equalization receiver. Turbo (iterative) equalization is a powerful suboptimal technique used in place of the computationally prohibitive but optimal maximum likelihood (ML) or maximum *a posteriori* (MAP) sequence detection based on a super trellis. 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 [1, 5]. In [71], 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). Based on this work, a variety of soft-in soft-out equalizers employing linear MMSE and decision feedback equalization (DFE) are proposed in [38, 39] and so forth.

For doubly-selective channels, an adaptive equalizer has been presented in [68], using extended Kalman filter (EKF) to incorporate channel estimation into the equalization process. This adaptive soft nonlinear Kalman equalizer jointly optimizes the estimates of the channel and data symbols in each iteration. Based on the turbo equalization approach proposed in [68] and our subblock-wise or decision-directed BEM-based approach, we present an adaptive turbo equalizer based on CE-BEM. This adaptive equalizer takes the decision of data symbols provided by the decoder as its *a priori* information and the performance can be improved iteratively. The extrinsic information transfer (EXIT) chart analyses of our BEM-based turbo equalization approach are also provided.

Based on the subblock-wise approach, decision-directed tracking of doubly-selective channels is presented using CE-BEM in Chapter 6. In [6], the decision-directed scheme was proposed that relies on the AR model only. Acting as virtual training symbols, the information symbol decisions, provided by a DFE (with delay $d \geq 0$) utilizing the estimated channel, are used in Chapter 6 to enhance the estimation of the BEM coefficients, so that much of the spectrum resource allocated to training can be saved. Although a time gap still exists between the available symbol decisions and the channel estimates required by the DFE, it can be successfully bridged by the CE-BEM-based channel prediction, without incurring much estimation variance. Our decision-directed scheme based on subblock tracking, updates the BEM coefficients of much smaller size of block than BEM period. The periodic training symbols are still necessary to recover the channel tracking from possible phase ambiguity due to the error propagation. To circumvent an arbitrary modeling error incurred due to the AR assumption for the BEM coefficients, an adaptive channel estimation scheme with no *a priori* model for the BEM coefficients is proposed, where the exponentially-weighted recursive least-squares (EW-RLS) algorithm is considered for subblock-wise channel tracking.

Computer simulation examples demonstrate the superior performance of the proposed channel estimation and equalization using CE-BEM, compared to several existing approaches.

1.2 Organization

The rest of this dissertation is organized as follows.

In Chapter 2, we provide system models and some background that we use for channel estimation and equalization in this dissertation. This chapter covers wireless channel characteristics and representations, TM training scheme, symbol detection using Kalman filter and Minimum-Mean-Square-Error Decision-Feedback Equalizer (MMSE-DFE) and turbo equalization.

In Chapter 3, a subblock-wise Kalman tracking approach to doubly-selective channel estimation is presented, exploiting the oversampled CE-BEM for the overall channel variations, and an AR model to update the BEM coefficients. The channel estimates are updated subblock-by-subblock unlike other existing block-by-block or symbol-by-symbol schemes.

In Chapter 4, we present a doubly-selective RLS channel estimator using CE-BEM, replacing Kalman filter in Chapter 3 with RLS filter so that we can avoid the modeling error incurred when using the arbitrary AR(1) model for BEM coefficients.

In Chapter 5, we propose an adaptive turbo equalization for doubly-selective channels using CE-BEM and provide EXIT chart analyses. Based on the turbo equalization approach proposed in [68], our proposed iterative turbo equalizer with nonlinear Kalman filtering jointly optimizes the estimates of BEM channel coefficients and data symbol detection in each iteration.

In Chapter 6, a decision-directed tracking approach for doubly-selective channels is considered, exploiting the CE-BEM for the overall channel variations. We track the BEM coefficients via Kalman filtering based on an AR model to update BEM coefficients, aided by symbol decisions from a DFE. We also present EW-RLS tracking with a forgetting factor, a finite-memory decision-directed tracking.

We extend all the proposed algorithms to multi-input multi-output (MIMO) scenario throughout the preceding chapters. The dissertation concludes with Chapter 7, where future research directions are also suggested.

Chapter 2
Wireless Channel and Background

In this chapter, we briefly review the wireless channel characteristics in Sec 2.1, that includes Jakes' model, which will be used as the model of the "real" channel in the simulation examples of this dissertation. In Section 2.2, we introduce the representations of time-varying channels, the AR models and CE-BEM, which are used for channel estimation purposes in different schemes. We summarize the TM training scheme in Section 2.3 introduced in [30,70]. In Section 2.4, we introduce Kalman detector (KD) and MMSE-DFE that we employ for symbol equalization at the receiver. We also present the well-known turbo equalization in Section 2.5.

2.1 Wireless Communication Channel Characteristics

Due to multipath propagation and Doppler spread, wireless channels are characterized by frequency- and time-selectivity [60]. A radio signal, experiencing distortions through transmission by fading, background noises, and interferences of every sort, becomes stochastic to an observer at the receiver. Small-scale fading, or simply fading, is the term to describe the rapid fluctuations of the amplitudes, phases, or multipath delays of a signal over a short period of time or travel distance, so that large-scale path loss may be ignored [60]. The goal of channel estimation and equalization is mainly to combat small-scale fading.

Fading can be attributed to physical factors including multipath propagation, relative motion between the transmitter and the receiver or surrounding objects, and the transmission bandwidth of the signal, etc. [65]. The presence of reflecting objects and scatterers makes the wireless channel constantly changing, which dissipates the signal energy and distorts the signal in amplitude, phase, and time. Multiple versions of the transmitted signal arrive at the receiver through different paths. The random amplitudes and phases of the different multipath components induce fading. The relative motion between the transmitter and the receiver, as well as the motion of the objects within the wireless channel, induces Doppler spreads, which are typically time-varying and become a source of fading also.

2.1.1 Rayleigh Fading

The Rayleigh distribution is commonly used to describe the statistical time-varying nature of the received envelope of a flat fading signal, or the envelope of an individual multipath component [60]. If we assume that many statistically independent scattering waves with random amplitudes and phases reach the receiver with the phases uniformly lying in $[0, 2\pi)$, and there is no dominant non-fading signal component present (no line-of-sight), by the central limit theorem, the real and imaginary parts of the sum of the scattering waves are both Gaussian. The signal envelope A obeys a Rayleigh distribution, which has a probability density function (pdf) given by

$$f_A(a) := \begin{cases} \frac{a}{\sigma^2}\exp(-\frac{a^2}{2\sigma^2}) & a \geq 0, \\ 0 & a < 0 \end{cases} \tag{2.1}$$

with σ^2 being the time-average power of the received signal before envelope detection. The phase θ of the received signal is uniformly distributed with pdf

$$f_\Theta(\theta) := \frac{1}{2\pi}, \theta \in [0, 2\pi)\,. \tag{2.2}$$

The autocorrelation function of the received signal for two-dimensional isotropic scattering and an omnidirectional receiving antenna is given by [15, 45]

$$R_A(\tau) = \sigma^2 \cos(\omega_c \tau) J_0(\omega_m \tau) \tag{2.3}$$

where ω_c is the carrier radian frequency, $J_0(\cdot)$ is the zero-order Bessel function of the first kind and ω_m is the maximum Doppler radian frequency spread. Any model that attempts to model the Rayleigh flat fading narrow-band wireless channel has to exhibit the statistical behaviors given by (2.1)-(2.3).

2.1.2 Jakes' Model

Clarke summarized the important characteristics of fading channels and provided a useful mathematical model [45]. According to this model, Jakes proposed a sum-of-sinusoids based simulator [65] that has been widely used and studied over the past decades. The simulator supposes the received signal $S(t)$ to be a superposition of waves

$$S(t) = E_0 \sum_{n=1}^{N} C_n \cos(\omega_c t + \omega_m t \cos A_n + \Phi_n) \tag{2.4}$$

where E_0 is the amplitude of the transmitted cosine wave, C_n is a random variable representing the attenuation of the n-th path, A_n is a random variable representing the angle of arrival of the n-th ray with respect to the direction of motion of the receiver, Φ_n is a random variable representing the phase shift undergone by the n-th ray. Note that the stochastic signal $S(t)$ representing the flat fading signal can be characterized by N sets of triples (C_n, A_n, Φ_n). The random variables C_n, A_n and Φ_n are assumed statistically independent.

To reduce the complexity, Jakes' model selects

$$C_n = \frac{1}{\sqrt{N}}, \qquad A_n = \frac{2\pi n}{N}, \qquad \Phi_n = 0, \tag{2.5}$$

where $n = 1, 2, \ldots, N$. Furthermore, N is of the form $N = 4M + 2$ where M is a positive integer.

However, the simplification in (2.5) makes this simulation model deterministic and wide-sense nonstationary [33, 75]. In [75], a modified Jakes' simulator was proposed. It is wide-sense stationary and its autocorrelation and cross correlation functions match the desired reference model exactly. Following [75], the normalized low-pass fading process of the statistical sum-of-sinusoids simulation model is defined by

$$X(t) = X_c(t) + jX_s(t), \tag{2.6a}$$

$$X_c(t) = \frac{2}{\sqrt{M}} \sum_{n=1}^{M} \cos(\psi_n) \cos(\omega_m t \cos\alpha_n + \phi), \tag{2.6b}$$

$$X_s(t) = \frac{2}{\sqrt{M}} \sum_{n=1}^{M} \sin(\psi_n) \cos(\omega_m t \cos\alpha_n + \phi). \tag{2.6c}$$

with

$$\alpha_n = \frac{2\pi n - \pi + \theta}{4M}, \; n = 1, 2, \ldots, M$$

where θ, ϕ and ψ_n are statistically independent and uniformly distributed over $[-\pi, \pi)$ for all n. As $M \to \infty$, the envelope $|X|$ is Rayleigh distributed and the phase $\Theta_X(t)$ is uniformly distributed over $[-\pi, \pi)$, for which the pdf's are given by

$$f_{|X|}(x) = x \, \exp(-\frac{x^2}{2}), \; x \geq 0,$$
$$f_{\Theta_X}(\theta) = \frac{1}{2\pi}, \; \theta \in [-\pi, \pi)\,.$$

A minor defect, however, occurs in model (2.6) when $\omega_m = 0$ or the Doppler spread is small: A Rayleigh distribution cannot be guaranteed [63]. This problem can be easily

resolved by replacing a common phase ϕ by ϕ_n, which is also uniformly distributed over $[-\pi, \pi)$ for all n. The simulation model is revised as [63]:

$$X(t) = X_c(t) + jX_s(t), \tag{2.7a}$$

$$X_c(t) = \frac{2}{\sqrt{M}} \sum_{n=1}^{M} \cos(\psi_n) \cos(\omega_m t \cos \alpha_n + \phi_n), \tag{2.7b}$$

$$X_s(t) = \frac{2}{\sqrt{M}} \sum_{n=1}^{M} \sin(\psi_n) \cos(\omega_m t \cos \alpha_n + \phi_n). \tag{2.7c}$$

2.2 Representations of Wireless Channels

For channel estimation or tracking purposes, accurate modeling of the temporal evolution of the channel plays an important role. A parsimonious and accurate channel representation is always preferred. Among various models for channel time variations, the AR process, particularly the first-order AR model, is regarded as a tractable model to describe a time-varying channel, where the channel is assumed to be Markovian, i.e., for the current channel symbol, the effect of channel symbols other than the immediately preceding one is negligible [16]. This Markovian assumption has been verified for Rayleigh fading channels in [16], by considering the mutual information between channel symbols. The AR model has been used for time varying channel estimation in [31, 43, 64, 68, 76].

The AR model, based on symbol-by-symbol update, is suitable for sequential time-domain processing. When we deal with block processing schemes, it is often more convenient to use a block-based channel models such as BEM's. The BEM that is optimal in MSE is the discrete Karhuen-Loève BEM (DKL-BEM), which is a reduced-rank decomposition of a certain type of Doppler spectrum [77]. The CE-BEM can be viewed as a special DKL-BEM based on a white Doppler spectrum, and the DPS-BEM corresponds to the DKL-BEM with a rectangular Doppler spectrum [77].

2.2.1 Autoregressive (AR) Model

It is possible to accurately represent a wide-sense stationary uncorrelated scattering (WSSUS) channel by a large order AR model; see [6, 68] and references therein. Let

$$\tilde{\mathbf{h}}(n) := [h(n;0)\ h(n;1)\ \cdots\ h(n;l)]^T \tag{2.8}$$

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

$$\tilde{\mathbf{h}}(n) = \sum_{i=1}^{P} \mathbf{A}_i \tilde{\mathbf{h}}(n-i) + \mathbf{G}_0 \tilde{\mathbf{w}}(n) \tag{2.9}$$

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 independent and identically distributed (i.i.d.) $(L+1)\times 1$ driving noise $\tilde{\mathbf{w}}(n)$ is zero-mean with identity covariance matrix. Suppose that we know the correlation function $\mathbf{R}_h(\tau) = E\left\{\tilde{\mathbf{h}}(n+\tau)\tilde{\mathbf{h}}^H(n)\right\}$ for lags $\tau = 0, 1, \cdots, P$. The following Yule-Walker equation holds for (2.9) [55]:

$$\mathbf{R}_h(\tau) = \sum_{i=0}^{P} \mathbf{A}_i \mathbf{R}_h(\tau - i) + \mathbf{G}_0 \mathbf{G}_0^H \delta(\tau). \tag{2.10}$$

Using (2.10) 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 (2.10) for $\tau = 0$, one can find $\mathbf{G}_0\mathbf{G}_0^H$, from which one can find (non-unique) $\mathbf{G}_0$ by computing its "square root" [42, p.358]. In [6, 68] only AR(1) or AR(2) models have been used.

In this dissertation, we will use AR models for some simulation comparisons where various channel taps are assumed to be mutually statistically independent. In this case we have an independent AR process for each channel tap. Furthermore, following [6, 68], we consider the first-order AR models, given by

$$h(n;l) = \tilde{\alpha}_l h(n-1;l) + \tilde{w}_l(n), \tag{2.11}$$

where $\tilde{\alpha}_l$ is the AR coefficient and the driving noise $\tilde{w}_l(n)$ is zero-mean white with variance $\sigma^2_{\tilde{w}l}$. If we assume that $h(n;l)$ is also zero-mean with variance σ^2_{hl}, then one picks (to match correlation functions at lags 0 and 1) [76]

$$\tilde{\alpha}_l = \frac{1}{\sigma^2_{hl}} E\left\{h(n;l)h^*(n-1;l)\right\}, \tag{2.12}$$

$$\sigma^2_{\tilde{w}l} = \sigma^2_{hl}(1 - |\tilde{\alpha}_l|^2). \tag{2.13}$$

It must be noted that in practice, one would not know $\mathbf{R}_h(\tau)$.

2.2.2 Complex Exponential Basis Expansion Model (CE-BEM)

Recently, deterministic CE-BEM have been widely investigated in wireless applications, especially when the multipath is caused by a few strong reflectors, and path delays exhibit variations due to the kinematics of the mobiles [11]. In these models, the time-varying taps

are expressed as a superposition of time-varying basis functions in modeling Doppler effects, with time-invariant coefficients. By assigning temporal variations to basis functions, rapidly fading channels with coherence time as small as a few tens of symbols can be captured. If the delay spread and the Doppler spread of the channel (or at least the upper bounds of them) are known, one can infer the basis functions of the CE-BEM [70]. Treating the basis functions as known parameters, estimation of a time-varying process is reduced to estimate time-invariant coefficients.

Consider a time-varying channel with impulse response $h(t;\tau)$ (response at time t to a unit impulse at time $t-\tau$) which includes transmit-receive filters as well as doubly-selective propagation effects. Let $s(t)$ denote the complex baseband, continuous-time input signal (with symbol duration T_s), and $x(t)$ denote the complex baseband, continuous-time received signal. The noise-free received signal $x(t)$ is the convolution of $s(t)$ and $h(t;\tau)$ [19]:

$$x(t) = \int_0^{\infty} h(t;\tau)s(t-\tau)d\tau. \tag{2.14}$$

Let $H(f;\tau) = \int_{\infty}^{-\infty} h(t;\tau)e^{-j2\pi ft}dt$ be the Fourier transform of $h(t;\tau)$. If $|H(f;\tau)| \approx 0$ for $|\tau| > \tau_d$, then τ_d is defined as the delay-spread of the channel; if $|H(f;\tau)| \approx 0$ for $|f| > f_d$, then f_d is defined as the Doppler spread of the channel [70]. Sampling $s(t), x(t)$ and $h(t;\tau)$ in (2.14) at the symbol rate, then for $t = nT_s \in [t_0, t_0 + TT_s)$, the sampled signal $x(n) := x(t)|_{t=nT_s}$ has the representation

$$x(n) = \sum_{l=0}^{L} h(n;l)s(n-l). \tag{2.15}$$

Over the block interval of $[t_0, t_0 + TT_s)$, the channel impulse response $\{h(n;l)\}_{n=0}^{T-1}$can be represented by Q coefficients $\{h_q(l)\}_{q=1}^{Q}$ (which remain invariant throughout this block but are allowed to change at the next block) and the corresponding Q Fourier basis functions that are common for each block. Then over the interval $[t_0, t_0 + TT_s)$, the discrete-time baseband equivalent channel model for the block can be described as [69, 70]:

$$h(n;l) = \sum_{q=1}^{Q} h_q(l)e^{j\omega_q n}, \quad l = 0, 1, \cdots, L \tag{2.16}$$

where $n = (i-1)T_B, (i-1)T_B + 1, \cdots, iT_B - 1$ and one chooses (Λ is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{2.17}$$

$$Q \geq 2\lceil f_d T T_s \rceil + 1, \tag{2.18}$$

$$\omega_q := \frac{2\pi}{T}\left[q - (Q+1)/2\right], \quad q = 1, 2, \cdots, Q \tag{2.19}$$

$$L := \lfloor \tau_d / T_s \rfloor, \tag{2.20}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration. The BEM coefficients $h_q(l)$'s remain invariant during this block, but are allowed to change at the next consecutive block; the Fourier basis functions $\{e^{j\omega_q n}\}$ ($q = 1, 2, \cdots, Q$) are common for every block. If the delay spread and the Doppler spread (or at least their upper bounds) are known, one can infer the basis functions of the CE-BEM [70]. Treating the basis functions as known, estimation of a time-varying process is reduced to estimating the invariant coefficients over a block of T_B symbols.

If $\Lambda > 1$ (e.g. $\Lambda = 2$ or 3), then the Doppler spectrum is said to be oversampled [14] compared to the case $\Lambda = 1$ where the Doppler spectrum is said to be critically sampled. In [11, 70] only $\Lambda = 1$ (henceforth called CE-BEM) is considered whereas [14] considers $\Lambda \geq 2$ (henceforth called over-sampled CE-BEM). For $\Lambda = 1$, the rectangular window of this truncated discrete Fourier transform (DFT)-based model introduces spectral leakage. To mitigate this leakage, the over-sampled CE-BEM with $\Lambda = 2$ or 3 has been considered in [14] although the basis functions are no longer orthogonal by over-sampling.

The model (2.16) is periodic with period T. Later we will use it for all time n (not just the block size T_B) by allowing the BEM coefficients $h_q(l)$s to change with time. So long as the effective "memory" of the "processors" used later is less than the model period (recall that the channel is by no means periodic), there are no deleterious effects due to the use of (2.16) for all time.

2.3 Block-Adaptive Channel Estimation using CE-BEM [30, 70]

In this section, we summarize the time-multiplexed (TM) training scheme in [30,70] that we employ in our subblock-wise tracking approach. Each transmitted block of symbols $s(n)$ of length T_B symbols is segmented into $\bar{P}$ subblocks of length $m_b := T_B/\bar{P}$ symbols, which has the time-multiplexed m_d information symbols and m_t training symbols ($m_b = m_d + m_t$). If $\mathbf{s}$ denotes a column-vector composed of $\{s(n)\}_{n=(i-1)T_B}^{iT_B-1}$ for i-th block, then it is arranged as

$$\mathbf{s} := \begin{bmatrix} \mathbf{b}_0^T & \mathbf{c}_0^T & \mathbf{b}_1^T & \mathbf{c}_1^T & \cdots & \mathbf{b}_{\bar{P}-1}^T & \mathbf{c}_{\bar{P}-1}^T \end{bmatrix}^T \tag{2.21}$$

where $\mathbf{b}_p$ is a column of m_d information symbols and $\mathbf{c}_p$ is a column of m_t training symbols for $p = 0, 1, \cdots, \bar{P} - 1$. Based on CE-BEM (2.16), [70] has shown that for $\Lambda = 1$ (the critically-sampled CE-BEM) and $\bar{P} \geq Q$, the optimal session contains an impulse guarded by zeros (silent periods), which has the structure

$$\mathbf{c}_p := \begin{bmatrix} \mathbf{0}_L^T & \gamma & \mathbf{0}_L^T \end{bmatrix}^T, \ \gamma > 0 \tag{2.22}$$

with $m_t = 2L + 1$. Thus, given a transmission block of size T_B, $(2L + 1)\bar{P}$ symbols have to be devoted for training and the remaining are available for information symbols. Let $n_p := pm_b + m_d + L$, $(p = 0, 1, \cdots, \bar{P} - 1)$, denote the unique locations of (nonzero) γ's in the $\mathbf{c}_p$'s in the $\bar{P}$ subblocks. Let $n_p := pm_b + m_d + L$, $(p = 0, 1, \cdots, \bar{P} - 1)$, denote the unique locations of (nonzero) γ's in the $\mathbf{c}_p$'s in the $\bar{P}$ subblocks.

For multiple users, let $\{s_k(n)\}$ denote k-th user's information sequence for $k = 1, 2, \cdots, K$. If $\mathbf{s}_k$ denotes a column-vector composed of $\{s_k(n)\}_{n=(i-1)T_B}^{iT_B-1}$ for i-th block, then it is arranged as

$$\mathbf{s}_k := \begin{bmatrix} \mathbf{b}_{k,0}^T & \mathbf{c}_{k,0}^T & \mathbf{b}_{k,1}^T & \mathbf{c}_{k,1}^T & \cdots & \mathbf{b}_{k,\bar{P}-1}^T & \mathbf{c}_{k,\bar{P}-1}^T \end{bmatrix}^T \tag{2.23}$$

where $\mathbf{b}_{k,p}$ is a column of m_d information symbols and $\mathbf{c}_{k,p}$ is a column of m_t training symbols for $p = 0, 1, \cdots, \bar{P} - 1$. We clearly have $T_B = \bar{P}m_b$. The training session for each user contains an impulse guarded by zeros (silent periods), which for the k-th user has the structure ($\gamma > 0$)

$$\mathbf{c}_{k,p} := \begin{bmatrix} \mathbf{0}_{(k-1)(L+1)+L}^T & \gamma & \mathbf{0}_{(K-k)(L+1)+L}^T \end{bmatrix}^T. \tag{2.24}$$

Therefore, $m_t = K(L + 1) + L$ symbols, which have to be devoted for training and the remaining are available for information symbols. Let $n_{k,p} := pm_b + m_d + (k - 1)(L + 1) + L$, $(p = 0, 1, \cdots, \bar{P} - 1)$ denote the unique locations of (nonzero) γ's in the $\mathbf{c}_{k,p}$'s in the $\bar{P}$ subblocks for k-th user.

As shown in [18,70], one needs to satisfy number of subblocks $\bar{P} \geq Q$ to uniquely identify the unknown BEM parameters. By (2.18), as T increases, $Q = 2\lceil f_d T T_s \rceil + 1$ increases (in discrete steps) for a given value of $f_d T_s$. Since the subblock size $m_b = T_B/\bar{P}$, therefore, $\bar{P} = T_B/m_b = T/(\Lambda m_b)$, as T increases, $\bar{P}$ also increases. Typically, one would first pick T_B (which would fix $T = \Lambda T_B$ with $\Lambda = 2$ for oversampled CE-BEM), and then pick m_b to satisfy $\bar{P} \geq Q$ for a given value of T or T_B, Q does not depend upon $\bar{P}$. If m_b turns out to be "too small", training overhead is large since $2L + 1$ symbols out of m_b are needed for training. On the other hand, suppose one first picks m_b to achieve a certain training overhead $(= (2L + 1)/m_b)$. Then in order to have parameter identifiability, one needs $\bar{P} \geq Q$, i.e., $T/(\Lambda m_b) \geq 2\lceil f_d T T_s \rceil + 1$. For example, under $T_s = 25\mu s$, $f_d = 400Hz$ and $m_b = 40$, there

exists no integer T with $\Lambda = 2$ for which we can satisfy $T/(\Lambda m_b) \geq 2\lceil f_d T T_s \rceil + 1$ whereas $T = 400$ satisfies this inequality when $\Lambda = 1$ ($T_B = 400, \bar{P} = 10$ and $Q = 9$). This implies that for block-wise updating, we cannot always use oversampled CE-BEM, only the less accurate critically sampled CE-BEM.

2.4 Useful Equalization Techniques

In communication systems, the channel estimates given by a channel estimator are fed into an equalizer for symbol detection. In this section, we introduce two equalizers, Kalman detector (KD) and Minimum Mean Square Error (MMSE) Decision Feedback Equalizer (DFE), that we employ in this paper to detect the transmitted symbols using the channel estimates. We consider a multi-input multi-output (MIMO) system with K inputs and N outputs for both equalizers; one can adapt easily to single-input single-output (SISO) or single-input multi-output (SIMO) systems.

Let $\{s_k(n)\}$ denote k-th user's information sequence that is input to the time-varying channel with discrete-time response $\{\mathbf{h}_k(n;l)\}$ (channel response for the k-th user at time instance n to a unit input at time instance $n-l$). We assume $\{s_k(n)\}$ is mutually independent and identically distributed (i.i.d.) with zero mean and variance $E\{s_k(n)\, s_k^*(n)\} = \sigma_{s_k}^2 = \sigma_s^2$ for $k = 1, 2, \cdots, K$. Then the symbol-rate noisy N-column channel output vector is given by $(n = 0, 1, \ldots)$

$$\mathbf{y}(n) = \sum_{k=1}^{K} \sum_{l=0}^{L} \mathbf{h}_k(n;l)\, s_k(n-l) + \mathbf{v}(n) \tag{2.25}$$

where the N-column vector $\mathbf{v}(n)$ is zero-mean, white, uncorrelated with $s_k(n)$, complex Gaussian noise, with the autocorrelation $E\{\mathbf{v}(n+\tau)\, \mathbf{v}^H(n)\} = \sigma_v^2 \mathbf{I}_N \delta(\tau)$. Define

$$\begin{aligned} \mathbf{s}(n) &:= \begin{bmatrix} s_1(n) & s_2(n) & \cdots & s_K(n) \end{bmatrix}^T \\ \mathbf{h}(n;l) &:= \begin{bmatrix} \mathbf{h}_1(n;l) & \mathbf{h}_2(n;l) & \cdots & \mathbf{h}_K(n;l) \end{bmatrix}. \end{aligned}$$

and we may rewrite (2.25) as

$$\mathbf{y}(n) = \sum_{l=0}^{L} \mathbf{h}(n;l)\, \mathbf{s}(n-l) + \mathbf{v}(n). \tag{2.26}$$

2.4.1 Kalman Detector (KD)

The Kalman filter, together with a quantizer, act as the symbol detector at the receiver end. We transform the Kalman filter for joint channel estimation and equalization in [68] to

KD only for symbol equalizaton. The state and the measurement equations are given by

$$\mathbf{S}_d(n) = \mathbf{\Phi}\mathbf{S}_d(n-1) + \mathbf{\Gamma}\bar{\mathbf{s}}(n) + \mathbf{\Gamma}\tilde{\mathbf{s}}(n), \tag{2.27}$$

$$\mathbf{y}(n) = \tilde{\mathbf{H}}_d(n)\mathbf{S}_d(n) + \mathbf{v}(n), \tag{2.28}$$

with the following definitions

$$\begin{aligned}
\mathbf{S}_d(n) &:= \begin{bmatrix}\mathbf{s}^T(n) & \mathbf{s}^T(n-1) & \cdots & \mathbf{s}^T(n-d)\end{bmatrix}^T,\\
\bar{\mathbf{s}}(n) &:= E\{\mathbf{s}(n)\}, \qquad \tilde{\mathbf{s}}(n) := \mathbf{s}(n) - \bar{\mathbf{s}}(n),\\
\mathbf{\Phi} &:= \begin{bmatrix}\mathbf{0}_d^T & 0\\ \mathbf{I}_d & \mathbf{0}_d\end{bmatrix} \otimes \mathbf{I}_K, \quad \mathbf{\Gamma} := \begin{bmatrix}1 & \mathbf{0}_d^T\end{bmatrix}^T \otimes \mathbf{I}_K,\\
\tilde{\mathbf{H}}_d(n) &:= \begin{bmatrix}\hat{\mathbf{h}}(n;0) & \hat{\mathbf{h}}(n;1) & \cdots & \hat{\mathbf{h}}(n;L) & \mathbf{0}_{N\times K(d-L)}\end{bmatrix}
\end{aligned}$$

where $\mathbf{s}(n)$ is K-column vector of symbols, $\hat{\mathbf{h}}(n;l)$ is the estimated $N \times K$ channel matrix and integer $d \geq L$; it will also be the equalization delay. Assume data symbols are zero-mean and white. If $s(n)$ is a data symbol, we have $\bar{s}(n) = 0, \tilde{s}(n) = s(n)$ and $\sigma_s^2(n) = \sigma_s^2$; if $s(n)$ is a training symbol, $\bar{s}(n) = s(n), \tilde{s}(n) = 0$ and $\sigma_s^2(n) = 0$.

Kalman filtering for the system described by (2.27) and (2.28) is initialized with

$$\hat{\mathbf{S}}_d(-1 \mid -1) = \mathbf{0}_{K(d+1)} \text{ and } \mathbf{P}(-1 \mid -1) = \mathbf{\Gamma}\mathbf{\Gamma}^T,$$

where $\hat{\mathbf{S}}_d(p \mid m)$ denotes the estimate of $\mathbf{S}_d(p)$ given the observations $\{\mathbf{y}(n)\}_{n=0}^m$, and $\mathbf{P}(p \mid m)$ denotes the error covariance matrix of $\hat{\mathbf{S}}_d(p \mid m)$, defined as

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

Then recursive filtering (for $n = 0, 1, \cdots$) is applied via the following steps:

1. Time update:

$$\begin{aligned}
\hat{\mathbf{S}}_d(n \mid n-1) &= \mathbf{\Phi}\hat{\mathbf{S}}_d(n-1 \mid n-1) + \mathbf{\Gamma}\bar{\mathbf{s}}(n),\\
\mathbf{P}(n \mid n-1) &= \mathbf{\Phi}\mathbf{P}(n-1 \mid n-1)\mathbf{\Phi}^T + \sigma_s^2(n)\mathbf{\Gamma}\mathbf{\Gamma}^T;
\end{aligned}$$

2. Kalman gain:

$$\begin{aligned}
\mathbf{P}_\eta(n) &= \sigma_v^2\mathbf{I}_N + \tilde{\mathbf{H}}_d(n)\mathbf{P}(n \mid n-1)\tilde{\mathbf{H}}_d^H(n),\\
\mathbf{K}(n) &= \mathbf{P}(n \mid n-1)\tilde{\mathbf{H}}_d^H(n)\mathbf{P}_\eta^{-1}(n);
\end{aligned}$$

3. Measurement update:

$$\hat{\mathbf{S}}_d(n \mid n) = \hat{\mathbf{S}}_d(n \mid n-1) + \mathbf{K}(n)\left\{\mathbf{y}(n) - \tilde{\mathbf{H}}_d(n)\hat{\mathbf{S}}_d(n \mid n-1)\right\},$$
$$\mathbf{P}(n \mid n) = \left\{\mathbf{I}_{K(d+1)} - \mathbf{K}(n)\tilde{\mathbf{H}}_d(n)\right\}\mathbf{P}(n \mid n-1).$$

After Kalman filtering for every n, the estimated vector is given as

$$\hat{\mathbf{S}}_d(n \mid n) = \left[\hat{\mathbf{s}}^T(n \mid n) \quad \hat{\mathbf{s}}^T(n-1 \mid n) \quad \cdots \quad \hat{\mathbf{s}}^T(n-d \mid n)\right]^T$$

and we extract its last (K-column vector) term $\hat{\mathbf{s}}(n-d \mid n)$ as the desired equalized output for K-users. Finally, we hard-quantize $\hat{\mathbf{s}}(n-d \mid n)$ to acquire the detected symbols.

Using the estimated channel, one may rewrite the received signal as

$$\mathbf{y}(n) = \sum_{l=0}^{L} \hat{\mathbf{h}}(n;l)\,\mathbf{s}(n-l) + \underbrace{\sum_{l=0}^{L}\left[\mathbf{h}(n;l) - \hat{\mathbf{h}}(n;l)\right]\mathbf{s}(n-l) + \mathbf{v}(n)}_{=:\tilde{\mathbf{v}}(n)} \tag{2.29}$$

where the "effective" noise is $\tilde{\mathbf{v}}(n)$ instead of $\mathbf{v}(n)$. In order to compensate for this channel estimation error, that is because the channel estimates $\left\{\hat{\mathbf{h}}(n;l)\right\}$ obtained by a channel estimator is not equal to the "true" channel response $\{\mathbf{h}(n;l)\}$, we take the variance of $\tilde{\mathbf{v}}(n)$ in (2.29) to be $\sigma_v^2 + 0.01\sigma_s^2$ instead of the variance of $\mathbf{v}(n)$, σ_v^2, for simulations presented in Section 3.3.2.

2.4.2 Mimimum Mean Square Error Decision Feedback Equalizer (MMSE-DFE) [40]

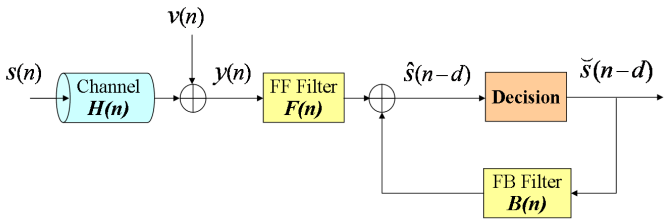


Figure 2.1: Decision-feedback equalizer (DFE).

The DFE structure is shown in Fig. 2.1 to equalize the delayed symbols $\mathbf{s}(n-d)$, with the feed-forward (FF) and feed-back (FB) filters. Since each measurement $\mathbf{y}(n)$ contains inter-symbol-interference (ISI) caused by prior symbols, DFE is designed to reduce ISI and

to recover $\mathbf{s}(n)$ using FIR filters. The FF filter takes current and prior measurements $\mathbf{y}_f$ as its input to get information correlated with ISI and to remove its effect.

Stack the inputs of the FF filter with l_f taps at time n into a "tall" vector

$$\mathbf{y}_f(n) := \begin{bmatrix} \mathbf{y}^T(n) & \mathbf{y}^T(n-1) & \cdots & \mathbf{y}^T(n-l_f+1) \end{bmatrix}^T,$$

where $\mathbf{y}(n)$ is N -column vector and also define $\mathbf{v}_f(n)$ likewise. Then, the received signal is given as

$$\mathbf{y}_f(n) = \mathbf{H}(n)\,\mathbf{s}_f(n) + \mathbf{v}_f(n), \tag{2.30}$$

with

$$\mathbf{H}(n) := \begin{bmatrix} \mathbf{h}(n;0) & \cdots & \mathbf{h}(n;L) & & \\ & \ddots & & \ddots & \\ & & \mathbf{h}(n-l_f+1;0) & \cdots & \mathbf{h}(n-l_f+1;L) \end{bmatrix}$$
$$\mathbf{s}_f(n) := \begin{bmatrix} \mathbf{s}^T(n) & \mathbf{s}^T(n-1) & \cdots & \mathbf{s}^T(n-l_f-L+1) \end{bmatrix}^T.$$

where $\mathbf{h}(n;l)$ is $N \times K$ matrix channel response at time n to a unit input at time $n-l$ and $\mathbf{s}(n)$ is K-column vector.

As shown in Fig. 2.1, the input to the FB filter comes from the decision output, denoted by $\breve{\mathbf{s}}_f(n-d)$. The FB filter uses prior symbol decisions to cancel the trailing ISI by mapping the estimate $\hat{\mathbf{s}}_f(n-d)$ to the closest point in the symbol constellation. We define the input vector of FB filter with l_b taps as

$$\mathbf{s}_b(n) := \begin{bmatrix} \breve{\mathbf{s}}^T(n-d) & \breve{\mathbf{s}}^T(n-d-1) & \cdots & \breve{\mathbf{s}}^T(n-d-l_b) \end{bmatrix}^T.$$

The estimate of the information symbol, $\hat{\mathbf{s}}(n-d)$ is obtained by combining the outputs of FF and FB filters and can be written at time n with delay d as

$$\hat{\mathbf{s}}(n-d) = \sum_{i=0}^{l_f-1} \mathbf{F}_i^T(n)\,\mathbf{y}(n-i) - \sum_{j=1}^{l_b} \mathbf{B}_j(n)\,\breve{\mathbf{s}}(n-d-j), \tag{2.31}$$

where $\mathbf{F}_i(n)$'s (that are $N \times K$ matrices) and $\mathbf{B}_j(n)$'s (that are $K \times K$ matrices) are the taps of FF and FB time-varying filters at time n, and $\breve{\mathbf{s}}(n-d-j)$ is the hard decision of $\hat{\mathbf{s}}(n-d-j)$. The estimate $\hat{\mathbf{s}}(n-d)$ is also fed into the quantizer to obtain the symbol decision $\breve{\mathbf{s}}(n-d)$.

Let $\mathbf{F}(n)$ and $\mathbf{B}(n)$ denote the vectors of time-varying taps of FF and FB filters,

$$\mathbf{F}(n) := \begin{bmatrix} \mathbf{F}_0^T(n) & \mathbf{F}_1^T(n) & \cdots & \mathbf{F}_{l_f-1}^T(n) \end{bmatrix}^T,$$
$$\mathbf{B}(n) := \begin{bmatrix} \mathbf{I} & \mathbf{B}_1(n) & \mathbf{B}_2(n) & \cdots & \mathbf{B}_{l_b}(n) \end{bmatrix}^T,$$

then the error signal is given by

$$\begin{aligned} \tilde{\mathbf{s}}(n-d) &= \mathbf{s}(n-d) - \hat{\mathbf{s}}(n-d) \\ &= \mathbf{B}(n)\mathbf{s}_b(n) - \mathbf{F}(n)\mathbf{y}_f(n). \end{aligned} \tag{2.32}$$

Assuming the decisions $\{\check{\mathbf{s}}(n)\}$ are correct and equal to $\{\mathbf{s}(n)\}$, we can solve a nonlinear optimization problem which mininmizes the variance of the error signal in (2.32),

$$\min_{\{\mathbf{F}(n),\mathbf{B}(n)\}} E\left\{\left|\mathbf{B}(n)\mathbf{s}_b(n) - \mathbf{F}(n)\mathbf{y}_f(n)\right|^2\right\}. \tag{2.33}$$

Solve a standard linear least-mean-squares estimation problem over $\mathbf{F}(n)$ with $\mathbf{B}(n)$ fixed and then we have a constrained optimization problem; note that the leading entry of $\mathbf{B}(n)$ is the identity matrix. Therefore, the FF and the FB time-varying filters of the MMSE-DFE are given by [40]

$$\mathbf{B}_{\mathrm{MMSE}}(n) = \mathbf{R}_\delta^{-1}\mathbf{\Psi}\left(\mathbf{\Psi}^T\mathbf{R}_\delta^{-1}\mathbf{\Psi}\right)^{-1}, \tag{2.34}$$
$$\mathbf{F}_{\mathrm{MMSE}}(n) = \mathbf{R}_{yy}^{-1}(n)\,\mathbf{R}_{sy}^H(n)\,\mathbf{B}_{\mathrm{MMSE}}(n), \tag{2.35}$$

where

$$\mathbf{\Psi} := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}^T \otimes \mathbf{I}_K,$$
$$\begin{aligned} \mathbf{R}_\delta &:= \mathbf{R}_{ss}(n) - \mathbf{R}_{sy}(n)\,\mathbf{R}_{yy}^{-1}(n)\,\mathbf{R}_{sy}^H(n) \\ &= \mathbf{\Phi}\left[\frac{1}{\sigma_v^2}\mathbf{H}(n)\,\mathbf{H}^H(n) + \frac{1}{\sigma_s^2}\mathbf{I}_{Nl_f}\right]^{-1}\mathbf{\Phi}^H. \end{aligned}$$

By the assumption that $\{\mathbf{s}(n)\}$ are independent and identically distributed (i.i.d.) with variance σ_s^2, and based on (2.30), we have

$$\mathbf{R}_{ss}(n) := E\left\{\mathbf{s}_b(n)\,\mathbf{s}_b^H(n)\right\} = \sigma_s^2\mathbf{I}_{K(l_b+1)},$$
$$\mathbf{R}_{sy}(n) := E\left\{\mathbf{s}_b(n)\,\mathbf{y}_f^H(n)\right\} = \sigma_s^2\mathbf{\Phi}\mathbf{H}^H(n),$$
$$\mathbf{R}_{yy}(n) := E\left\{\mathbf{y}_f(n)\,\mathbf{y}_f^H(n)\right\} = \sigma_s^2\mathbf{H}(n)\,\mathbf{H}^H(n) + \sigma_v^2\mathbf{I}_{Nl_f}$$

where

$$\mathbf{s}_b(n) = \mathbf{\Phi}\mathbf{s}_f(n), \quad \mathbf{\Phi} := \left[\mathbf{0}_{(l_b+1)\times d} \quad \mathbf{I}_{l_b+1} \quad \mathbf{0}_{(l_b+1)\times\left(l_f+L-d-l_b-1\right)}\right] \otimes \mathbf{I}_K.$$

Using (2.34), (2.35) in (2.31), we have the symbol estimate $\{\hat{\mathbf{s}}\,(n-d)\}$. Since the "true" channel response $\{\mathbf{h}\,(n;l)\}$ is not available at the receiver, we use the channel estimates $\{\hat{\mathbf{h}}\,(n;l)\}$ obtained by a channel estimator to design the MMSE-DFE [See (2.29) in 2.4.1 for a formal equation]. In order to compensate for this channel estimation error in (2.30), for simulations where we use MMSE-DFE, we increased the variance of $\mathbf{v}\,(n)$ in (2.30) from σ_v^2 to $\sigma_v^2 + 0.01\sigma_s^2$ in our simulation presented later.

2.5 Turbo Equalization

In digital communications, a turbo equalizer is a type of receiver used to detect a message corrupted by a communication channel with ISI. It approaches the performance of a maximum *a posteriori* (MAP) receiver via iterative message passing between a soft-in soft-out (SfiSfo) equalizer and a SfiSfo decoder [46]. It is closely related to turbo codes, as a turbo equalizer may be considered a turbo decoder if the channel is viewed as a convolutional code.

The optimal equalization methods for minimizing sequence error rate or the bit error rate (BER) are based on maximum likelihood (ML) estimation, which turns into MAP estimation in the presence of *a priori* information about the transmitted data. For instance, see Viterbi algorithm (VA) [13, 19, 54] for MAP/ML sequence estimation and BCJR algorithm [29] for MAP/ML symbol estimation.

Communicating soft information probability distribution between the equalizer and the decoder, instead of hard information (symbol estimates only), improves the BER performance but usually requires more complex decoding algorithms. State-of-the-art systems for a variety of communication channels employ convolutional codes and ML equalizers together with an interleaver after the encoder and a deinterleaver before the decoder [66, 74]. Interleaving shuffles symbols within a given time frame or block of data and thus decorrelates error events introduced by the equalizer betweeen neigboring symbols.

An optimal joint processing of the equalization and decoding steps is usually impossible due to complexity considerations. A number of iterative receiver algorithms repeat the equalization and decoding tasks on the same set of received data, where feedback information from the decoder is incorporated into the equalization process. This method, called turbo equalization, was originally developed for concatenated convolutional codes (turbo code [4]) and is now adapted to various communication problems.

The MAP/ML-based solutions often suffer from high computational load for channels with long memory or large constellation sizes (expensive equalizer) or convolutional codes with long memory (expensive decoder). This situation is exacerbated by the need to perform equalization and decoding several times for each block of data. A major research issue is thus the complexity reduction of such iterative algorithms.

2.5.1 Principle of Turbo Equalization [39]

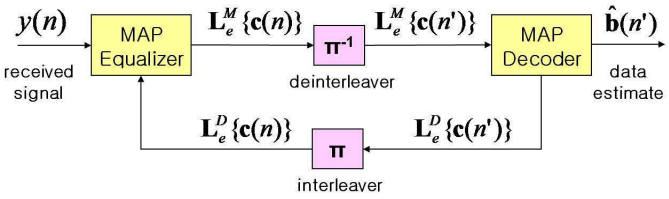


Figure 2.2: A turbo equalization receiver

We consider a simple transmitter where a sequence of data $\mathbf{b}(n') = [b^1(n'), b^2(n'), \cdots, b^{k_0}(n')] \in \{1,0\}^{k_0}$ is encoded to the code symbols $\mathbf{c}(n') = [c^1(n'), c^2(n'), \cdots, c^{n_0}(n')] \in \{1,0\}^{n_0}$ with a code rate $R_c = k_0/n_0$, which is interleaved in a block and then mapped into binary phase shift keying (BPSK) symbols. Fig. 2.2 depicts the receiver structure for turbo equalization [5], where both the SfiSfo equalizer and SfiSfo decoder are of the MAP type. The extrinsic log-likelihood ratios (LLR's), $\mathbf{L}_e\{\mathbf{c}(\cdot)\}$ are transfered iteratively between the equalizer and decoder. [The subscript "e" for representing "extrinsic", the superscript "E" and "D" for output of "Equalizer" and "Decoder" respectively.] The MAP equalizer computes the *a posteriori* probabilities (APP's) given T_r received symbols, $P\{c^i(n) = b \mid \mathbf{y}(l), 1 \le l \le T_r\}, b \in \{1,0\}$ and generates the extrinsic LLR as (*a posteriori* LLR – *a priori* LLR)

$$L_e^E\left\{c^i(n)\right\} := \ln \frac{P\{c^i(n)=1 \mid \mathbf{y}(l),\ 0 \le l < T_r\}}{P\{c^i(n)=0 \mid \mathbf{y}(l),\ 0 \le l < T_r\}} - \underbrace{\ln \frac{P\{c^i(n)=1\}}{P\{c^i(n)=0\}}}_{=:L\{c^i(n)\}}. \tag{2.36}$$

The *a priori* LLR of the MAP equalizer, $L\{c^i(n)\}$ is provided by the interleaved output of the MAP decoder at the previous iteration but $L\{c^i(n)\} = 0$ for the first iteration. The extrinsic LLR's $\mathbf{L}_e^E\{\mathbf{c}(n)\}$ produced by the MAP demodulator is sent to the MAP decoder as the *a priori* LLR's for channel decoding. Based on the *a priori* LLR's and the channel code constraints, the MAP decoder computes the APPs $P\left\{c^i(n') = b \mid \mathbf{L}_e^E\{\mathbf{c}(l)\}, 1 \le l \le T_r\right\}, b \in$

$\{1, 0\}$ and generates the extrinsic LLR as

$$L_e^D\left\{c^i(n')\right\} := \ln \frac{P\{c^i(n') = 1 \mid \mathbf{L}\{\mathbf{c}(l)\},\ 0 \le l < T_r\}}{P\{c^i(n') = 0 \mid \mathbf{L}\{\mathbf{c}(l)\},\ 0 \le l < T_r\}} - \underbrace{\ln \frac{P\{c^i(n') = 1\}}{P\{c^i(n') = 0\}}}_{=:L\{c^i(n')\}}. \tag{2.37}$$

The extrinsic output of the MAP decoder is fedback to the MAP equalizer iteratively. Note that (2.36) and (2.37) are valid only if the *a posteriori* outputs are independent of the *a priori* inputs for the equalizer and the decoder. Assuming ideal interleaver between the equalizer and the decoder, we can apply the turbo principle and the correct ordering of the LLRs $\mathbf{L}\{\mathbf{c}(n')\} = \pi^{-1}\left[\mathbf{L}_e^E\{\mathbf{c}(n)\}\right]$ and $\mathbf{L}\{\mathbf{c}(n)\} = \pi\left[\mathbf{L}_e^D\{\mathbf{c}(n')\}\right]$, which are input to the MAP decoder and the MAP equalizer repectively. The MAP decoder also compute the *a posteriori* probabilities of the input data bit and then the data estimate $\hat{\mathbf{b}}(n') = [\hat{b}^1(n'), \hat{b}^2(n'), \cdots, \hat{b}^{k_0}(n')]$ as

$$\hat{b}^i(n') = \arg\max_{b \in \{1,0\}} P\left\{b^i(n') = b \mid \mathbf{L}\{\mathbf{c}(l)\}, 1 \le l \le T_r\right\}. \tag{2.38}$$

2.5.2 Maximum *A Posteriori* (MAP) Decoding Algorithm for Convolutional Codes [72]

In this section we introduce a recursive forward and backward algorithm for the MAP decoding that operates on a trellis description for the code, based on BCJR algorithm [29]. The MAP decoder updates the extrinsic LLR's $\mathbf{L}_e^D\{\mathbf{c}(n')\}$ of the coded bits at each iteration and generates the *a posteriori* LLR's $\mathbf{L}\{\mathbf{b}(n')\}$ of the information bits at last iteration, based on the *a priori* LLR's $\mathbf{L}\{\mathbf{c}(n')\}$ and code trellis structure.

A convolutional code with memory length l_m has finite 2^{l_m} states and its trellis structure describes the transitions between states at time n and $n+1$. In Chapter 5, we consider a recursive systematic convolutional encoder with $l_m = 2$ (constraint length of 3), a 4-state machine. Denote the input information bits at time n that cause the state transition from $\Psi(n-1) = \psi'$ to $\Psi(n) = \psi$ by $\mathbf{b}(n; \psi' \to \psi)$ and the corresponding output coded bits by $\mathbf{c}(n; \psi' \to \psi)$. The MAP decoder computes the state transition probability $P\{\Psi(n-1) = \psi', \Psi(n) = \psi\}$ using the *a priori* probabilities of the coded bits.

Assume that the encoder starts in state $\Psi(0) = \mathbf{0}$ and also ends in state $\Psi(\tau) = \mathbf{0}, \tau = T_r + l_m$, where we have at least l_m additional zero inputs following the information bit

sequence $\{\mathbf{b}(n)\}_{n=1}^{T_r}$. Then the state transition probability can be given as

$$\begin{aligned} P\{\Psi(n-1)=\psi', \Psi(n)=\psi\} &= \phi(n-1;\psi')\theta(n;\psi)P\{\mathbf{c}(n;\psi'\to\psi)\}, \\ &= \phi(n-1;\psi')\theta(n;\psi)\prod_{j=1}^{n_0} P\left\{c^j(n;\psi'\to\psi)\right\}, \end{aligned} \tag{2.39}$$

where $\phi(n;\psi)$ denotes the total probability of all path segments starting from the origin of the trellis that terminate in state ψ at time n; and $\theta(n;\psi)$ denotes the total probability of all path segments terminating at the end of the trellis that originate from state ψ at time n. The probability $\phi(n;\psi)$ and $\theta(n;\psi)$ can be computed through the following forward and backward recursions:

$$\phi(n;\psi) = \sum_{\Psi'} \phi(n-1;\psi')P\{\mathbf{c}(n;\psi'\to\psi)\}, \quad n=1,2,\cdots,\tau, \tag{2.40}$$

$$\theta(n;\psi) = \sum_{\Psi'} \theta(n+1;\psi')P\{\mathbf{c}(n+1;\psi\to\psi')\}, \quad n=\tau-1,\tau-2,\cdots,0, \tag{2.41}$$

where initial conditions are given as

$$\begin{aligned} \phi(0,\mathbf{0}) = 1,\ \phi(0,\psi) = 0 \text{ for } \psi \neq \mathbf{0}, \\ \theta(\tau,\mathbf{0}) = 1,\ \theta(\tau,\psi) = 0 \text{ for } \psi \neq \mathbf{0}, \end{aligned}$$

and Ψ' is a set of the state ψ' for all possible transition from ψ' to ψ. At each recursion, since $\phi(n,\psi)$ and $\theta(n;\psi)$ are the probabilities for ψ and time n, we can normalize as

$$\phi(n;\psi) = \frac{\phi(n;\psi)}{\sum_\Psi \phi(n;\psi)}, \qquad \theta(n;\psi) = \frac{\theta(n;\psi)}{\sum_\Psi \theta(n;\psi)},$$

where Ψ is a set of all the possible state ψ.

Given the *a priori* LLR's $\mathbf{L}\{\mathbf{c}(n')\}$ and the trellis structure of convolution code, we have the extrinsic *a posteriori* LLR for the coded bit $c^i(n')$ as

$$\begin{aligned}
L_e^D\left\{c^i(n')\right\} &= \log\frac{P\{c^i(n')=1 \mid \mathbf{L}\{\mathbf{c}(n')\}\}}{P\{c^i(n')=0 \mid \mathbf{L}\{\mathbf{c}(n')\}\}} - \log\frac{P\{c^i(n')=1\}}{P\{c^i(n')=0\}} \\
&= \log\frac{\sum_{\Psi(i,1)} P\{\Psi(n'-1)=\psi', \Psi(n')=\psi\}/P\{c^i(n')=1\}}{\sum_{\Psi(i,0)} P\{\Psi(n'-1)=\psi', \Psi(n')=\psi\}/P\{c^i(n')=0\}} \\
&= \log\frac{\sum_{\Psi(i,1)} \phi(n'-1;\psi')\theta(n';\psi)P\{\mathbf{c}(n';\psi'\to\psi)\}/P\{c^i(n')=1\}}{\sum_{\Psi(i,0)} \phi(n'-1;\psi')\theta(n';\psi)P\{\mathbf{c}(n';\psi'\to\psi)\}/P\{c^i(n')=0\}} \\
&= \log\frac{\sum_{\Psi(i,1)} \phi(n'-1;\psi')\theta(n';\psi)\prod_{j=1}^{n_0} P\{c^j(n';\psi'\to\psi)\}/P\{c^i(n')=1\}}{\sum_{\Psi(i,0)} \phi(n'-1;\psi')\theta(n';\psi)\prod_{j=1}^{n_0} P\{c^j(n';\psi'\to\psi)\}/P\{c^i(n')=0\}} \\
&= \log\frac{\sum_{\Psi(i,1)} \phi(n'-1;\psi')\theta(n';\psi)\prod_{j=1,j\neq i}^{n_0} P\{c^j(n')=c^j(\psi'\to\psi)\}}{\sum_{\Psi(i,0)} \phi(n'-1;\psi')\theta(n';\psi)\prod_{j=1,j\neq i}^{n_0} P\{c^j(n')=c^j(\psi'\to\psi)\}}
\end{aligned}$$

where $\Psi(i,b)$ is a set of all possible state pairs (ψ',ψ) for $c^i(\psi'\to\psi)=b\in\{1,0\}$.

The MAP decoder also generates the decoded bits $\hat{\mathbf{b}}(n')$ at the last iteration. We can compute the *a posteriori* LLR of the information bit by

$$L\left\{b^i(n')\right\} = \log\frac{\sum_{\mathcal{U}(i,1)} \phi(n'-1;\psi')\theta(n';\psi)\prod_{j=1}^{n_0} P\{c^j(n')=c^j(\psi'\to\psi)\}}{\sum_{\mathcal{U}(i,0)} \phi(n'-1;\psi')\theta(n';\psi)\prod_{j=1}^{n_0} P\{c^j(n')=c^j(\psi'\to\psi)\}}, \tag{2.42}$$

where $\mathcal{U}(i,b)$ is a set of all possible state pairs (ψ',ψ) for $b^i(\psi'\to\psi)=b\in\{1,0\}$. Counting on the sign of LLR, $L\{b^i(n')\}$ in (2.42), we can decide the information bit $\hat{b}^i(n')\in\{1,0\}$. [See more details in [72].]

2.6 Conclusions

In this chapter, we reviewed characteristics and representations of wireless channels. We will use the Jakes' model that follows [75] (with a correction in the appendix of [63]) as the model of the "real" channel in all the simulations in this dissertation. To describe the temporal variation of the channel for the purpose of channel estimation, we will use CE-BEM, a block-based channel model that is often more convenient than the symbol-based AR models. We also provided some background that we use for channel estimation and equalization in the following chapters. We summarized the TM training scheme in [30,70] which is useful to estiamte channel in subblock-wise tracking approaches in Chapter 3 and 4. We will employ the Kalman detector (KD) and MMSE-DFE to detect the transmitted symbols using channel estimates provided by a channel estimator. For the coded communication systems, the turbo equalization is a powerful technique that we will investigate in Chapter 5.

Chapter 3
Doubly-selective Adaptive Channel Estimation Using Exponential Basis Models and Subblock Tracking

We present a subblock-wise tracking approach to doubly-selective channel estimation, exploiting the oversampled CE-BEM for the overall channel variations, and an AR model to update the BEM coefficients. 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 that of the channel. We track the BEM coefficients via Kalman filtering, based on time-multiplexed periodically transmitted training symbols. Simulation examples demonstrate its superior performance over some existing doubly-selective channel tracking schemes.

3.1 Introduction

The first-order AR model, which has been very popular to describe a time-varying channel, is not appropriate for a fast-fading channel, since the channel estimates can only be obtained from training symbols [76]. Potential solutions lie in exploiting the detected symbols for channel tracking [6, 43, 64], where error propagation due to incorrect detections can be pronounced for fast-fading channels. Recently, BEM has been widely investigated to represent doubly-selective channels in wireless applications [8, 11, 35, 63, 70], for which, the time-varying channel taps are expressed as superpositions of time-varying basis functions in modeling Doppler effects, weighted by time-invariant coefficients. In contrast to AR models that describe temporal variation on a symbol-by-symbol update basis, a BEM depicts the evolution of the channel over a period (block) of time. Intuitively, the BEM coefficients are more convenient to track in a fast-fading environment, since they evolve much more slowly in time (based on a block-by-block update) than the real channel.

We propose a subblock-wise tracking approach for doubly-selective channels using time-multiplexed (TM) training. It exploits the CE-BEM for the overall channel variations of each (overlapping) block and a first-order AR model to describe the evolutions of the BEM coefficients. The time-varying nature of the channel can be well captured in the CE-BEM by (known) Fourier basis functions, and the slow-varying BEM coefficients are updated via Kalman filtering at each training session; during information sessions, channel estimates are generated by CE-BEM using the estimated BEM coefficients. This approach achieves better

performance in fast-fading environments, than using conventional symbol-wise AR models or block-wise BEM representations.

Channel tracking based on BEMs has been considered in [8, 26, 41] using block-by-block updating, unlike our contribution where we exploit subblock-wise updating. The distinction is as follows. One block comprises several subblocks. For parameter identifiability, one needs the number of subblocks at least as large as the number of basis functions used for channel modeling. Unlike the block-wise schemes where the receiver has to collect the whole block of data in order to generate channel estimates and perform equalization, in the proposed approaches the receiver is able to accomplish the two tasks after every subblock.

3.2 Subblock-wise Channel Estimation using CE-BEM

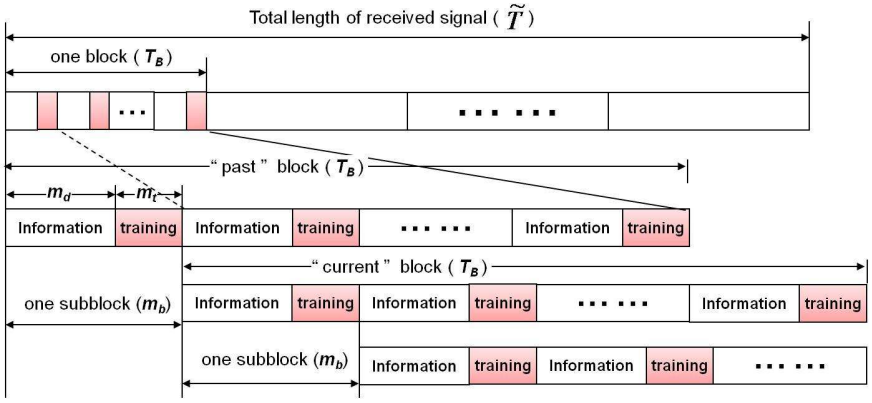


Figure 3.1: Subblock-wise channel estimation: overlapping blocks and subblocks, where one block comprises several subblocks and each subblock has an information session followed by a training session.

Suppose that we collect the received signal over a time period $\tilde{T}$ symbols. We wish to estimate the time-variant channel using a useful channel model and time-multiplexed training, and then detect the information symbols using this estimated channel. For the CE-BEM, if we choose $\tilde{T}$ as the block size, then in general by (3.4) we have a large number of basis functions to be estimated, thereby degrading the channel estimation performance. If we divide $\tilde{T}$ into blocks of (much smaller) size T_B each, and then fit the CE-BEM block by block, we have smaller Q; however, estimation of $h_q(l)$'s is now based on shorter observation size of T_B symbols which might also degrade channel estimation performance. Thus one has to strike a balance between the estimation variance and the block size. Such considerations do not apply to the symbol-wise AR channel model fitting. In the sequel, we propose a novel subblock-wise tracking approach to CE-BEM channel estimation where we update estimates of $h_q(l)$'s every subblock based on all past training sessions.

As we shall see later, use of subblock updating avoids some of the limitations of blockwise estimation. In subblock tracking there is no longer a strict definition of the block size T_B unlike blockwise estimation, allowing for a more flexible choice of parameters such as subblock size and training overhead. Intuitively, updating more often as in subblock tracking should be superior to (or at least as good as) less frequent updating as in block-wise estimation. As noted earlier in Section 2.3, the parameter identifiability requirement of $\bar{P} \geq Q$ for block-wise approach impose certain restrictions on the selection of the block and subblock sizes, which, in turn, may preclude the use oversampled CE-BEM, only the less accurate critically sampled CE-BEM may be used. Our simulation results will illustrate this fact together with the fact that subblock-wise updating encounters no such problem.

We also note that the same considerations as above prevent the use of $T_B = m_b$ in the block-wise scheme because this would lead to $\bar{P} = 1$. Alternatively, to make the block-wise scheme update the BEM parameters as frequently as the subblock-wise schemes, one may take $T_B = m_b$ but to achieve parameter identifiability, one has to insert at least Q "clusters" of zero-padded training impulses (as Section 2.3). To make this more concrete, let us consider an example where we take $m_b = 80, f_d T_s = 0.0025$ and $L = 8$. With $T_B = m_b = 80$, one obtains $Q = 3$; note that for doubly selective channels, when using CE-BEM, minimum $Q = 3$. Thus, we need 3 clusters of zero-padded training impulses with each cluster of length $2L + 1 = 17$. Therefore, one assigns 51 symbols out 80 for training leaving just 29 information symbols - a poor choice resulting in poor spectral efficiency. Such a design has 63.75% training overhead whereas our proposed subblock schemes have a training overhead of 21.25% (17 out of 80 symbols for training).

By exploiting the invariance of the coefficients of CE-BEM over each block (and hence each of the $\bar{P}$ subblocks per block of length T_B symbols), we seek subblock-wise tracking of the BEM coefficients of the doubly-selective channel. Consider two overlapping blocks (each of T_B symbols) that differ by only one subblock of size m_b: the "past" block and "current" block begin at time n_0 and $n_0 + m_b$ respectively. Since the two blocks overlap so significantly, one would expect the BEM coefficients to vary only a little from the past block to the current one. Therefore, we can track $h_q(l)$ in (3.2) subblock-by-subblock via a first-order AR model for their variations, rather than the anew with every non-overlapping block as in [70]. See Fig. 3.1 that illustrates the overlapping blocks and subblocks in our approach.

3.2.1 Subblock-wise Kalman Tracking [52]

Consider a doubly-selective (time- and frequency-selective), single-input multi-output (SIMO), finite impulse response (FIR) linear channel with N outputs. Let $\{s(n)\}$ denote a scalar information sequence that is input to the time-varying channel with discrete-time

response $\{h(n;l)\}$ (N-column vector channel response at time instance n to a unit input at time instance $n-l$). We assume $\{s(n)\}$ is independent and identically distributed (i.i.d.) with zero mean and variance σ_s^2. Then the symbol-rate noisy N-column channel output vector is given by ($n = 0, 1, \ldots$)

$$\mathbf{y}(n) = \sum_{l=0}^{L} \mathbf{h}(n;l)\, s(n-l) + \mathbf{v}(n) \tag{3.1}$$

where the N-column vector $\mathbf{v}(n)$ is zero-mean, white complex Gaussian noise, with variance $\sigma_v^2 \mathbf{I}_N$. We assume that $\{\mathbf{h}(n;l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [60].

The time-varying channel with discrete-time response $\{\mathbf{h}(n;l)\}$ is represented by CE-BEM as ($n = (i-1)T_B, (i-1)T_B + 1, \cdots, iT_B - 1$ for i-th block and $l = 0, 1, \cdots, L$)

$$\mathbf{h}(n;l) = \sum_{q=1}^{Q} \mathbf{h}_q(l) e^{j\omega_q n}, \tag{3.2}$$

where $\mathbf{h}_q(l)$ is the N-column time-invariant BEM coefficient vector and one chooses (Λ is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{3.3}$$

$$Q \geq 2\lceil f_d T T_s \rceil + 1, \tag{3.4}$$

$$\omega_q := \frac{2\pi}{T}\left[q - (Q+1)/2\right], \quad q = 1, 2, \cdots, Q \tag{3.5}$$

$$L := \lfloor \tau_d / T_s \rfloor, \tag{3.6}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration.

Let $h_{r,q}(l)$ denote the r-th component of the N column vector $\mathbf{h}_q(l)$. Stack the BEM coefficients in (3.2) into vectors as

$$\mathbf{h}_r^{(l)} := \begin{bmatrix} h_{r,1}(l) & h_{r,2}(l) & \cdots & h_{r,Q}(l) \end{bmatrix}^T \tag{3.7}$$

$$\mathbf{h}_r := \begin{bmatrix} \mathbf{h}_r^{(0)\,T} & \mathbf{h}_r^{(1)\,T} & \cdots & \mathbf{h}_r^{(L)\,T} \end{bmatrix}^T \tag{3.8}$$

$$\mathbf{h} := \begin{bmatrix} \mathbf{h}_1^T & \mathbf{h}_2^T & \cdots & \mathbf{h}_N^T \end{bmatrix}^T \tag{3.9}$$

of size $Q, M := Q(L+1)$ and $NQ(L+1)$ respectively. The coefficient vectors in (3.7), (3.8) and (3.9) of the p-th overlapping block ($p = 0, 1, \cdots$) will be denoted by $\mathbf{h}_r^{(l)}(p)$, $\mathbf{h}_r(p)$, and

$\mathbf{h}(p)$ respectively. Note that the p-th block and the following $(p+1)$-st block differ by just one subblock. Since a fading channel can follow well a Markov model [16], we further assume that the BEM coefficients over each overlapping block are also Markovian. Although one could fit a general AR model of (high) order P (as in Section 2.2.1), here we seek a "simple" formulation given by the first-order AR model, i.e.,

$$\mathbf{h}(p) = \mathbf{A}_1 \mathbf{h}(p-1) + \mathbf{w}(p), \tag{3.10}$$

where we assume $\mathbf{A}_1 = \alpha \mathbf{I}_{NQ(L+1)}$ is the AR coefficient matrix (implying that all taps have the same Doppler spectrum), and the driving noise vector $\mathbf{w}(p)$ is zero-mean white complex Gaussian with variance $\sigma_w^2 \mathbf{I}_{NQ(L+1)}$ and statistically independent of $\mathbf{h}(p-1)$. We provide a details about this simple AR(1) model and choice of α in Section 3.4. Assuming the channel is wide-sense stationary (WSS) and coefficients $h_{r,q}(l)$'s are independent, we have

$$\sigma_w^2 = \sigma_h^2 (1 - |\alpha|^2)/Q \tag{3.11}$$

where $\sigma_h^2 := E\{h_r(n;l)\, h_r^*(n;l)\}$.

Under this formulation, we do not need a "strict" definition of the block size T_B — although the channel is still represented by (3.2) for arbitrary time n, the BEM coefficients $h_{r,q}(l)$'s are updated every subblock based on the training symbols. A key parameter now is the CE-BEM period T, not the block size T_B. Later we use (3.2) for all times n, not just the particular block of size T_B symbols, by allowing the coefficients $h_{r,q}(l)$'s to change with time (subblock-wise).

Based on CE-BEM given by (3.2), define

$$\begin{aligned} \boldsymbol{\mathcal{E}}(n) &:= \left[e^{-j\omega_1 n} \quad e^{-j\omega_2 n} \quad \cdots \quad e^{-j\omega_Q n} \right]^T, \\ \mathbf{s}(n) &:= \left[s(n) \quad s(n-1) \quad \cdots \quad s(n-L) \right]^T, \end{aligned}$$

and then the received signal can be written as

$$\mathbf{y}(n) = \left[\mathbf{s}(n) \otimes \mathbf{I}_N \right]^T \left[\mathbf{I}_{N(L+1)} \otimes \boldsymbol{\mathcal{E}}(n) \right]^H \mathbf{h}(p) + \mathbf{v}(n) \tag{3.12}$$

when the p-th subblock is being received at time n, by (3.1), (3.2)-(3.6) and (3.7)-(3.9). Treating (3.10) and (3.12) as the state and the measurement equations respectively, Kalman filtering can be applied to track the BEM coefficient vector $\mathbf{h}(p)$ for each subblock.

We will employ the time-multiplexed (TM) training scheme in Section 2.3 [30] where each subblock (of equal length m_b symbols) consist of a data session (of length m_d symbols)

and a succeeding training session (of length $m_t = 2L+1$ symbols). Then the received signal at r-th output at time n_p+l with $l = 0, 1, \cdots, L$ is given by (assuming timing synchronization)

$$y_r(n_p + l) = \gamma h_r(n_p + l; l) + v_r(n_p + l). \tag{3.13}$$

for $r = 1, 2, \cdots, N$. Using CE-BEM (3.2) in these $y_r(n_p + l)$'s, one can uniquely solve for $h_{r,q}(l)$'s via a least-square approach. Then the channel estimates are given by the CE-BEM (3.2) using the estimated BEM coefficients. Using CE-BEM in (3.2) and (3.7)-(3.9), we can simplify (3.12) at time $n_p + l$ as ($p = 0, 1, \cdots$ and $l = 0, 1, \cdots, L$)

$$y_r(n_p + l) = \gamma \boldsymbol{\mathcal{E}}^H(n_p + l)\, \mathbf{h}_r^{(l)}(p) + v_r(n_p + l) \tag{3.14}$$

Note that each (sub-)channel at r-th output can be tracked using r-th received signal in (3.14) respectively without interference. We intend to use only training sessions for subblock-wise channel tracking regardless of data sessions. Further define

$$\begin{aligned}
\mathbf{y}_r(p) &:= \begin{bmatrix} y_r(n_p) & y_r(n_p+1) & \cdots & y_r(n_p+L) \end{bmatrix}^T, \\
\mathbf{v}_r(p) &:= \begin{bmatrix} v_r(n_p) & v_r(n_p+1) & \cdots & v_r(n_p+L) \end{bmatrix}^T,
\end{aligned}$$

and

$$\boldsymbol{\Psi}(p) := \begin{bmatrix} \boldsymbol{\mathcal{E}}(n_p) & & & \\ & \boldsymbol{\mathcal{E}}(n_p+1) & & \\ & & \ddots & \\ & & & \boldsymbol{\mathcal{E}}(n_p+L) \end{bmatrix}^H.$$

Then by (3.14),

$$\mathbf{y}_r(p) = \gamma \boldsymbol{\Psi}(p)\, \mathbf{h}_r(p) + \mathbf{v}_r(p), \tag{3.15}$$

which gives us a formulation to estimate the BEM coefficients using training symbols (sessions).

Thus we have obtained a linear discrete-time system represented by (3.10) and (3.15), with which Kalman filtering can be applied to track the BEM coefficient vector $\mathbf{h}_r(p)$ for each subblock. Kalman tracking in the training mode is initialized with

$$\hat{\mathbf{h}}_r(-1 \mid -1) = \mathbf{0}_M \text{ and } \mathbf{R}_{h_r}(-1 \mid -1) = \sigma_w^2 \mathbf{I}_M,$$

where $\hat{\mathbf{h}}_r(p \mid m)$ denotes the estimate of $\mathbf{h}_r(p)$ given the observations $\{\mathbf{y}_r(n)\}_{n=0}^{m}$, and $\mathbf{R}_{h_r}(p \mid m)$ denotes the error covariance matrix of $\hat{\mathbf{h}}_r(p \mid m)$, defined as being equal to

$$E\{[\hat{\mathbf{h}}_r(p \mid m) - \mathbf{h}_r(p)][\hat{\mathbf{h}}_r(p \mid m) - \mathbf{h}_r(p)]^H\}.$$

Kalman recursive filtering (for $p = 0, 1, \cdots$) one by one (for each r-th output) via the following steps [32] (we have omitted the subscript r in the following steps):

1. Time update:

$$\begin{aligned}\hat{\mathbf{h}}(p \mid p-1) &= \alpha \hat{\mathbf{h}}(p-1 \mid p-1),\\ \mathbf{R}_h(p \mid p-1) &= |\alpha|^2 \mathbf{R}_h(p-1 \mid p-1) + \sigma_w^2 \mathbf{I}_M;\end{aligned}$$

2. Kalman gain:

$$\begin{aligned}\mathbf{R}_\eta(p) &= |\gamma|^2 \mathbf{\Psi}(p) \mathbf{R}_h(p \mid p-1) \mathbf{\Psi}^H(p) + \sigma_v^2 \mathbf{I}_{L+1},\\ \mathbf{K}(p) &= \gamma \mathbf{R}_h(p \mid p-1) \mathbf{\Psi}^H(p) \mathbf{R}_\eta^{-1}(p);\end{aligned}$$

3. Measurement update:

$$\begin{aligned}\hat{\mathbf{h}}(p \mid p) &= \hat{\mathbf{h}}(p \mid p-1) + \mathbf{K}(p)\left[\mathbf{y}(p) - \gamma \mathbf{\Psi}(p) \hat{\mathbf{h}}(p \mid p-1)\right],\\ \mathbf{R}_h(p \mid p) &= [\mathbf{I}_M - \gamma \mathbf{K}(p) \mathbf{\Psi}(p)] \mathbf{R}_h(p \mid p-1).\end{aligned}$$

After Kalman filtering for every p, we generate the channel for the p-th subblock by the estimate $\hat{\mathbf{h}}(p \mid p)$ via the CE-BEM (3.2) as

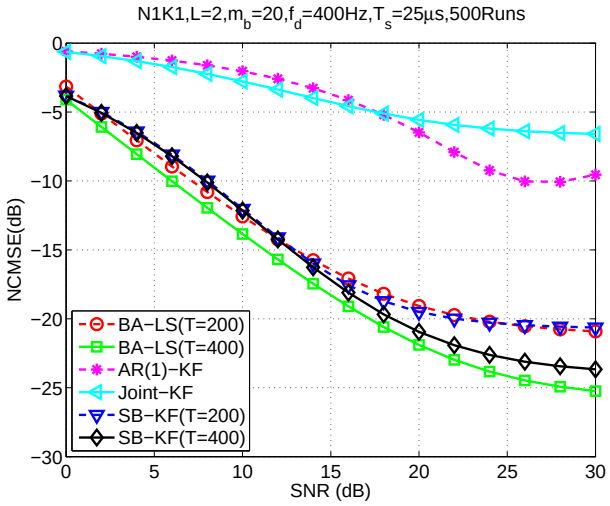
$$\hat{h}_r(n; l) = \boldsymbol{\mathcal{E}}^H(n) \hat{\mathbf{h}}_r^{(l)}(p \mid p) \tag{3.16}$$

for $n = pm_b, pm_b + 1, \cdots, (p+1)m_b - 1$. The definition of $\hat{\mathbf{h}}_r^{(l)}(p \mid p)$ is similar to (3.7).

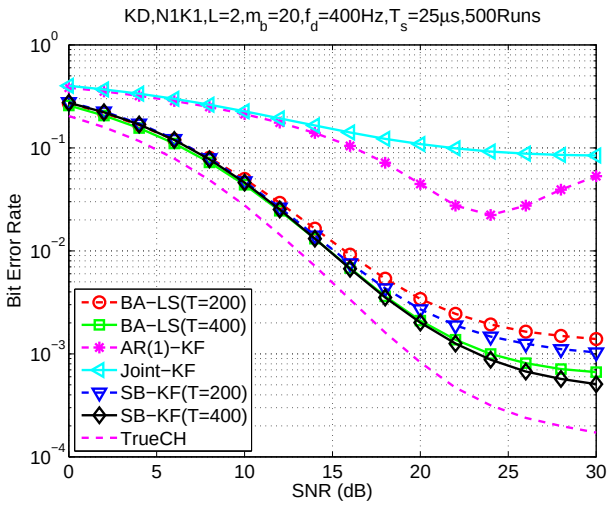
3.2.2 Simulation Examples

Example 1

A random time- and frequency-selective Rayleigh fading channel is considered. We assume $\mathbf{h}(n; l)$ are zero-mean, complex Gaussian, and spatially white. We take $L = 2$ (3 taps) in (3.1), and $\sigma_h^2 = E\{h_r(n; l) h_r^*(n; l)\} = 1/(L+1)$. For different l's, $\mathbf{h}(n; l)$'s are



(a) NCMSE vs SNR



(b) BER vs SNR, with Kalman detector of equalization delay $d = 5$

Figure 3.2: Subblock-wise Kalman channel estimation: performance comparison for SNR's under $N = K = 1, f_d T_s = 0.01, m_b = 20$.

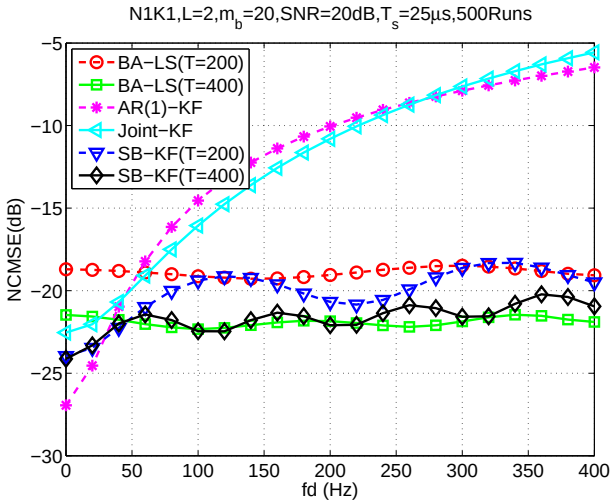
mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75] (with a correction in the appendix of [63]).

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 0 to 400Hz, or the normalized Doppler spread $f_d T_s$ from 0 to 0.01 (corresponding to a maximum mobile velocity from 0 to 216km / h). 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.

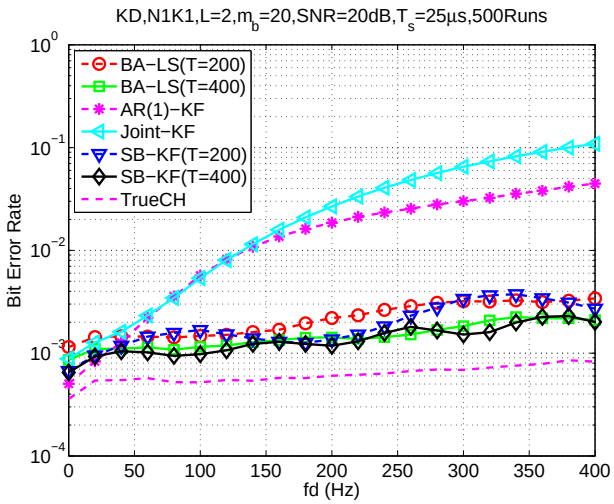
The TM training scheme of [70], which is optimal for channels satisfying the critically sampled CE-BEM representation, is adopted, where each subblock of equal length m_b symbols consists of an information session of m_d symbols and a succeeding training session of m_t symbols ($m_b = m_d + m_t$). We assume that each information symbol has unit power, while at every training session given by (2.22), we set $\gamma = \sqrt{2L+1}$ so that the average power per symbol at training sessions is equal to that of information sessions. In the simulations, we set $\gamma = \sqrt{5}$ for $m_b = 20$ ($m_d = 15$ and $m_t = 5$) or $m_b = 40$ ($m_d = 35$ and $m_t = 5$).

We compared the following five schemes:

1. The block-adaptive channel estimation in [70], where the transmitted symbols are segmented into consecutive blocks of T_B symbols each. Every block consists of $\bar{P}$ ($\geq Q$) subblocks as introduced in Section 2.3. For each non-overlapping block, we estimate the BEM coefficients anew via a least-squares approach, and obtain the channel estimates



(a) NCMSE vs f_d

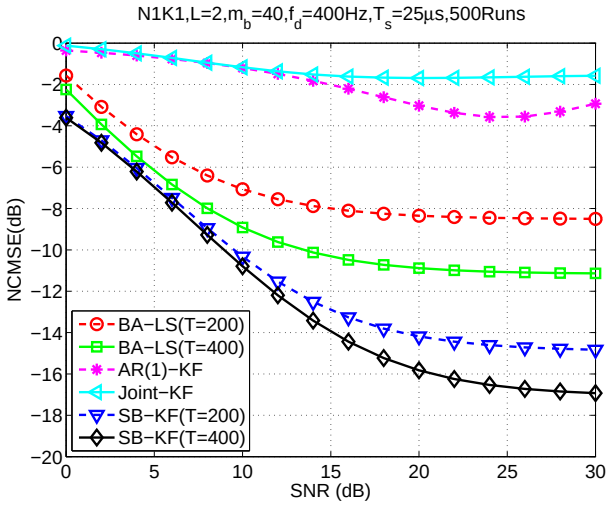


(b) BER vs f_d, with Kalman detector of equalization delay $d = 5$

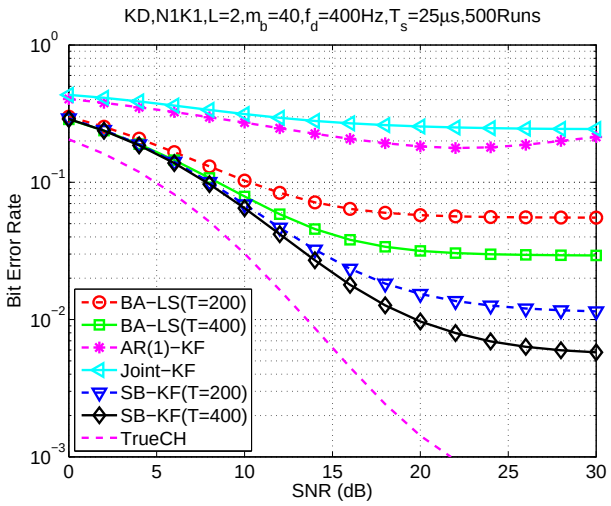
Figure 3.3: Subblock-wise Kalman channel estimation: performance comparison for Doppler spread under $N = K = 1, \text{SNR} = 20\,\text{dB}, m_b = 20$.

over this block by the CE-BEM. We consider two models corresponding to $T = 200$ and 400 respectively, so that $Q = 5$ and 9 by (3.4). We use an oversampled CE-BEM with $T_B = T/2$ for $m_b = 20$ in order to suppress spectral leakage; whereas for $m_b = 40$, since an oversampled CE-BEM is not possible to satisfy $\bar{P} \geq Q$, we take $T_B = T$. [For parameter identifiability, one needs $\bar{P} \geq Q$, i.e., $T/(\Lambda m_b) \geq 2\lceil f_d T T_s \rceil + 1$. With $T_s = 25\mu s$, $f_d = 400Hz$ and $m_b = 40$, there exists no integer with $\Lambda = 2$ for which we can satisfy $T/(\Lambda m_b) \geq 2\lceil f_d T T_s \rceil + 1$. This implies that for blockwise updating with $m_b = 40$, we cannot use oversampled CE-BEM, only the less accurate critically sampled CE-BEM.] Considering the oversampled CE-BEM basis functions are not orthogonal, we apply a regularized least-square approach with a regularization parameter $\beta = 1$ to estimate of BEM coefficients, and then obtain the channel estimates over this block via the CE-BEM. Furthermore, the results were worse in the oversampled CE-BEM case when no regularization was used, caused by possible ill-conditioning of certain matrices when the basis functions are not orthogonal. For i-th (non-overlapping) block which has $\bar{P}$ subblocks, the estimate of $M \times 1$ BEM coefficient vector $\hat{\mathbf{h}}(i)$ is given by ($M = Q(L+1)$ is the number of unknown BEM coefficients)

$$\hat{\mathbf{h}}(i) = \left[\gamma^2 \mathbf{\Psi}_B^H(i)\mathbf{\Psi}_B(i) + \beta \mathbf{I}_M\right]^{-1} \gamma \mathbf{\Psi}_B^H(i)\mathbf{y}_B(i), \tag{3.17}$$



(a) NCMSE vs SNR



(b) BER vs SNR, with Kalman detector of equalization delay $d = 5$

Figure 3.4: Subblock-wise Kalman channel estimation: performance comparison for SNR's under $N = K = 1, f_d T_s = 0.01, m_b = 40$.

where

$$\mathbf{y}_B(i) = \gamma \mathbf{\Psi}_B(i)\mathbf{h}(i) + \mathbf{v}_B(i), \tag{3.18}$$

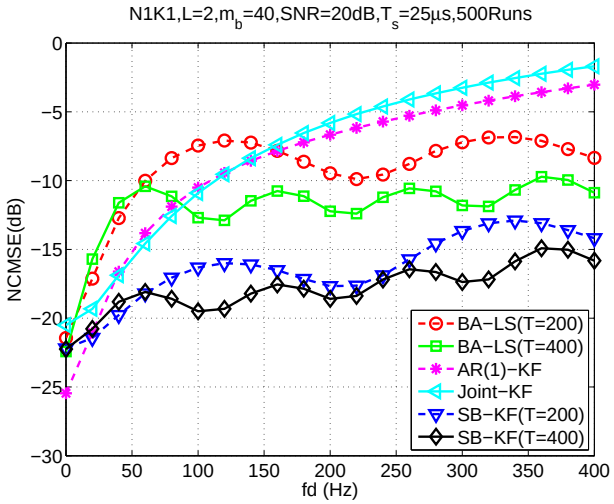
defining

$$\mathbf{y}_B(i) = \left[\mathbf{y}_s\left(\bar{P}\right) \quad \mathbf{y}_s\left(\bar{P}-1\right) \quad \cdots \quad \mathbf{y}_s\left(1\right)\right]^T,$$
$$\mathbf{\Psi}_B(i) = \left[\mathbf{\Psi}\left(\bar{P}\right) \quad \mathbf{\Psi}\left(\bar{P}-1\right) \quad \cdots \quad \mathbf{\Psi}\left(1\right)\right]^T,$$

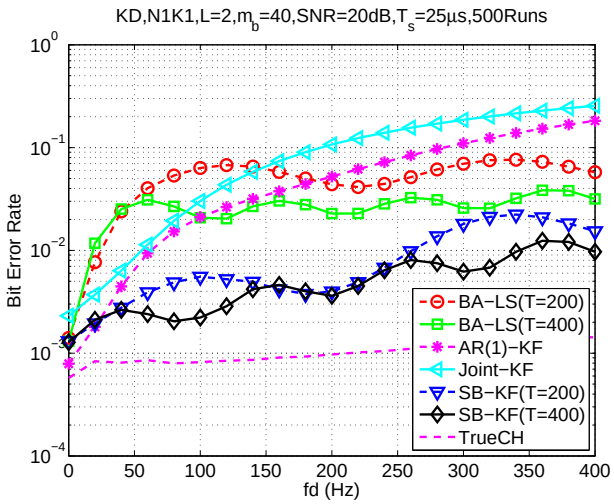
and then obtain the channel estimates over this block by the CE-BEM. In the figures, this scheme is denoted by "BA-LS".

2. The channel tracking scheme in [76] using the symbol-wise first-order AR model and Kalman filtering. Channel tracking is performed at training sessions only. For the information sessions, the receiver updates the channel via $\hat{h}(n;l) = \alpha_c \hat{h}(n-1;l)$. We assume that only the upper-bound of the Doppler spread is known. Then for Jakes' model, $\alpha_c = J_0(2\pi f_d T_s) = 0.999$ for $f_d T_s = 0.01$, where $J_0(\cdot)$ denotes the zero-th Bessel function of the first kind. This scheme is denoted by "AR(1)-KF".

3. We also compared the approach of joint channel estimation and data detection via extended Kalman filtering (EKF) in [68], where the channel taps are also described by AR(1) model. For fairness, the turbo equalization (and channel coding) procedure in [68] is omitted. This scheme is denoted by "Joint-KF".



(a) NCMSE vs f_d



(b) BER vs f_d, with Kalman detector of equalization delay $d = 5$

Figure 3.5: Subblock-wise Kalman channel estimation: performance comparison for Doppler spread under $N = K = 1, \text{SNR} = 20\,\text{dB}, m_b = 40$.

4. Our subblock-wise tracking using CE-BEM, which is denoted by "SB-KF", takes $T = 200$ and 400 for two different settings of CE-BEM, and $Q = 5$ and 9 correspondingly. We also take AR coefficient $\alpha = 0.99$ for $T = 200$ and $\alpha = 0.9944$ for $T = 400$ when $m_b = 20$, while $\alpha = 0.94$ for $T = 200$ and $\alpha = 0.9667$ for $T = 400$ when $m_b = 40$(these values were selected empirically via simulations).

5. Perfect channel estimates are available at the receiver, which is denoted by "TrueCH".

We evaluate the performances of various schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

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

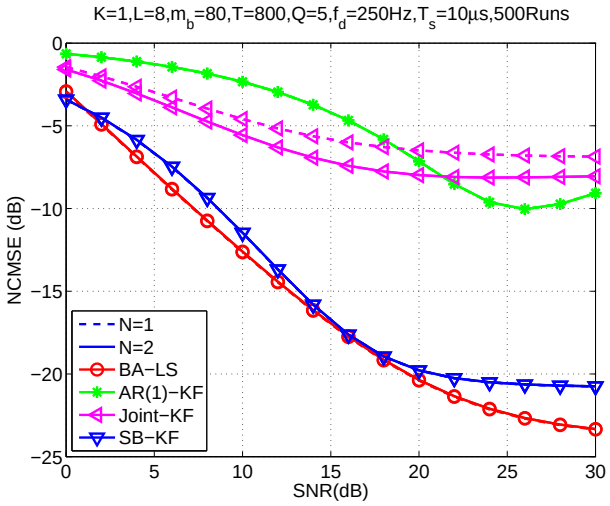
where $\mathbf{h}^{(i)}(n;l)$ is the true channel and $\hat{\mathbf{h}}^{(i)}(n;l)$ is the estimated channel at the i-th Monte Carlo run, among total M_r runs, and T_N is the total observation length in symbols. The Kalman detector in Section 2.4.1 with delay $d = 5$ is used using the channel estimates obtained by each scheme. In each run, a symbol sequence of length $T_N = 4000$ is modulated by QPSK and the first 200 symbols are discarded in evaluations. All the simulation results are based on 500 runs.

In Fig. 3.2, the performances of the five schemes with $m_b = 20$ ($m_d = 15$ and $m_t = 5$) are compared for different SNR's under normalized Doppler spread $f_d T_s = 0.01$. The

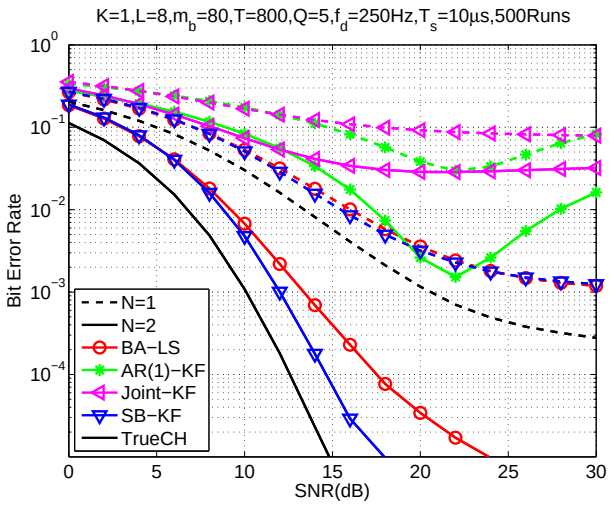
block-adaptive scheme with the regularized LS approach (BA-LS) and subblock-wise Kalman tracking (SB-KF), based on CE-BEM, has better performance than the symbol-wise schemes, AR(1)-based Kalman filtering (AR(1)-KF) and joint channel estimation and data detection (Joint-KF). For CE-BEM-based schemes, more basis functions ($T = 400, Q = 9$) in (3.2) translate into better performance although the larger block size ($T_B = 200$) may increase estimation variance.

In Fig. 3.3, the performances are compared for different Doppler spread f_d's under fixed SNR = 20 dB. It is clear that the symbol-wise AR(1)-KF and Joint-KF deteriorate sharply due to the modeling inadequacies of the symbol-wise AR representation. In contrast, the NCMSE's and BER's of the CE-BEM-based BA-LS and SB-KF vary more gradually with increasing Doppler spreads, because the modeling error of the CE-BEM is much smaller. Noting that the CE-BEM used in the schemes is based solely on the maximum Doppler shift $f_d T_s = 0.01$, it follows that the performances of the CE-BEM-based schemes are not sensitive to the "actual" Doppler spread if it is no larger than the assumed maximum value. That is, we do not have to know the exact Doppler spread of the channel — an upper bound of it suffices. Since more unknown parameters are involved in the CE-BEM-based schemes and hence result in higher estimation variance, these schemes are slightly inferior to the symbol-wise AR-based schemes for small f_d's; but as f_d increases, the CE-BEM-based schemes outperform the AR-based schemes.

In Fig. 3.4, we have longer information sessions to achieve a more spectrally-efficient transmission, where the subblock size $m_b = 40$ ($m_d = 35$ and $m_t = 5$) so that only 12.5% of the transmitted symbols are dedicated to training; for the small subblock with $m_b = 20$, 25% of the transmitted symbols are dedicated to training. Similarly, symbol-wise AR(1)-KF and Joint-KF cannot capture the fast-varying channel even with high SNR's. For the BA-LS, we cannot use the oversampled CE-BEM with this longer subblock and hence results in worse performance in NCMSE's and BER's due to the spectral leakage. [It is when we use the oversampled CE-BEM that block-adaptive LS has similar performance as our proposed subblock-wise Kalman tracking.] However, our proposed subblock-wise Kalman tracking is insensitive to this problem and maintain satisfactory performances. In Fig. 3.5, four schemes with $m_b = 40$ are compared over different Doppler spread f_d's. Definitely, block-adaptive scheme based on critically-sampled CE-BEM deteriorates except for small (nearby zero) Doppler spread; the spectral leakage is also small with small f_d. Our subblock-wise Kalman tracking, where it updates the channel estimates subblock-by-subblock using CE-BEM, outperforms other existing schemes.



(a) NCMSE vs SNR



(b) BER vs SNR, with Kalman detector of equalization delay $d = 10$

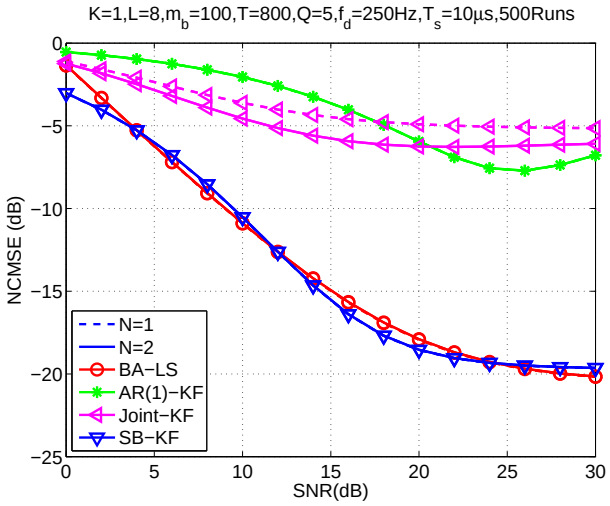
Figure 3.6: Subblock-wise Kalman channel estimation: performance comparison for SNR's under $K = 1, f_d T_s = 0.0025, m_b = 80$.

Example 2

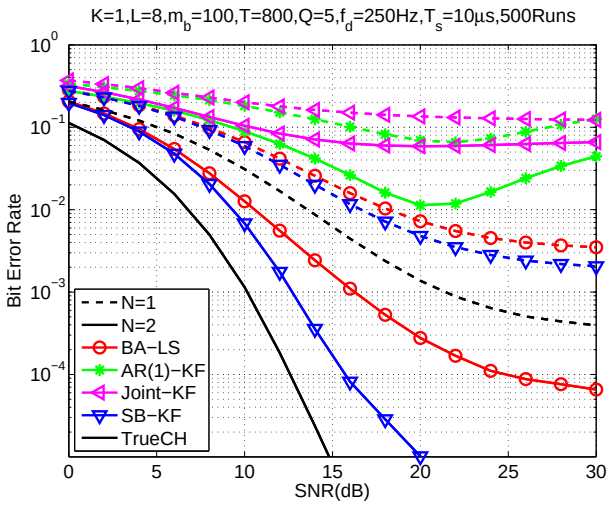
The random Rayleigh fading channel in this example is as in Example 1, except that we change the channel length L from 2 to 8 leading to 9 taps, and the power delay profile is exponential satisfying $E\{|h_r(n:l)|^2\} = \sigma_h^2(l) = ae^{-l/L}$ where the constant a is picked to satisfy $\sum_{l=0}^{L} \sigma_h^2(l) = 1$. We consider a communication system with carrier frequency of 2GHz, data rate of 0.1MBd (mega-Bauds), therefore $T_s = 10\,\mu$s, and a varying Doppler spread $f_d = 250\,$Hz, or the normalized Doppler spread $f_d T_s = 0.0025$ (corresponding to a maximum mobile velocity of 135km / h). The training session is described by (2.22) of length $m_t = 2L + 1 = 17$ symbols with $\gamma = \sqrt{2L+1}$ so that the average symbol power of training sessions is equal to that of information sessions. We select the period of the CE-BEM $T = 800$ symbols, and hence $Q = 5$ by (3.4) and a block size $T_B = 400$ for oversampled CE-BEM.

Four estimation and tracking schemes are compared: block-adaptive channel estimation in [70] using the regularized LS approach ("BA-LS"), channel tracking scheme in [76] using a first-order AR model ("AR(1)-KF"), joint channel estimation and data detection via EKF in [68] ("Joint-KF") and our subblock-wise Kalman tracking ("SB-KF"). The BER's are evaluated by employing the Kalman detector with delay $d = 10$, using the channel estimates obtained by each scheme.

In Figs. 3.6, the performances of the four schemes with the subblock size $m_b = 80$ are compared for different SNR's. For the subblock-wise Kalman tracking, we picked the AR



(a) NCMSE vs SNR



(b) BER vs SNR, with Kalman detector of equalization delay $d = 10$

Figure 3.7: Subblock-wise Kalman channel estimation: performance comparison for SNR's under $K = 1, f_d T_s = 0.0025, m_b = 100$.

coefficient $\alpha = 0.993$. Based on CE-BEM, block-adaptive scheme [70] with regularized LS approach and subblock-wise Kalman filtering has superior performance to the symbol-wise AR-based AR(1)-KF [76] and Joint-KF [68]. It is sure that CE-BEM is more effective to estimate channel than AR model, especially in fast-varying channel. We also consider single-input multiple-output system with 2 receivers to distinguish the performances of BA-LS and SB-KF. Due to the assumption that the channel is spatially uncorrelated for different outputs, the NCMSE curves for $N = 1$ and 2 coincide with each other but Joint-KF estimating channel and data symbols jointly. It is worth pointing out that superior average channel NCMSE does not necessarily translate into superior BER since a monotone relationship between average NCMSE and BER does not necessarily exist. The BER's with multi-receivers show the proposed subblock-wise Kalman filtering outperforms block-adaptive LS approach. [One can replace Kalman detector with MMSE-DFE in Section 2.4.2 that makes more distinguishable BER performances. We will use DFE later.]

In Figs. 3.7, we have a little longer information sessions, where we set the subblock size $m_b = 100$ and the AR coefficient $\alpha = 0.992$ for the subblock-wise Kalman tracking. Admitting the worse performance with the less training overhead (21.3% for $m_b = 80$ and 17% for $m_b = 100$), the deteriorations of BA-LS in NCMSE and BER are severe due to the identifiability problem. For the block-adaptive schemes, we consider the oversampling ($\Lambda = 2$), although parameter identifiability does not hold for oversampling since the number of subblocks ($\bar{P} = T_B/m_b = (T/\Lambda)/m_b = 4$) is less than one of unknown parameters ($Q = 5$).

[One can use the critically-sampling ($\Lambda = 1$) to satisfy the parameter identifiability. However, our experiments for this example show the BA-LS solution with critically-sampled CE-BEM is even worse than with oversampled one.] Our proposed subblock-wise Kalman tracking updates the channel estimates subblock-by-subblock and hence the block size T_B is not a key parameter as in BA-LS. As a result shown in Figs, the subblock-wise approach has more advantages than block-wise or symbol-wise ones.

3.3 Subblock-wise MIMO Channel Estimation using CE-BEM

3.3.1 Subblock-wise Kalman Tracking for MIMO Channel

Consider a doubly-selective multi-input multi-output (MIMO), finite impulse response (FIR) linear channel with K inputs and N outputs. Let $\{s_k(n)\}$ denote k-th user's information sequence that is input to the time-varying channel with discrete-time response $\{\mathbf{h}_k(n;l)\}$ (channel response for the k-th user at time instance n to a unit input at time instance $n-l$). We assume $\{s_k(n)\}$ is mutually independent and identically distributed (i.i.d.) with zero mean and variance $E\{s_k(n)\, s_k^*(n)\} = \sigma_{s_k}^2 = \sigma_s^2$ for $k = 1, 2, \cdots, K$. Then the symbol-rate noisy N-column channel output vector is given by ($n = 0, 1, \ldots$)

$$\mathbf{y}(n) = \sum_{k=1}^{K} \sum_{l=0}^{L} \mathbf{h}_k(n;l)\, s_k(n-l) + \mathbf{v}(n) \tag{3.19}$$

where the N-column vector $\mathbf{v}(n)$ is zero-mean, white, uncorrelated with $s_k(n)$, complex Gaussian noise, with the autocorrelation $E\{\mathbf{v}(n+\tau)\,\mathbf{v}^H(n)\} = \sigma_v^2 \mathbf{I}_N \delta(\tau)$. We assume that $\{\mathbf{h}_k(n;l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [60], independent for different k's. Define

$$\begin{aligned} \mathbf{s}(n) &:= \begin{bmatrix} s_1(n) & s_2(n) & \cdots & s_K(n) \end{bmatrix}^T \\ \mathbf{h}(n;l) &:= \begin{bmatrix} \mathbf{h}_1(n;l) & \mathbf{h}_2(n;l) & \cdots & \mathbf{h}_K(n;l) \end{bmatrix}. \end{aligned}$$

and we may rewrite (3.19) as

$$\mathbf{y}(n) = \sum_{l=0}^{L} \mathbf{h}(n;l)\, \mathbf{s}(n-l) + \mathbf{v}(n). \tag{3.20}$$

In CE-BEM [11, 14, 70], over the i-th block consisting of an observation window of T_B symbols, the channel is represented as ($n = (i-1)T_B, (i-1)T_B + 1, \cdots, iT_B - 1$ and

$l = 0, 1, \cdots, L$)

$$\mathbf{h}_k(n; l) = \sum_{q=1}^{Q} \mathbf{h}_{k,q}(l) e^{j\omega_q n}, \tag{3.21}$$

where $\mathbf{h}_{k,q}(l)$ is the N-column time-invariant BEM coefficient vector for k-th user and one chooses (Λ is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{3.22}$$

$$Q \geq 2\lceil f_d T T_s \rceil + 1, \tag{3.23}$$

$$\omega_q := \frac{2\pi}{T}\left[q - (Q+1)/2\right], \quad q = 1, 2, \cdots, Q \tag{3.24}$$

$$L := \lfloor \tau_d / T_s \rfloor, \tag{3.25}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration.

Now extend the proposed subblock-wise approach in Section 3.2 to MIMO systems. Let $h_{rk,q}(l)$ denote the $r-$th component of the column $\mathbf{h}_{k,q}(l)$. Stack the BEM coefficients in (3.21) into vectors as

$$\mathbf{h}_{rk}^{(l)} := \begin{bmatrix} h_{rk,1}(l) & h_{rk,2}(l) & \cdots & h_{rk,Q}(l) \end{bmatrix}^T \tag{3.26}$$

$$\mathbf{h}_{rk} := \begin{bmatrix} \mathbf{h}_{rk}^{(0)\,T} & \mathbf{h}_{rk}^{(1)\,T} & \cdots & \mathbf{h}_{rk}^{(L)\,T} \end{bmatrix}^T \tag{3.27}$$

$$\mathbf{h} := \begin{bmatrix} \mathbf{h}_{11}^T & \cdots & \mathbf{h}_{N1}^T & \cdots & \mathbf{h}_{1K}^T & \cdots & \mathbf{h}_{NK}^T \end{bmatrix}^T \tag{3.28}$$

of size Q, $M := Q(L+1)$ and $NKQ(L+1)$, respectively. The coefficient vectors in (3.26), (3.27) and (3.28) of the p-th subblock will be denoted by $\mathbf{h}_{rk}^{(l)}(p)$, $\mathbf{h}_{rk}(p)$, and $\mathbf{h}(p)$. Since a fading channel can follow well a Markov model [16], we further assume that the BEM coefficients over each overlapping block are also Markovian. A simplified formulation is given by the first-order AR model, i.e.,

$$\mathbf{h}(p) = \mathbf{A}_1 \mathbf{h}(p-1) + \mathbf{w}(p), \tag{3.29}$$

where we assume $\mathbf{A}_1 = \alpha \mathbf{I}_{NKQ(L+1)}$ is the AR coefficient matrix, and the driving noise vector $\mathbf{w}(p)$ is zero-mean white complex Gaussian with variance $\sigma_w^2 \mathbf{I}_{NKQ(L+1)}$ and statistically independent of $\mathbf{h}(p-1)$. Assuming the channel is wide-sense stationary (WSS) and coefficients $h_{rk,q}(l)$'s are independent, we have

$$\sigma_w^2 = \sigma_h^2 (1 - |\alpha|^2)/Q \tag{3.30}$$

where $\sigma_h^2 := E\left\{h_{rk}\left(n;l\right)h_{rk}^*\left(n;l\right)\right\}$.

Based on CE-BEM given by (3.21), define (where $\mathbf{s}\left(n\right)$ is K-column vector)

$$\boldsymbol{\mathcal{E}}\left(n\right) := \left[e^{-j\omega_1 n} \quad e^{-j\omega_2 n} \quad \cdots \quad e^{-j\omega_Q n}\right]^T, \tag{3.31}$$

$$\mathbf{S}\left(n\right) := \left[\mathbf{s}^T\left(n\right) \quad \mathbf{s}^T\left(n-1\right) \quad \cdots \quad \mathbf{s}^T\left(n-L\right)\right]^T, \tag{3.32}$$

and then the received signal can be written as

$$\mathbf{y}\left(n\right) = \left[\mathbf{S}\left(n\right)\otimes\mathbf{I}_N\right]^T\left[\mathbf{I}_{NK(L+1)}\otimes\boldsymbol{\mathcal{E}}\left(n\right)\right]^H\mathbf{h}\left(p\right)+\mathbf{v}\left(n\right) \tag{3.33}$$

when the p-th subblock is being received at time n, by (3.20), (3.21)-(3.25) and (3.26)-(3.28). Treating (3.29) and (3.33) as the state and the measurement equations respectively, Kalman filtering can be applied to track the BEM coefficient vector $\mathbf{h}\left(p\right)$ for each subblock.

We will employ the TM training scheme proposed in Section 2.3 [30]. Then by design, a given user's training does not affect that of any other user during training session. Hence the received signal between r-th output and k-th input can be written as (assuming timing synchronization)

$$y_r\left(n_{k,p}+l\right) = \gamma h_{rk}\left(n_{k,p}+l;l\right)+v_r\left(n_{k,p}+l\right) \tag{3.34}$$

for $l=0,1,\cdots,L$, $k=1,2,\cdots,K$ and $r=1,2,\cdots,N$. Using (3.21) in these $y_r\left(n_{k,p}+l\right)$'s, one can uniquely solve for $h_{r_{k,q}}(l)$'s via a least-square (LS) approach. Then the channel estimates are given by the CE-BEM (3.21) using the estimated BEM coefficients. Therefore, every element of $N\times K$ channel matrix,

$$\mathbf{h}\left(n;l\right) = \begin{bmatrix} h_{11}\left(n;l\right) & h_{12}\left(n;l\right) & \cdots & h_{1K}\left(n;l\right) \\ \vdots & \vdots & h_{rk}\left(n;l\right) & \vdots \\ h_{N1}\left(n;l\right) & h_{N2}\left(n;l\right) & \cdots & h_{NK}\left(n;l\right) \end{bmatrix},$$

can be tracked independently one by one. Using (3.34), in order to track every (sub-)channel between N outputs and K inputs respectively, we can simplify (3.33) at time $n_{k,p}+l$ as ($p=0,1,\cdots$ and $l=0,1,\cdots,L$)

$$y_r\left(n_{k,p}+l\right) = \gamma\boldsymbol{\mathcal{E}}^H\left(n_{k,p}+l\right)\mathbf{h}_{rk}^{(l)}\left(p\right)+v_r\left(n_{k,p}+l\right) \tag{3.35}$$

We intend to use only training sessions for subblock-wise channel tracking regardless of data sessions. Further define

$$\begin{aligned}
\mathbf{y}_r(p) &:= \begin{bmatrix} y_r(n_{k,p}) & y_r(n_{k,p}+1) & \cdots & y_r(n_{k,p}+L) \end{bmatrix}^T, \\
\mathbf{v}_r(p) &:= \begin{bmatrix} v_r(n_{k,p}) & v_r(n_{k,p}+1) & \cdots & v_r(n_{k,p}+L) \end{bmatrix}^T, \\
\mathbf{\Psi}(p) &:= \begin{bmatrix} \boldsymbol{\mathcal{E}}(n_{k,p}) & & & \\ & \boldsymbol{\mathcal{E}}(n_{k,p}+1) & & \\ & & \ddots & \\ & & & \boldsymbol{\mathcal{E}}(n_{k,p}+L) \end{bmatrix}^H.
\end{aligned}$$

Then by (3.35),

$$\mathbf{y}_r(p) = \gamma \mathbf{\Psi}(p) \mathbf{h}_{rk}(p) + \mathbf{v}_r(p). \tag{3.36}$$

Thus we have obtained a linear discrete-time system represented by (3.29) and (3.36). Kalman filtering can be applied to track $\mathbf{h}_{rk}(p)$, each (sub-)channel's BEM coefficients between r-th receiver and k-th user. Since Kalman recursive filtering (for $p = 0, 1, \cdots$) is applied one by one (for each pair rk) independently, we can just follow the same algorithms in Section 3.2. After Kalman filtering for every p, we generate the channel for the p-th subblock by the estimate $\hat{\mathbf{h}}(p \mid p)$ via the CE-BEM (3.21) as

$$\hat{h}_{rk}(n; l) = \boldsymbol{\mathcal{E}}^H(n) \hat{\mathbf{h}}_{rk}^{(l)}(p \mid p) \tag{3.37}$$

for $n = pm_b, pm_b + 1, \cdots, (p+1)m_b - 1$. The definition of $\hat{\mathbf{h}}_{rk}^{(l)}(p \mid p)$ is similar to (3.26). Thus we obtain the entire $N \times K$ channel matrix for the current subblock.

Computational Complexity

Here we consider computational complexity of channel estimation algorithm using the floating point operation (flop) count for the channel estimation algorithm of each simulation program. One floating point multiply/divide and associated adds/subtracts is called a flop since the number of multiplies and divides often determines the CPU time; general multiplication of $(m \times p)$ matrix $\mathbf{A}$ and $(p \times n)$ matrix $\mathbf{B}$ has mnp flops. For the explicit inverse of $(n \times n)$ matrix $\mathbf{A}$ with $(n \times p)$ or $(p \times n)$ matrix $\mathbf{B}$, i.e., $\mathbf{A}^{-1}\mathbf{B}$ or $\mathbf{B}\mathbf{A}^{-1}$ is solved by LU factorization using Gaussian elimination with partial pivoting and hence has $n^3/3 + pn^2$ flops.

The detailed flops counts for one iteration of subblock-wise tracking algorithm are shown in Table 3.1. Note that we have $\bar{P} = T_B/m_b$ iterations for one block length of T_B. Defining

the number of flops for one cycle of subblock-wise tracking and block-wise LS one as ($M = Q(L+1), L_p = L+1$)

$$f_c(\text{SB-KF}) = M^3 + M^2(3L_p+3) + M(2L_p^2+5L_p+1) + L_p^3/3 + L_p^2,$$
$$f_c(\text{BA-LS}) = M^3/3 + M^2(2\bar{P}L_p+1) + M(\bar{P}L_p+1),$$

we compare the overall flops counting for $T = 200, Q = 5$ and $T = 400, Q = 9$ with a block of length $T_a = 400$ in Table 3.2. We set the overall flops for BA-LS($T = 200, Q = 5$) to be one and take $T_B = T/\Lambda, \Lambda = 2(\text{oversampling}), m_b = 20, \bar{P} = T_B/m_b, L_p = L+1 = 3$.

Note that the computational complexity of MIMO channels with K inputs and N outputs is obtained by simply NK times of its single-input single-output (SISO) case since the algorithm for MIMO channels are implemented one-by-one independently for every pair between r-th output and k-th input by the design of TM training scheme.

Table 3.1: Subblock-wise Kalman filtering: flop count for channel estimation over one subblock of m_b symbols.

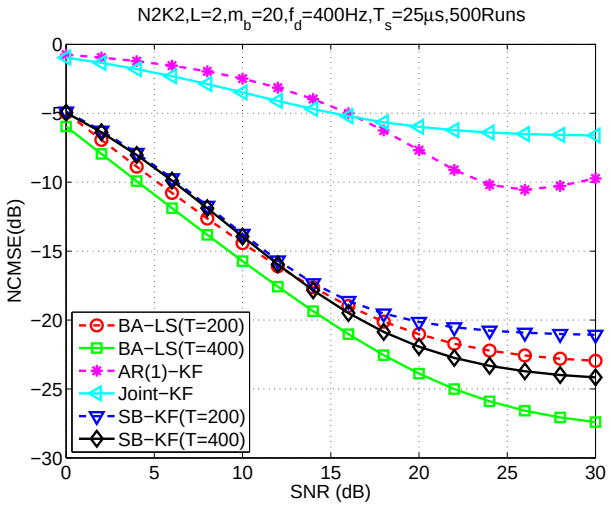
Operation	Number of flops
$\hat{\mathbf{h}}(p \mid p-1) = \alpha\hat{\mathbf{h}}(p-1 \mid p-1)$	M
$\mathbf{R}_h(p \mid p-1) = \lvert\alpha\rvert^2\mathbf{R}_h(p-1 \mid p-1) + \sigma_w^2\mathbf{I}_M$	$M^2(2)$
$\mathbf{R}_\eta(p) = \lvert\gamma\rvert^2\mathbf{\Psi}(p)\mathbf{R}_h(p \mid p-1)\mathbf{\Psi}^H(p) + \sigma_v^2\mathbf{I}_{L+1}$	$M^2L_p + M(L_p^2+L_p) + L_p^2$
$\mathbf{K}(p) = \gamma\mathbf{R}_h(p \mid p-1)\mathbf{\Psi}^H(p)\mathbf{R}_\eta^{-1}(p)$	$M^2(L_p+1) + ML_p^2 + L_p^3/3$
$\hat{\mathbf{h}}(p \mid p) = \hat{\mathbf{h}}(p \mid p-1) + \mathbf{K}(p)\left[\mathbf{y}(p) - \gamma\mathbf{\Psi}(p)\hat{\mathbf{h}}(p \mid p-1)\right]$	$M(3L_p)$
$\mathbf{R}_h(p \mid p) = [\mathbf{I}_M - \gamma\mathbf{K}(p)\mathbf{\Psi}(p)]\mathbf{R}_h(p \mid p-1)$	$M^3 + M^2L_p + ML_p$
Total : $M^3 + M^2(3L_p+3) + M(2L_p^2+5L_p+1) + L_p^3/3 + L_p^2$	

Table 3.2: Block-wise least-square and subblock-wise Kalman filtering: comparative flop count for channel estimation over one block of $T_a = 400$ symbols.

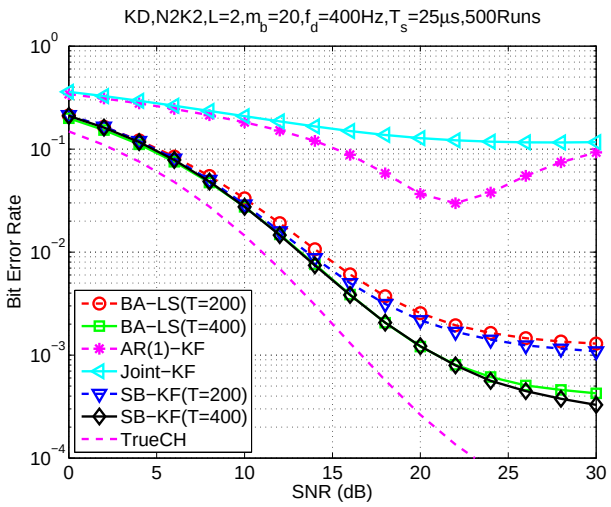
Scheme	Number of flops	relative flops
BA-LS(T=200,Q=5)	$f_c(\text{BA-LS})(T_a/T_B)$	1.00
BA-LS(T=400,Q=9)	$f_c(\text{BA-LS})(T_a/T_B)$	3.11
SB-KF(T=200,Q=5)	$\bar{P}f_c(\text{SB-KF})(T_a/T_B), \bar{P} = 5$	3.96
SB-KF(T=400,Q=9)	$\bar{P}f_c(\text{SB-KF})(T_a/T_B), \bar{P} = 10$	17.61

3.3.2 Simulation Examples

Example 1



(a) NCMSE vs SNR



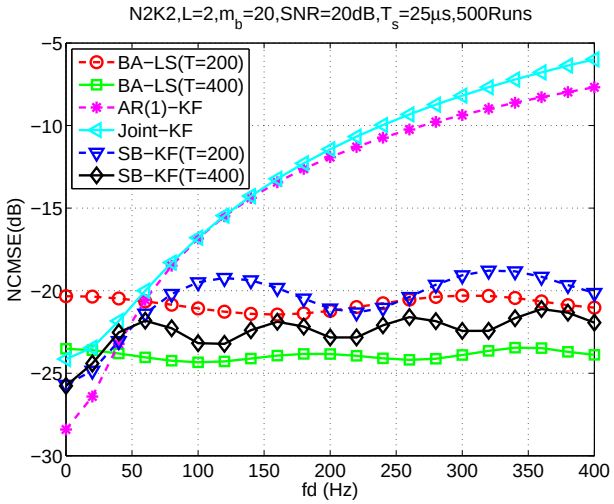
(b) BER vs SNR, with Kalman detector of equalization delay $d = 5$

Figure 3.8: Subblock-wise Kalman MIMO channel estimation: performance comparison for SNR's under $N = K = 2, f_d T_s = 0.01, m_b = 20$.

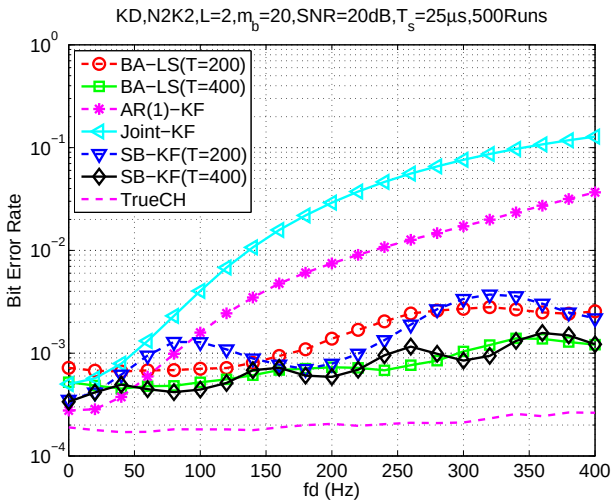
A random time- and frequency-selective Rayleigh fading MIMO channel is considered. We assume $\mathbf{h}(n;l)$ are zero-mean, complex Gaussian, and spatially white. We take $L = 2$ (3 taps) in (3.20), and $\sigma_h^2 = E\{h_{rk}(n;l)h_{rk}^*(n;l)\} = 1/(L+1)$ between k-th user and r-th receiver. For different l's, $\mathbf{h}(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75] (with a correction in the appendix of [63]).

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 0 to 400Hz, or the normalized Doppler spread $f_d T_s$ from 0 to 0.01 (corresponding to a maximum mobile velocity from 0 to 216km / h). 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.

The TM training scheme of [30], which is optimal for channels satisfying critically sampled CE-BEM representation, is adopted, where each subblock of equal length m_b symbols consists of an information session of m_d symbols and a succeeding training session of m_t symbols ($m_b = m_d + m_t$). We assume that each information symbol has unit power, while at every training session given by (2.24), we set $\gamma = \sqrt{K(L+1)+L}$ so that the average power per symbol at training sessions is equal to that of information sessions. In the simulations,



(a) NCMSE vs f_d



(b) BER vs f_d, with Kalman detector of equalization delay $d = 5$

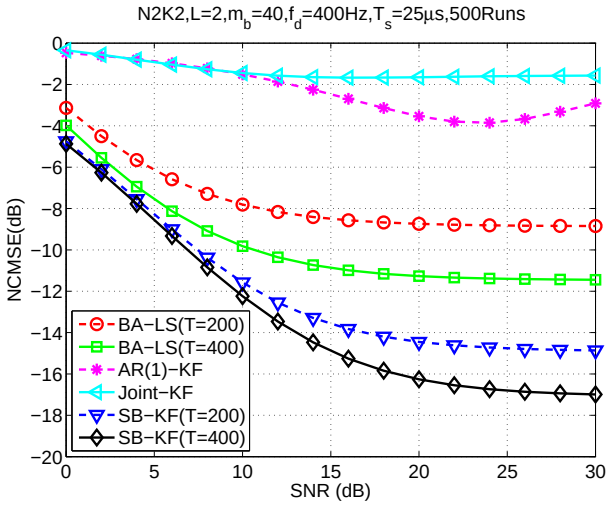
Figure 3.9: Subblock-wise Kalman MIMO channel estimation: performance comparison for Doppler spread under $N = K = 2, \mathrm{SNR} = 20\,\mathrm{dB}, m_b = 20$.

we consider a simple two-receiver and two-user scenario, i.e., $N = 2, K = 2$ with the same transmitted power. We set $m_b = 20$, $m_d = 12$ and $m_t = 8$ with $\gamma = \sqrt{8}$ for every user following the TM training scheme.

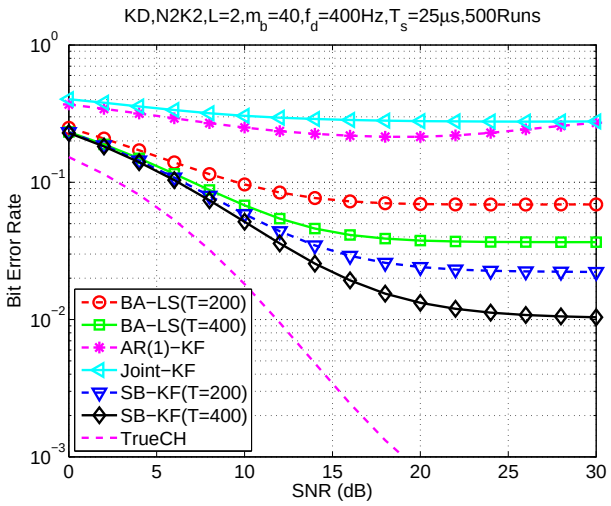
We compared the following five schemes:

1. The block-adaptive channel estimation in [30], where the transmitted symbols are segmented into consecutive blocks of T_B symbols each. Every block consists of $\bar{P}$ ($\geq Q$) subblocks as introduced in Section 2.3. We consider two models corresponding to $T = 200$ and 400 respectively, so that $Q = 5$ and 9 by (3.23). We use an oversampled CE-BEM with $T_B = T/2$ for $m_b = 20$ since the over-sampled CE-BEM approximates WSSUS channels much better than critically sampled CE-BEM [14]; whereas for $m_b = 40$, since an oversampled CE-BEM is not possible to satisfy $\bar{P} \geq Q$, we take $T_B = T$. Considering the oversampled CE-BEM basis functions are not orthogonal, we apply a regularized least-square approach with a regularization parameter $\beta = 1$; furthermore, the results were worse in the oversampled CE-BEM case when no regularization was used, caused by possible ill-conditioning of certain matrices when the basis functions are not orthogonal. In the figures, this scheme is denoted by "BA-LS".

2. The channel tracking scheme in [76] using a first-order AR model. Channel tracking is performed at training sessions only. For data sessions, the receiver updates the channel via $\hat{\mathbf{h}}(n;l) = \alpha_c \hat{\mathbf{h}}(n-1;l)$. We assume that only the upper-bound of the Doppler



(a) NCMSE vs SNR



(b) BER vs SNR, with Kalman detector of equalization delay $d = 5$

Figure 3.10: Subblock-wise Kalman MIMO channel estimation: performance comparison for SNR's under $N = K = 2, f_d T_s = 0.01, m_b = 40$.

spread is known. Then for Jakes' model, $\alpha_c = J_0(2\pi f_d T_s) = 0.999$ for $f_d T_s = 0.01$, where $J_0(\cdot)$ denotes the zero-th Bessel function of the first kind. This scheme is denoted by "AR(1)-KF".

3. We also compared the approach of joint channel estimation and data detection via extended Kalman filtering (EKF) in [68] without the turbo equalization (and channel coding) procedure, using an AR(1) model for channel taps as above. This scheme is denoted by "Joint-KF".

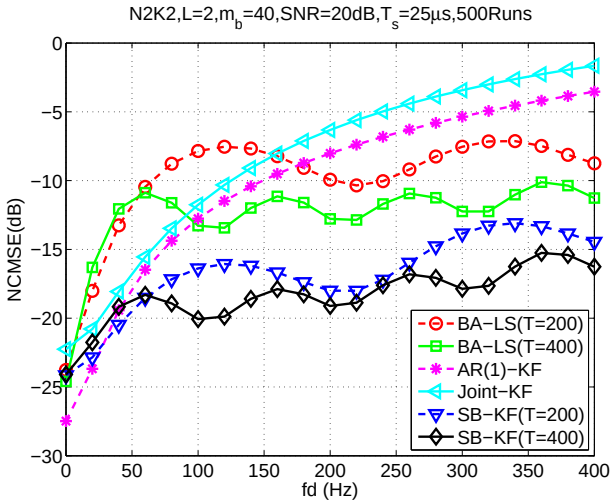
4. Our subblock-wise tracking using CE-BEM, which is denoted by "SB-KF", takes $T = 200$ and 400 for two different settings of CE-BEM, and $Q = 5$ and 9 correspondingly. We also take AR coefficient, $\alpha = 0.99$ for $T = 200$ and $\alpha = 0.9944$ for $T = 400$.

5. Perfect channel estimates are available at the receiver, which is denoted by "TrueCH".

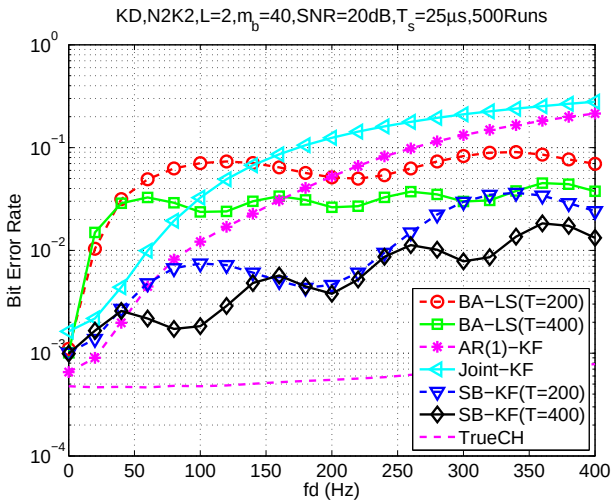
We evaluate the performances of various schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

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

where $\mathbf{h}^{(i)}(n;l)$ is the true channel and $\hat{\mathbf{h}}^{(i)}(n;l)$ is the estimated channel at the i-th Monte Carlo run, among total M_r runs. The Kalman detector in Section 2.4.1 with delay $d = 5$ is



(a) NCMSE vs f_d



(b) BER vs f_d, with Kalman detector of equalization delay $d = 5$

Figure 3.11: Subblock-wise Kalman MIMO channel estimation: performance comparison for Doppler spread under $N = K = 2, \text{SNR} = 20\,\text{dB}, m_b = 40$.

used using the channel estimates obtained by each scheme. In each run, a symbol sequence of length, $T_N = 4000$ for each user is modulated by QPSK and the first 200 symbols are discarded in evaluations. All the simulation results are based on 500 runs.

In Fig. 3.8, the performances of the five schemes are compared for different SNR's under fixed normalized Doppler spread $f_d T_s = 0.01$. It is seen that the CE-BEM-based tracking schemes, BA-LS [30] and SB-KF, have superior performance compared to the symbol-wise AR-based AR(1)-KF and Joint-KF. The symbol-wise AR-model cannot capture the fast-varying channel even with high SNR's. Using the oversampled CE-BEM, more basis functions $(T = 400, Q = 9)$ in (3.21) result in better performance. For the block-adaptive scheme, larger block size $(T_B = 200 = T/\Lambda, \Lambda = 2$ for oversampled CE-BEM) may induce larger estimation variance. However, this strategy can well improve the performance of the proposed SB-KF scheme since all past data are implicitly utilized in Kalman filtering-based subblock tracking and we also update every subblock.

In Fig. 3.9, we compare the same schemes over different Doppler spread f_d's with SNR $=$ 20 dB. For slowly fading channels, the CE-BEM-based trackings are slightly inferior to the symbol-wise AR(1)-KF or Joint-KF, since more unknown parameters (BEM coefficients) are involved in and therefore result in higher estimation variance. As f_d increases, the BA-LS and SB-KF gradually outperforms the AR(1)-model-based schemes, since the time variations of the channel have been well captured in CE-BEM while AR(1)-KF and Joint-KF schemes cannot track the fast channel variations. Since the CE-BEM used in the schemes is based

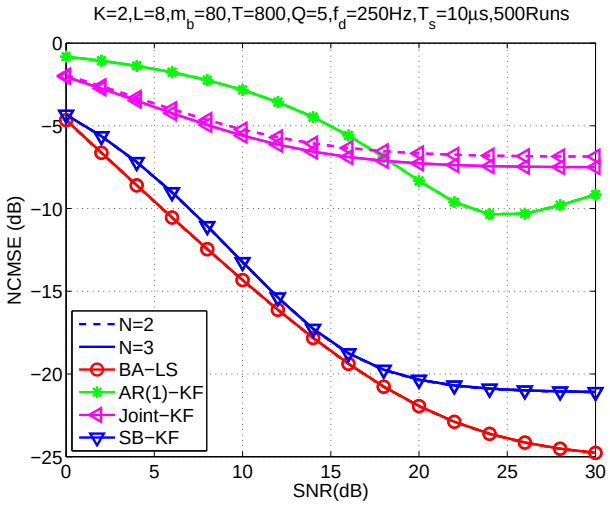
solely on the maximum Doppler shift $f_d T_s = 0.01$, the performances of the CE-BEM-based schemes are not sensitive to the "actual" Doppler spread if it is no larger than the assumed maximum value. For the purpose of channel estimation, we need not the exact Doppler spread but an upper bound of it when using CE-BEM.

In Fig. 3.10, we increase the subblock size $m_b = 20$ to 40 and hence reduce the training ratio 40% to 20%, so that we can save the transmitted symbols used for the additive training. For the block-adaptive LS approach, the critically-sampled CE-BEM is considered and we have $T_B = T(\Lambda = 1), \bar{P} = T_B/m_b \geq Q$. Note that, for this longer subblock size $m_b = 40$, BA-LS scheme with the oversampled CE-BEM cannot satisfy the unknown parameter identifiability condition. The performances in NCMSE and BER of BA-LS with critically-sampled CE-BEM deteriorate significantly due to the spectral leakage. However, unlike block-adaptive scheme, the proposed subblock-wise approach has no identifiability condition to estimate unknown BEM coefficients since we update subblock-by-subblock and the block size T_B is not a strict factor.

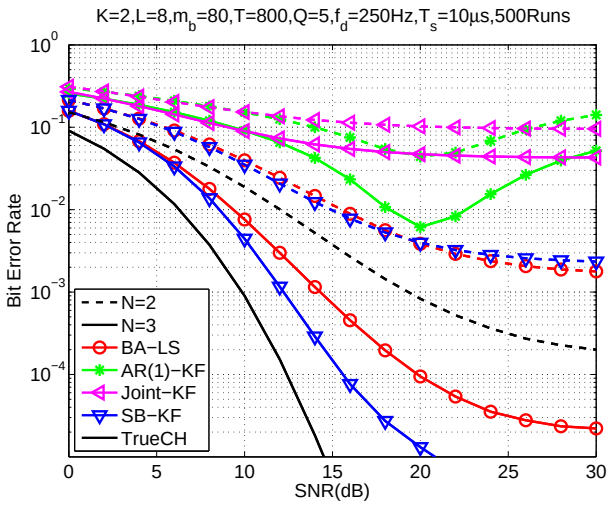
Fig. 3.5 show the performances of this longer subblock size $m_b = 40$ over different Doppler spread f_d's. It is seen that CE-BEM is more adequate to keep track of the wireless channel than symbol-wise AR model; AR(1)-KF and Joint-KF deteriorate as f_d increases. The block-adaptive LS based on the critically-sampled CE-BEM also deteriorates with Doppler spread; it has a good performance only when $f_d \approx 0$. [One can try BA-LS with the oversampled CE-BEM, neglecting the identifiability. However, the latter performances is similar as the former ones.] On the other hand, while SB-KF has some variations and slightly goes worse as f_d increases, the overall performance is satisfactory.

Example 2

The random Rayleigh fading MIMO channel in this example is as in Example 1, except that we change the channel length L from 2 to 8 leading to 9 taps, and the power delay profile is exponential satisfying $E\{|h_{rk}(n:l)|^2\} = \sigma_h^2(l) = ae^{-l/L}$ between k-th user and r-th receiver where the constant a is pick to satisfy $\sum_{l=0}^{L} \sigma_h^2(l) = 1$. We consider a communication system with carrier frequency of 2GHz, data rate of 0.1MBd (mega-Bauds), therefore $T_s = 10\,\mu$s, and a varying Doppler spread $f_d = 250\,$Hz, or the normalized Doppler spread $f_d T_s = 0.0025$ (corresponding to a maximum mobile velocity of 135km / h). The training session is described by (2.24) of length $m_t = K(L+1) + L = 26$ symbols with $\gamma = \sqrt{K(L+1) + L}$ so that the average symbol power of training sessions is equal to that of information sessions. We select the period of the CE-BEM $T = 800$ symbols, and hence $Q = 5$ by (3.23) and a block size $T_B = 400$ for oversampled CE-BEM.



(a) NCMSE vs SNR

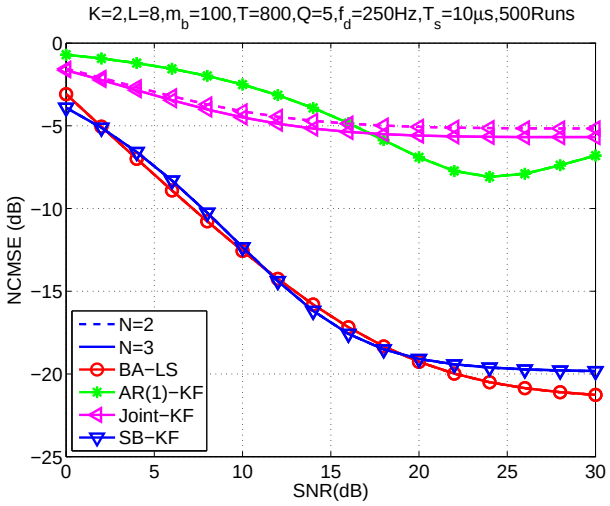


(b) BER vs SNR, with Kalman detector of equalization delay $d = 10$

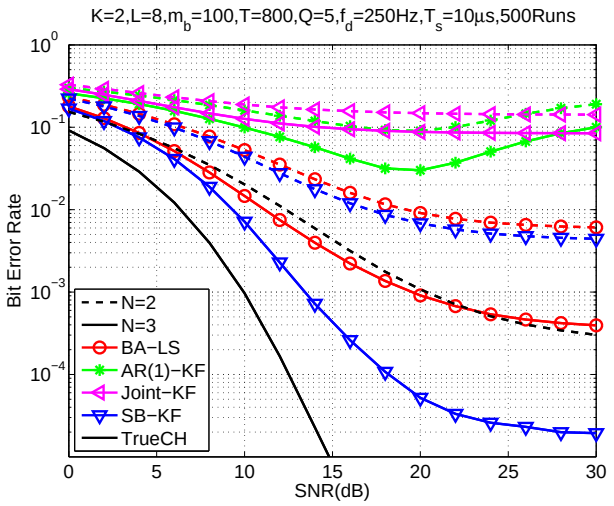
Figure 3.12: Subblock-wise Kalman MIMO channel estimation: performance comparison for SNR's under $K = 2, f_d T_s = 0.0025, m_b = 80$.

Four estimation and tracking schemes are compared: block-adaptive channel estimation in [30] using the regularized LS approach ("BA-LS"), channel tracking scheme in [76] using a first-order AR model ("AR(1)-KF"), joint channel estimation and data detection via EKF in [68] ("Joint-KF") and our subblock-wise Kalman tracking ("SB-KF"). The BER's are evaluated by employing the Kalman detector with delay $d = 10$, using the channel estimates obtained by each scheme.

In Fig. 3.12, the performances of the above schemes with the subblock size $m_b = 80$ are compared for different SNR's. For the subblock-wise Kalman tracking, we picked the AR coefficient $\alpha = 0.993$. Similarly as Example 1, the symbol-wise AR(1)-KF [76] and Joint-KF [68] approaches is not appropriate for the fast-varying channels. Using the oversampled CE-BEM, the block-adaptive scheme [30] with regularized LS approach (BA-LS) has similar performances as the subblock-wise Kalman filtering (SB-KF). Clearly, the CE-BEM-based schemes, where BEM coefficients slowly change block by block, outperform AR-based ones. To compare BA-LS and SB-KF schemes more clearly, we also have the examples with $N = 3$ receivers. Since we assume the channel is spatially uncorrelated for different outputs, the NCMSE curves show the same performances for $N = 2$ and $N = 3$. [Joint-KF has different NCMSE curves for $N = 2$ and $N = 3$ since it estimates channel and data symbols jointly.] The superiority in NCMSE does not guarantee the superiority in BER since NCMSE and BER have no monotone relationship each other. When increasing the number of receivers,



(a) NCMSE vs SNR



(b) BER vs SNR, with Kalman detector of equalization delay $d = 10$

Figure 3.13: Subblock-wise Kalman MIMO channel estimation: performance comparison for SNR's under $K = 2, f_d T_s = 0.0025, m_b = 100$.

we have better performance in BER's. In particular, the proposed subblock-wise Kalman filtering outperforms block-adaptive LS approach with more receivers.

In Fig. 3.13, we increase the subblock size $m_b = 80$ to $m_b = 100$ and hence have more spectrally-efficient transmission; the training overhead decreases from 15% to 12%. We set the AR coefficient $\alpha = 0.992$ for the subblock-wise Kalman tracking. The performances of all schemes become worse due to the lower frequency of training sessions. But the relative superiority of the subblock-wise schemes over block-wise schemes is unchanged.

3.4 AR(1) Coefficient α for Subblock-wise Kalman Tracking

Here we discuss choice of the AR coefficient α for subblock-wise channel estimation using Kalman filter investigated in this chapter for a class of (true) channel models. We assume that the components of the MIMO channel $\mathbf{h}(n;l)$ are mutually independent and identically distributed (i.i.d.) with zero-mean, complex Gaussian, and spatially white with autocorrelation $E\left\{\mathbf{h}(n;l)\,\mathbf{h}^H(n;l)\right\} = \sigma_h^2 \mathbf{I}_N$. Then a simplified AR(1) model for the BEM coefficient vector is given by

$$\mathbf{h}(p) = \mathbf{A}_1 \mathbf{h}(p-1) + \mathbf{w}(p), \tag{3.38}$$

where $\mathbf{A}_1$ is the AR coefficient matrix, and the driving noise vector $\mathbf{w}(p)$ is zero-mean white complex Gaussian with variance $\sigma_w^2\mathbf{I}$ and statistically independent of $\mathbf{h}(p-1)$. Defining

$$\mathbf{h}_B(n;l) := [\mathbf{h}(n;l)\ \mathbf{h}(n+1;l)\ \cdots\ \mathbf{h}(n+T_B-1;l)]^T, \tag{3.39}$$

consider two overlapping blocks that differ by just one subblock: $\mathbf{h}_B(pm_b;l)$ and $\mathbf{h}_B((p+1)m_b;l)$ with $\mathbf{h}^{(l)}(p)$ and $\mathbf{h}^{(l)}(p+1)$ as the corresponding BEM coefficient vector respectively. Further define

$$\mathbf{E}(n) := \left[\boldsymbol{\mathcal{E}}(n)\quad \boldsymbol{\mathcal{E}}(n+1)\quad \cdots\quad \boldsymbol{\mathcal{E}}(n+T_B+1)\right]^H \otimes \mathbf{I}_{NK}, \tag{3.40}$$

$$\boldsymbol{\Gamma} := diag\left\{e^{j\omega_1 m_b}, e^{j\omega_2 m_b}, \cdots, e^{j\omega_Q m_b}\right\} \otimes \mathbf{I}_{NK}, \tag{3.41}$$

where $\mathbf{E}(n)$ is $NKT_B \times NKQ$ and $\boldsymbol{\Gamma}$ is $NKQ \times NKQ$. It follows that

$$\mathbf{h}_B(pm_b;l) = \mathbf{E}(pm_b)\mathbf{h}^{(l)}(p), \tag{3.42}$$

$$\mathbf{h}_B((p+1)m_b;l) = \mathbf{E}(pm_b)\boldsymbol{\Gamma}\mathbf{h}^{(l)}(p+1). \tag{3.43}$$

Defining

$$\tilde{\mathbf{h}}_B(n) := \left[\mathbf{h}_B^T(n;0)\quad \mathbf{h}_B^T(n;1)\quad \cdots\quad \mathbf{h}_B^T(n;L)\right]^T, \tag{3.44}$$

we have

$$\tilde{\mathbf{h}}_B(pm_b) = \tilde{\mathbf{E}}(pm_b)\mathbf{h}(p), \tag{3.45}$$

$$\tilde{\mathbf{h}}_B((p+1)m_b) = \tilde{\mathbf{E}}(pm_b)\tilde{\boldsymbol{\Gamma}}\mathbf{h}(p+1), \tag{3.46}$$

where $\tilde{\mathbf{E}}(pm_b) := diag\{\mathbf{E}(pm_b),\ \mathbf{E}(pm_b),\ \cdots\ \mathbf{E}(pm_b)\}$, and $\tilde{\boldsymbol{\Gamma}} := diag\{\boldsymbol{\Gamma},\ \boldsymbol{\Gamma},\ \cdots,\ \boldsymbol{\Gamma}\}$.

If (3.38) holds, then using the Yule-Walker equation

$$\mathbf{A}_1 = E\left\{\mathbf{h}(p+1)\mathbf{h}^H(p)\right\}\left[E\left\{\mathbf{h}(p)\mathbf{h}^H(p)\right\}\right]^{-1}, \tag{3.47}$$

where using (3.45) and (3.46) we have

$$E\left\{\mathbf{h}(p+1)\mathbf{h}^H(p)\right\} = \tilde{\boldsymbol{\Gamma}}^{-1}\tilde{\mathbf{E}}^{\dagger}(pm_b)E\left\{\tilde{\mathbf{h}}_B((p+1)m_b)\tilde{\mathbf{h}}_{\mathbf{B}}^{\mathbf{H}}(\mathbf{pm_b})\right\}\tilde{\mathbf{E}}^{\dagger H}(pm_b), \tag{3.48}$$

$$E\left\{\mathbf{h}(p)\mathbf{h}^H(p)\right\} = \tilde{\mathbf{E}}^{\dagger}(pm_b)E\left\{\tilde{\mathbf{h}}_B(pm_b)\tilde{\mathbf{h}}_{\mathbf{B}}^{\mathbf{H}}(\mathbf{pm_b})\right\}\tilde{\mathbf{E}}^{\dagger H}(pm_b), \tag{3.49}$$

and $E\left\{\tilde{\mathbf{h}}_B((p+1)m_b)\tilde{\mathbf{h}}_B^H(pm_b)\right\}$ and $E\left\{\tilde{\mathbf{h}}_B(pm_b)\tilde{\mathbf{h}}_B^H(pm_b)\right\}$ can be calculated using (3.39) and (3.44) if we know the channel correlation function $\mathbf{R}_h(\tau)$.

Since typically $\mathbf{R}_h(\tau)$ will not be available, in order to simplify we assumed $\mathbf{A}_1 = \alpha\mathbf{I}$ and $E\left\{\mathbf{w}(p)\mathbf{w}^H(p)\right\} = \sigma_w^2\mathbf{I}$, leading to

$$\mathbf{h}(p) = \alpha\mathbf{h}(p-1) + \mathbf{w}(p). \tag{3.50}$$

If the channel is wide-sense stationary uncorrelated scattering (WSSUS) and coefficients $h_q(l)$'s are independent (as assumed in [70]), then by (3.50) and Yule-Walker equations, we have

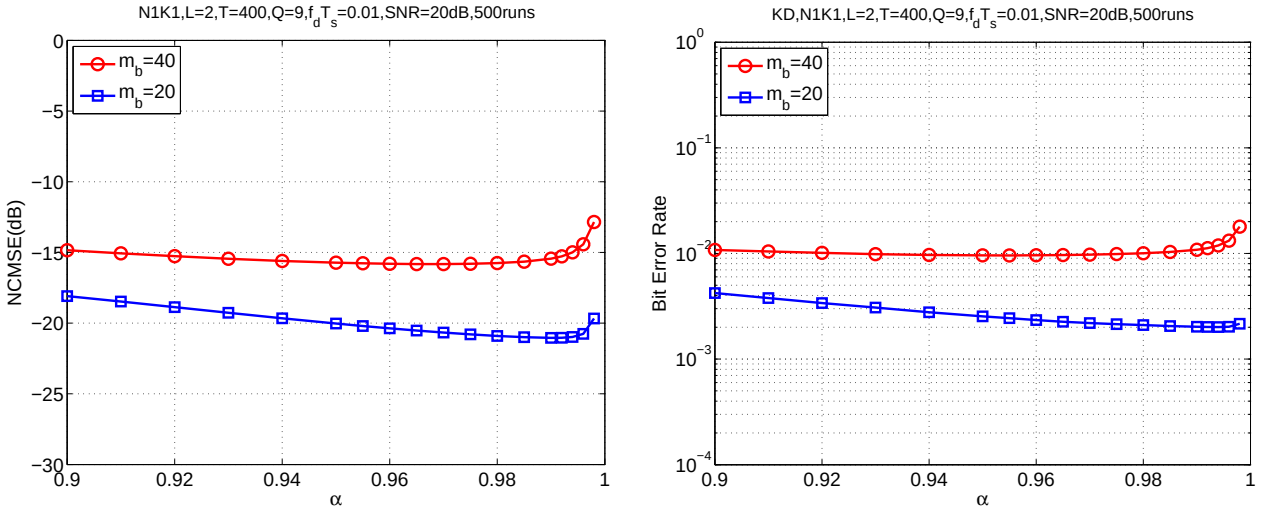
$$\begin{aligned}\alpha &= \left(\frac{E\left\{\mathbf{h}^H(p+1)\mathbf{h}(p)\right\}}{E\left\{\mathbf{h}^H(p)\mathbf{h}(p)\right\}}\right)^* \\ &= \frac{tr\left\{\tilde{\mathbf{\Gamma}}^{-1}\tilde{\mathbf{E}}^\dagger(pm_b)E\left\{\tilde{\mathbf{h}}_B((p+1)m_b)\tilde{\mathbf{h}}_B^H(pm_b)\right\}\tilde{\mathbf{E}}^{\dagger H}(pm_b)\right\}}{tr\left\{\tilde{\mathbf{E}}^\dagger(pm_b)E\left\{\tilde{\mathbf{h}}_B(pm_b)\tilde{\mathbf{h}}_B^H(pm_b)\right\}\tilde{\mathbf{E}}^{\dagger H}(pm_b)\right\}},\end{aligned} \tag{3.51}$$

and $\sigma_w^2 = \sigma_h^2(1-|\alpha|^2)/Q$. Assuming for different l's, $\mathbf{h}\,(n;l)$'s are mutually independent and satisfy Jakes' model, we know the correlation of $h(n;l)$ and then we can calculate α via (3.51). For the over-sampled CE-BEM, $\tilde{\mathbf{E}}(n)$ may be ill-conditioned since the basis functions are not orthogonal. Note that (3.51) also requires knowledge of $\mathbf{R}_h(\tau)$. In order to avoid this, one can somewhat arbitrarily pick a value of α; this has been done in, e.g. [28] (in a different but similar context). Since the coefficients evolve slowly, we will have $\alpha \approx 1$ (but $\alpha < 1$ for tracking uses). The results of (3.51) are similar to (2.12) in Section 2.2.1 except that in the former changes occur every subblock whereas in the latter the changes occur every symbol.

To gain more insight, let us consider a specific channel tap $h(n;l)$ following the Jakes' spectrum (also used in all simulation examples in this paper). When $T = 400, T_B = 200, Q = 9$, and $f_dT_s = 0.01$, one gets $\alpha = 0.9814$ and 0.9425 for $m_b = 20$ and 40, respectively; i.e., dependence between subblocks decreases with increasing m_b. With f_dT_s held at 0.01, when T and T_B are halved to 200 and 100, respectively, (leading to $Q = 5$), one gets $\alpha = 0.9605$ and 0.8791 for $m_b = 20$ and 40, respectively; i.e., as T_B increases, α increases, and vice versa. When $T = 400, T_B = 200$, and $f_dT_s = 0.005$ (leading to $Q = 5$), one gets $\alpha = 0.9893$ and 0.9605 for $m_b = 20$ and 40, respectively; i.e., as f_d decreases, α increases and vice versa. These numerical results are consistent with one's intuition: faster channel variations call for smaller α values (less dependence), and vice versa.

In Figs. 3.14, the performances of subblock-wise Kalman tracking are shown for different values of α based on 500 Monte Carlo runs with the same environment as the simulation example in Section 3.2.2 and 3.3.2. In our simulations, we picked the AR(1) coefficient $\alpha = 0.9944$ and 0.9667 for $m_b = 20$ and 40 respectively, where $T = 400, Q = 9$ in CE-BEM.

It is seen that while the performance is not sensitive to the value of α over a relatively wide range of values, it does deteriorate as α approaches one. Note that α is close to one ($\alpha \approx 1$) but it is not one since $\alpha = 1$ in AR(1) model implies time-invariance and $\alpha < 1$ permits tracking by discounting older values of the channel BEM coefficients.



(a) NCMSE vs α

(b) BER vs α, with DFE of $l_f = 8, l_b = 2$ and delay $d = 5$

Figure 3.14: Subblock-wise Kalman channel estimation: performances for AR(1) coefficient α's, under $N = K = 1, f_d T_s = 0.01, \text{SNR} = 20\text{dB}, m_b = 20$ and 40.

3.5 Conclusions

We presented a subblock-wise tracking approach to doubly-selective channel estimation, exploiting the oversampled complex exponential basis expansion model (CE-BEM) for the overall channel variations, and an autoregressive (AR) model to update the BEM coefficients. We tracked the BEM coefficients via Kalman filtering, based on time-multiplexed periodically transmitted training symbols. Rather than track each channel tap gain, we estimated the BEM coefficients subblock by subblock, and then (re-)generated the channel via the CE-BEM with the estimated BEM coefficients. In this way, the modeling mismatch introduced by the conventionally used symbol-wise AR channel model can be greatly reduced, and hence better performance can be achieved in fast-fading environments. We also extended the subblock-wise tracking to MIMO systems. Simulation examples demonstrated its superior performance over some existing doubly-selective channel tracking schemes, over a wide range of Doppler spreads.

Chapter 4

Recursive Least-squares Doubly-selective Channel Estimation Using Exponential Basis Models and Subblock-wise Tracking

We present the subblock-wise channel estimation scheme, exploiting the oversampled complex exponential basis expansion model (CE-BEM) for the overall channel variations. 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 that of the channel. We apply the exponentially-weighted (EW) and sliding-window (SW) recursive least-squares (RLS) algorithms to track the BEM coefficients, using time-multiplexed periodically transmitted training symbols. Simulation examples demonstrate its superior performance over the conventional block-wise channel estimator.

4.1 Introduction

In Chapter 3, a subblock-wise tracking approach was proposed for doubly-selective channels using time-multiplexed (TM) training. It exploits the CE-BEM for the overall channel variations, and a first-order AR model to describe the evolutions of the BEM coefficients. The slow-varying BEM coefficients, rather than the fast-varying channel, are tracked and updated at each training session; during information sessions, channel estimates are generated by the CE-BEM using the estimated BEM coefficients. The 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. The subblock-wise approaches achieve better performance in fast-fading environments, than using conventional symbol-wise AR models [6] or block-wise BEM representations [30, 70].

However, the approach in Chapter 3 used the first-order AR model for BEM coefficients, which is not necessarily suited to a real channel environment, and possibly causes modeling error in estimation. To solve this problem, we propose an adaptive channel estimation scheme without an arbitrary *a priori* model for the BEM coefficients. Two finite-memory adaptive filtering algorithms, the EW- and SW-RLS algorithm, are considered for subblock-wise channel tracking.

A decision-directed channel tracking via the RLS algorithm within a sliding window using polynomial BEM has been investigated in [8] and channel estimation via Kalman filtering using polynomial or complex exponential BEM for OFDM systems has been explored

in [26, 41, 44]; all these approaches use block-by-block updating (for single input systems). Our contribution exploits subblock-wise updating, where several subblocks comprise one block. For parameter identifiability one needs the number of subblocks at least as large as the number of basis functions used for channel modeling. Data in just one subblock do not satisfy the parameter identifiability requirements. We also extend the proposed schemes to multi-input multi-output (MIMO) systems.

4.2 Subblock-wise RLS Channel Estimation using CE-BEM [23]

Suppose that we collect the received signal over a time period T_B, a block size. We estimate the time-variant channel using the useful channel models and time-multiplexed training, and then detect the information symbols using this estimated channel. By exploiting the invariance of the coefficients of CE-BEM over each block, hence, each of the $\bar{P}$ subblocks per block of length T_B symbols, we seek to adaptively estimate the doubly-selective channel via subblock-wise tracking.

Consider two overlapping blocks (each consisting of sequences of T_B symbols) that differ by only one subblock of length m_b: the "past" block beginning at time n_0 and the "current" block beginning at time $n_0 + m_b$. Since the two blocks overlaps so significantly, one would expect the BEM coefficients to vary only a little from the past block to the current overlapping block. Therefore, we propose a subblock-wise channel tracking as in [52] where we estimate $\mathbf{h}_q(l)$ in (4.2) subblock-by-subblock for their variations, rather than the anew with every non-overlapping block as in [70].

Consider a doubly-selective (time- and frequency-selective), single-input multi-output (SIMO), finite impulse response (FIR) linear channel with N outputs. Let $\{s(n)\}$ denote a scalar information sequence that is input to the time-varying channel with discrete-time response $\{\mathbf{h}(n;l)\}$ (N-column vector channel response at time instance n to a unit input at time instance $n-l$), assuming $\{s(n)\}$ is independent and identically distributed (i.i.d.) with zero mean and variance σ_s^2. Then the symbol-rate noisy N-column channel output is given by ($n = 0, 1, \ldots$)

$$\mathbf{y}(n) = \sum_{l=0}^{L} \mathbf{h}(n;l)\, s(n-l) + \mathbf{v}(n) \tag{4.1}$$

where the N-column vector $\mathbf{v}(n)$ is zero-mean, white complex Gaussian noise, with variance $\sigma_v^2 \mathbf{I}_N$. We assume that $\{\mathbf{h}(n;l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [60].

The time-varying channel with discrete-time response $\{\mathbf{h}(n;l)\}$ is represented by CE-BEM as ($n=(i-1)T_B,(i-1)T_B+1,\cdots,iT_B-1$ for i-th block and $l=0,1,\cdots,L$)

$$\mathbf{h}(n;l)=\sum_{q=1}^{Q}\mathbf{h}_q(l)e^{j\omega_q n}, \tag{4.2}$$

where $\mathbf{h}_q(l)$ is the N-column time-invariant BEM coefficient and one chooses (Λ is an integer)

$$T:=\Lambda T_B,\quad \Lambda\geq 1, \tag{4.3}$$
$$Q\geq 2\lceil f_d T T_s\rceil+1, \tag{4.4}$$
$$\omega_q:=\frac{2\pi}{T}\left[q-(Q+1)/2\right],\quad q=1,2,\cdots,Q \tag{4.5}$$
$$L:=\lfloor \tau_d/T_s\rfloor, \tag{4.6}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration.

Let $h_{r,q}(l)$ denote the r-th component of the N column vector $\mathbf{h}_q(l)$. Stack the BEM coefficients in (4.2) into vectors as

$$\mathbf{h}_r^{(l)}:=\begin{bmatrix}h_{r,1}(l) & h_{r,2}(l) & \cdots & h_{r,Q}(l)\end{bmatrix}^T \tag{4.7}$$
$$\mathbf{h}_r:=\begin{bmatrix}\mathbf{h}_r^{(0)\,T} & \mathbf{h}_r^{(1)\,T} & \cdots & \mathbf{h}_r^{(L)\,T}\end{bmatrix}^T \tag{4.8}$$
$$\mathbf{h}:=\begin{bmatrix}\mathbf{h}_1^T & \mathbf{h}_2^T & \cdots & \mathbf{h}_N^T\end{bmatrix}^T \tag{4.9}$$

of size $Q, M:=Q(L+1)$ and $NQ(L+1)$ respectively. The coefficient vectors in (4.7), (4.8) and (4.9) of the p-th subblock will be denoted by $\mathbf{h}_r^{(l)}(p), \mathbf{h}_r(p)$ and $\mathbf{h}(p)$.

Based on CE-BEM given by (4.2), define

$$\boldsymbol{\mathcal{E}}(n):=\begin{bmatrix}e^{-j\omega_1 n} & e^{-j\omega_2 n} & \cdots & e^{-j\omega_Q n}\end{bmatrix}^T,$$
$$\mathbf{s}(n):=\begin{bmatrix}s(n) & s(n-1) & \cdots & s(n-L)\end{bmatrix}^T,$$

and then the received signal can be written as

$$\mathbf{y}(n)=\left[\mathbf{s}(n)\otimes\mathbf{I}_N\right]^T\left[\mathbf{I}_{N(L+1)}\otimes\boldsymbol{\mathcal{E}}(n)\right]^H\mathbf{h}(p)+\mathbf{v}(n) \tag{4.10}$$

when the p-th subblock is being received at time n, by (4.1), (4.2)-(4.6) and (4.7)-(4.9).

We will employ the time-multiplexed (TM) training scheme as Section 2.3, proposed in [70]. Then the received signal at r-th output at time n_p+l with $l=0,1,\cdots,L$ is given

by (assuming timing synchronization)

$$y_r\left(n_p+l\right)=\gamma h_r\left(n_p+l;l\right)+v_r\left(n_p+l\right). \tag{4.11}$$

for $r=1,2,\cdots,N$. We also can simplify (4.10) at time n_p+l as $(l=0,1,\cdots,L)$

$$y_r\left(n_p+l\right)=\gamma\boldsymbol{\mathcal{E}}^H\left(n_p+l\right)\mathbf{h}_r^{(l)}\left(p\right)+v_r\left(n_p+l\right) \tag{4.12}$$

Note that each (sub-)channel at r-th output can be tracked using r-th received signal in (4.12) respectively without interference. We intend to use only training sessions for subblock-wise channel tracking regardless of data sessions. Further define

$$\begin{aligned}\mathbf{y}_r\left(p\right)&:=\begin{bmatrix}y\left(n_p\right) & y\left(n_p+1\right) & \cdots & y\left(n_p+L\right)\end{bmatrix}^T,\\ \mathbf{v}_r\left(p\right)&:=\begin{bmatrix}v\left(n_p\right) & v\left(n_p+1\right) & \cdots & v\left(n_p+L\right)\end{bmatrix}^T,\end{aligned}$$

and

$$\boldsymbol{\Psi}\left(p\right):=\begin{bmatrix}\boldsymbol{\mathcal{E}}\left(n_p\right) & & & \\ & \boldsymbol{\mathcal{E}}\left(n_p+1\right) & & \\ & & \ddots & \\ & & & \boldsymbol{\mathcal{E}}\left(n_p+L\right)\end{bmatrix}^H.$$

Then by (4.12),

$$\mathbf{y}_r\left(p\right)=\gamma\boldsymbol{\Psi}\left(p\right)\mathbf{h}_r\left(p\right)+\mathbf{v}_r\left(p\right). \tag{4.13}$$

Since CE-BEM is periodic with period T in (4.2), the algorithm memory should be less than T in order to avoid periodicity of the BEM coefficients. Note that the real channel is not periodic. Therefore, an adaptive algorithm with "finite-memory" is preferred which we implement via either exponentially-weighted RLS or sliding-window RLS approaches.

4.2.1 Exponentially-Weighted RLS Tracking

Given an $(L+1)\times 1$ measurement vector $\mathbf{y}_r\left(p\right)$, a training impulse γ and an $(L+1)\times M$ basis function matrix $\boldsymbol{\Psi}\left(p\right)$ in (4.13), we can apply exponentially-weighted regularized RLS (EW-RLS) algorithm [2, Chapter 12] to track an unknown $M\times 1$ BEM coefficient vector $\mathbf{h}_r\left(p\right)$. Choose $\mathbf{h}_r\left(p\right)$ to minimize the cost function

$$\lambda^{p+1}\beta\left\|\mathbf{h}_r\right\|^2+\sum_{i=0}^{p}\lambda^{p-i}\left\|\mathbf{y}_r(i)-\gamma\boldsymbol{\Psi}(i)\mathbf{h}_r\right\|^2 \tag{4.14}$$

where $\beta > 0$ is a regularization parameter and $0 < \lambda < 1$ is the forgetting factor. Note that the forgetting factor is used to give more weight to recent data and less weight to past data. For the subblock-wise updating, we take λ to be (much) smaller than one (e.g., 0.65) although it is very close to one (e.g., 0.99) for the symbol-wise updating.

EW-RLS tracking is initialized with $\hat{\mathbf{h}}_r(-1) = \mathbf{0}_{M\times 1}$ and $\mathbf{P}_{h_r}(-1) = \beta^{-1}\mathbf{I}_M$. Mimicking [2, Algorithm 12.3.1], EW-RLS recursion (for $p = 0, 1, \cdots$) is applied one by one for each r-th output via the following steps (we have omitted the subscript r for convenience):

$$\begin{aligned}
\mathbf{\Gamma}(p) &= \lambda\mathbf{I}_{L+1} + \gamma^2\mathbf{\Psi}(p)\mathbf{P}(p-1)\mathbf{\Psi}^H(p),\\
\mathbf{G}(p) &= \gamma\mathbf{P}(p-1)\mathbf{\Psi}^H(p)\mathbf{\Gamma}^{-1}(p),\\
\hat{\mathbf{h}}(p) &= \hat{\mathbf{h}}(p-1) + \mathbf{G}(p)\left[\mathbf{y}_s(p) - \gamma\mathbf{\Psi}(p)\hat{\mathbf{h}}(p-1)\right],\\
\mathbf{P}(p) &= \lambda^{-1}\left[\mathbf{I}_M - \gamma\mathbf{G}(p)\mathbf{\Psi}(p)\right]\mathbf{P}(p-1)
\end{aligned}$$

where $\hat{\mathbf{h}}(p)$ denotes the estimate of $\mathbf{h}(p)$ given the observations $\{\mathbf{y}(0), \mathbf{y}_s(1), \cdots, \mathbf{y}_s(p)\}$.

After RLS recursion for every p, we generate the channel for the p-th subblock by the estimate $\hat{\mathbf{h}}_r(p)$ via the CE-BEM (4.2) as

$$\hat{h}_r(n;l) = \boldsymbol{\mathcal{E}}^H(n)\,\hat{\mathbf{h}}_r^{(l)}(p) \tag{4.15}$$

for $n = pm_b, pm_b+1, \cdots, (p+1)m_b - 1$. The definition of $\hat{\mathbf{h}}_r^{(l)}(p)$ is similar to (4.7).

4.2.2 Sliding-Window RLS Tracking

Compared with EW-RLS algorithm that weakens the effects of all past data using the exponential weights, the sliding-window RLS (SW-RLS) algorithm removes the effect of earlier data using the "downdating" least-squares recursions. The downdating removes the oldest received subblock sample from the window and the updating inserts the next received subblock sample into the window. Combining the update and downdate procedures, SW-RLS algorithm only utilizes the data in a sliding window of length W, a fixed length of data [2, Chapter 12].

Since the BEM coefficients are invariant within a block of T_B symbols in (4.2), we set $W = \lfloor T_B/m_b \rfloor$ subblocks so that only the current subblock and the past $(W-1)$ subblocks within one block are used for adaptation. The cost function for SW-RLS is given by

$$\beta\|\mathbf{h}_r\|^2 + \sum_{i=p-W+1}^{p}\|\mathbf{y}_r(p) - \gamma\mathbf{\Psi}(i)\mathbf{h}_r\|^2 \tag{4.16}$$

where if $p - W + 1 < 0$, we set $i = 0$.

SW-RLS tracking is initialized using EW-RLS algorithm with $\lambda = 1$ to get the first sliding window for $p = 0, 1, \cdots, W-1$. Then set $\hat{\mathbf{h}}_{r,u}(W-1) = \hat{\mathbf{h}}_r(W-1)$ and $\mathbf{P}_{h_r,u}(W-1) = \mathbf{P}_{h_r}(W-1)$. Mimicking [2, Problem 12.7], SW-RLS recursion (for $p = W, W+1, \cdots$) is applied independently for each r-th output via the following steps (we have omitted the subscript r since the following steps are common for every output):

1. Downdating:

$$\begin{aligned}
\mathbf{\Gamma}_d(p-1) &= \mathbf{I}_{L+1} - \gamma^2 \mathbf{\Psi}(p-W)\mathbf{P}_u(p-1)\mathbf{\Psi}^H(p-W), \\
\mathbf{G}_d(p-1) &= \gamma \mathbf{P}_u(p-1)\mathbf{\Psi}^H(p-W)\mathbf{\Gamma}_d^{-1}(p-1), \\
\hat{\mathbf{h}}_d(p-1) &= \hat{\mathbf{h}}_u(p-1) - \mathbf{G}_d(p-1)\left[\mathbf{y}(p-W) - \gamma \mathbf{\Psi}(p-W)\hat{\mathbf{h}}_u(p-1)\right], \\
\mathbf{P}_d(p-1) &= \left[\mathbf{I}_M + \gamma \mathbf{G}_d(p-1)\mathbf{\Psi}(p-W)\right]\mathbf{P}_u(p-1)
\end{aligned}$$

2. Updating:

$$\begin{aligned}
\mathbf{\Gamma}_u(p) &= \mathbf{I}_{L+1} + \gamma^2 \mathbf{\Psi}(p)\mathbf{P}_d(p-1)\mathbf{\Psi}^H(p), \\
\mathbf{G}_u(p) &= \gamma \mathbf{P}_d(p-1)\mathbf{\Psi}^H(p)\mathbf{\Gamma}_u^{-1}(p), \\
\hat{\mathbf{h}}_u(p) &= \hat{\mathbf{h}}_d(p-1) + \mathbf{G}_u(p)\left[\mathbf{y}(p) - \gamma \mathbf{\Psi}(p)\hat{\mathbf{h}}_d(p-1)\right], \\
\mathbf{P}_u(p) &= \left[\mathbf{I}_M - \gamma \mathbf{G}_u(p)\mathbf{\Psi}(p)\right]\mathbf{P}_d(p-1).
\end{aligned}$$

Now $\hat{\mathbf{h}}_{r,u}(p)$ is the estimate of $\mathbf{h}_r(p)$ based on the observations $\{\mathbf{y}_r(i)\}_{i=p-W+1}^{p}$. The channel estimates for every p-th subblock are also generated using (4.15) by setting $\hat{\mathbf{h}}_r(p) = \hat{\mathbf{h}}_{r,u}(p)$.

4.2.3 Simulation Examples

Example 1

A random time- and frequency-selective Rayleigh fading channel is considered. We assume $\mathbf{h}(n;l)$ are zero-mean, complex Gaussian, and spatially white. We take $L = 2$ (3 taps) in (4.1), and $\sigma_h^2 = E\{h_r(n;l)h_r^*(n;l)\} = 1/(L+1)$. For different l's, $\mathbf{h}(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75] (with a correction in the appendix of [63]).

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 0 to 400Hz, or the normalized Doppler spread $f_d T_s$ from 0 to 0.01 (corresponding to a maximum mobile velocity from 0 to 216km / h). 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.

The TM training scheme of [70], which is optimal for channels satisfying critically sampled CE-BEM representation, is adopted, where each subblock of equal length m_b symbols consists of an information session of m_d symbols and a succeeding training session of m_t symbols ($m_b = m_d + m_t$). We assume that each information symbol has unit power, while at every training session, we set $\gamma = \sqrt{2L+1}$ so that the average power per symbol at training sessions is equal to that of information sessions. In the simulations, we set $\gamma = \sqrt{5}$ for $m_b = 20$ ($m_d = 15$ and $m_t = 5$) or $m_b = 40$ ($m_d = 35$ and $m_t = 5$).

For CE-BEM, we select the period $T = 400$ and hence $Q = 9$ by (4.4). In Chapter 3, we see that, for the fast-fading channels, the symbol-wise approaches based on AR model (AR(1)-KF and Joint-KF in Section 3.2.2) have worse performances than block-wise or subblock-wise ones based on CE-BEM (BA-LS and SB-KF in Section 3.2.2). Here we compared the following four schemes:

1. The block-adaptive channel estimation in [70], where the transmitted symbols are segmented into consecutive blocks of T_B symbols each. Every block consists of $\bar{P}$ ($\geq Q$) subblocks as introduced in Section 2.3. We use an oversampled CE-BEM with $T_B = T/2$ for $m_b = 20$ in order to suppress spectral leakage; whereas for $m_b = 40$, since an oversampled CE-BEM is not possible to satisfy $\bar{P} \geq Q$, we take $T_B = T$. Considering the oversampled CE-BEM basis functions are not orthogonal, we apply a regularized least-square approach with a regularization parameter $\beta = 1$; so it is fair to compare with our proposed schemes. In the figures, this scheme is denoted by "BA-LS".

2. Our subblock-wise Kalman filtering in Chapter 3, which uses the first order AR model as *a priori* model for BEM coefficients. We take the AR coefficient $\alpha = 0.9944$ and 0.9667 for $m_b = 20$ and 40 respectively. In the figures, this scheme is denoted by "SB-KF".

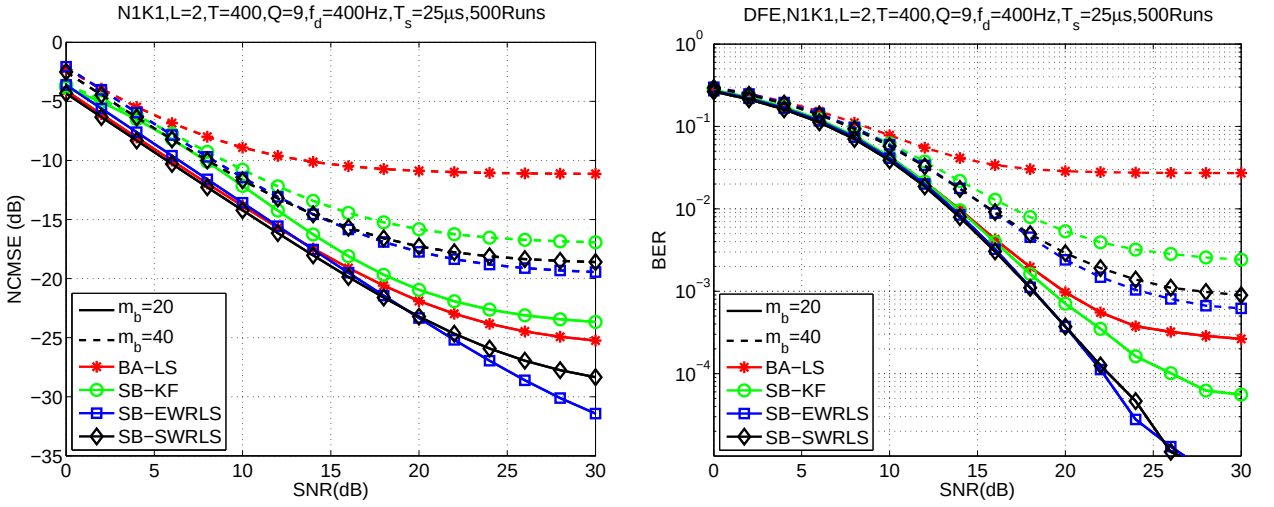
3. Proposed subblock-wise EW-RLS algorithm with $\beta = 1$. For $m_b = 20$ and 40, we take the forgetting factor $\lambda = 0.65$ and 0.5 respectively (those values were determined empirically : see the Section 4.4 for the choices). In the figures, this scheme is denoted by "SB-EWRLS".

4. Proposed subblock-wise SW-RLS algorithm with $\beta = 1$. We take $T_B = T/2 = 200$, so that for $m_b = 20$ and 40, the window size $W = 10$ and 5 respectively. In the figures, this scheme is denoted by "SB-SWRLS".

We evaluate the performances of those schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

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

where $\mathbf{h}^{(i)}(n;l)$ is the true channel and $\hat{\mathbf{h}}^{(i)}(n;l)$ is the estimated channel at the i-th Monte Carlo run, among total M_r runs, and T_N is the total observation length in symbols. For each of the above four schemes, the MMSE-DFE described in Section 2.4.2 [40] is employed at the receiver, using the obtained channel estimates, with $l_f = 8$, $l_b = 2$ and the delay $d = 5$ symbols. In each run, a symbol sequence of length, $T_N = 4000$ for each user is modulated by QPSK and the first 200 symbols are discarded in evaluations. All the simulation results are based on 500 runs.

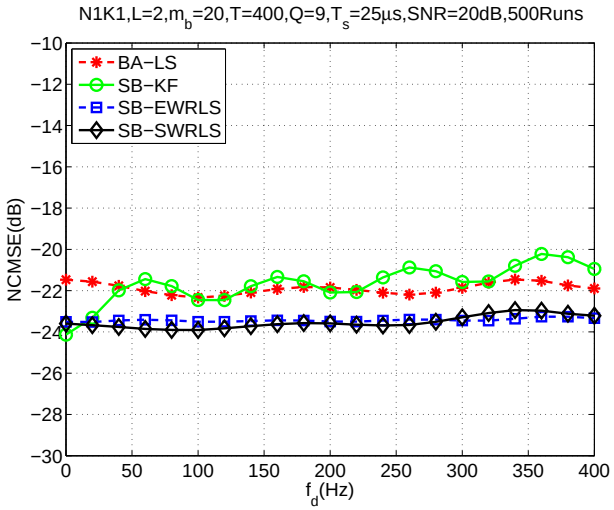


(a) NCMSE vs SNR

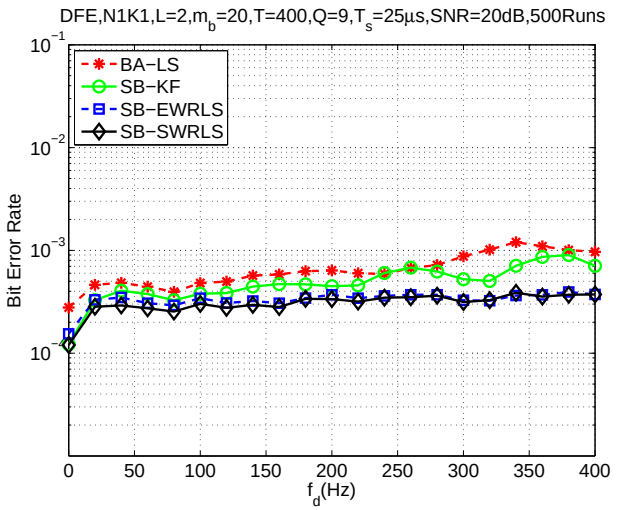
(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and delay $d = 5$

Figure 4.1: Subblock-wise RLS channel estimation: performance comparison for SNR's under $N = K = 1, f_dT_s = 0.01$.

In Fig. 4.1, the performances of the four schemes are compared for different SNR's under the normalized Doppler spread $f_dT_s = 0.01$. Since we use only training session for channel estimation, the smaller subblock with more training ($m_b = 20$) results in better performance than the larger subblock ($m_b = 40$). Since the subblock-wise approaches update every subblock, they have superior performance to the block-wise estimation scheme in [70]. Assuming no *a priori* models of the BEM coefficients, the two proposed subblock-wise RLS tracking schemes (SB-EWRLS and SB-SWRLS) outperform the subblock-wise Kalman



(a) NCMSE vs f_dT_s



(b) BER vs f_dT_s, with DFE of $l_f = 8, l_b = 2$ and delay $d = 5$

Figure 4.2: Subblock-wise RLS channel estimation: performance comparison for Doppler spread under $N = K = 1, \text{SNR} = 20\,\text{dB}$.

tracking scheme (SB-KF) we proposed earlier in Chapter 3 and they have similar NCMSE and BER performances. Noting that when $m_b = 20$, the subblock-wise Kalman tracking scheme is better than the block-wise LS scheme when BER is the performance criterion and is worse when average channel MSE is the performance criterion, superior average channel MSE does not necessarily translate into superior average BER since a monotone relationship between average MSE (averaged over all taps and runs) and average BER does not necessarily exist. [To further investigate this "counter-intuitive" behavior, we plotted histograms (not shown here) of the channel MSE and the BER for the two schemes (BA-LS and SB-KF) with $m_b = 20$ for a specific case corresponding to SNR $= 28\,\text{dB}$ in Fig. 4.1a and 4.1b. The histograms show that the channel MSE of the BA-LS scheme has a spread around the mean value that is more than twice that for the SB-KF scheme. This, in turn, translates into a larger BER for the BA-LS scheme in quite a few runs, compared to the SB-KF scheme.]

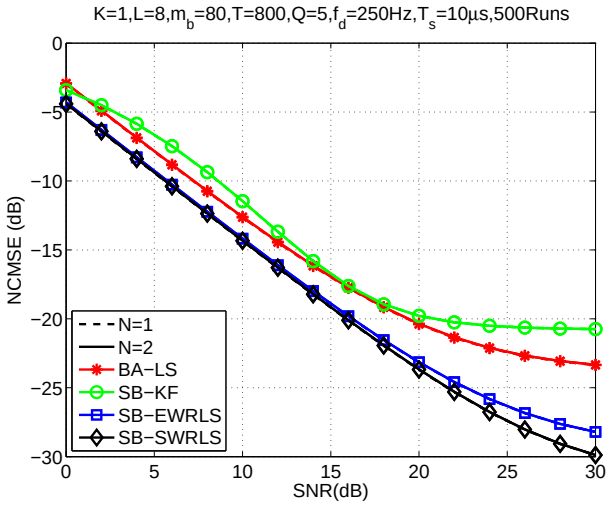
In Fig. 4.2, we compare the same schemes over different Doppler spread f_d's under SNR $= 20\,\text{dB}$, the subblock size $m_b = 20$. Since all these schemes are based on CE-BEM, their performances are not sensitive to the actual Doppler spread. However, the proposed SB-EWRLS and SB-SWRLS have more stable and reliable performance than BA-LS and SB-KF over all the range of Doppler spread. Note that an arbitrary model for channel is not necessarily suited to a real channel. The SB-KF assume the BEM coefficients follow the first-order AR process, which may introduce additional errors in channel estimation.

Example 2

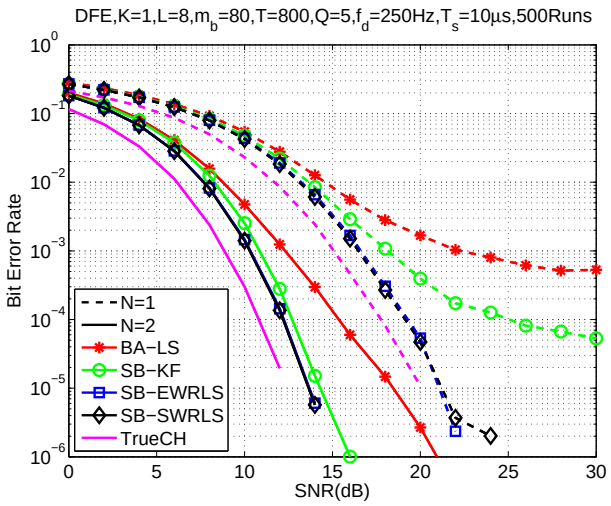
The random Rayleigh fading channel in this example is as in Example 1, except that we change the channel length L from 2 to 8 leading to 9 taps, and the power delay profile is exponential satisfying $E\{|h_r(n:l)|^2\} = \sigma_h^2(l) = ae^{-l/L}$ where the constant a is pick to satisfy $\sum_{l=0}^{L} \sigma_h^2(l) = 1$. We consider a communication system with carrier frequency of 2GHz, data rate of 0.1MBd (mega-Bauds), therefore $T_s = 10\,\mu$s, and a varying Doppler spread $f_d = 250\,$Hz, or the normalized Doppler spread $f_d T_s = 0.0025$ (corresponding to a maximum mobile velocity of 135km / h). The training session is described by (2.22) of length $m_t = 2L + 1 = 17$ symbols with $\gamma = \sqrt{2L+1}$ so that the average symbol power of training sessions is equal to that of information sessions. We select the period of the CE-BEM $T = 800$ symbols, and hence $Q = 5$ by (4.4) and a block size $T_B = 400$ for oversampled CE-BEM.

Five estimation and tracking schemes are compared: block-adaptive channel estimation in [70] using the regularized LS approach ("BA-LS"), the subblock-wise Kalman tracking ("SB-KF") introduced in Chapter 3, the proposed subblock-wise EW-RLS tracking ("SB-EWRLS") and SW-RLS tracking ("SB-SWRLS"). With the subblock size $m_b = 80$, we take the AR coefficient $\alpha = 0.993$ for SB-KF, the forgetting factor $\lambda = 0.5$ for SB-EWRLS and the window size $W = T_B/m_b = 5$ for SB-SWRLS. [See Section 3.4 and 4.4 for the choices of the AR coefficient α and the forgetting factor λ respectively.] We also consider the perfect channel estimates("TrueCH"). The BER's are evaluated by employing the MMSE-DFE with $l_f = 14$, $l_b = 8$ and the delay $d = 10$, using the channel estimates obtained by each scheme.

In Figs. 4.3, the performances of the five schemes are compared for different SNR's. Due to the assumption that the channel is spatially uncorrelated for different outputs, the NCMSE curves for $N = 1$ and 2 coincide with each other. The three subblock-wise schemes (SB-KF, SB-EWRLS and SB-SWRLS) outperform the block-wise LS estimation (BA-LS) in [70]. Definitely, the proposed two finite-memory RLS schemes, SB-EWRLS and SB-SWRLS, have similar NCMSE and BER performance, and they are slightly better than the subblock-wise Kalman tracking scheme since the latter scheme assumes the BEM coefficients follow a first-order AR model and this may introduce additional modeling errors. Note that SB-EWRLS and SB-SWRLS schemes track the channel subblock-by-subblock using the regularized least-square solution with no *a priori* models for the BEM coefficients.



(a) NCMSE vs SNR



(b) BER vs SNR, with DFE of $l_f = 14, l_b = 8$ and delay $d = 10$

Figure 4.3: Subblock-wise RLS channel estimation: performance comparison for SNR's under $K = 1, f_d T_s = 0.0025, m_b = 80$.

4.3 Subblock-wise RLS MIMO Channel Estimation using CE-BEM

4.3.1 RLS Tracking for MIMO Channel

Consider a doubly-selective multi-input multi-output (MIMO), finite impulse response (FIR) linear channel with K inputs and N outputs. Let $\{s_k(n)\}$ denote k-th user's information sequence that is input to the time-varying channel with discrete-time response $\{\mathbf{h}_k(n;l)\}$ (channel response for the k-th user at time instance n to a unit input at time instance $n-l$), assuming $\{s_k(n)\}$ is mutually independent and identically distributed (i.i.d.) with zero mean and variance $E\{s_k(n)\, s_k^*(n)\} = \sigma_{s_k}^2 = \sigma_s^2$ for $k = 1, 2, \cdots, K$. Then the symbol-rate noisy N-column channel output vector is given by $(n = 0, 1, \ldots)$

$$\mathbf{y}(n) = \sum_{k=1}^{K} \sum_{l=0}^{L} \mathbf{h}_k(n;l)\, s_k(n-l) + \mathbf{v}(n) \tag{4.17}$$

where the N-column vector $\mathbf{v}(n)$ is zero-mean, white, uncorrelated with $s_k(n)$, complex Gaussian noise, with the autocorrelation $E\{\mathbf{v}(n+\tau)\, \mathbf{v}^H(n)\} = \sigma_v^2 \mathbf{I}_N \delta(\tau)$. We assume that $\{\mathbf{h}_k(n;l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [60],

independent for different k's. Define

$$\mathbf{s}(n) := \begin{bmatrix} s_1(n) & s_2(n) & \cdots & s_K(n) \end{bmatrix}^T$$
$$\mathbf{h}(n;l) := \begin{bmatrix} \mathbf{h}_1(n;l) & \mathbf{h}_2(n;l) & \cdots & \mathbf{h}_K(n;l) \end{bmatrix}.$$

and we may rewrite (4.17) as

$$\mathbf{y}(n) = \sum_{l=0}^{L} \mathbf{h}(n;l)\, \mathbf{s}(n-l) + \mathbf{v}(n). \tag{4.18}$$

In CE-BEM [11, 14, 70], over the i-th block consisting of an observation window of T_B symbols, the channel is represented as ($n = (i-1)T_B, (i-1)T_B + 1, \cdots, iT_B - 1$ and $l = 0, 1, \cdots, L$)

$$\mathbf{h}_k(n;l) = \sum_{q=1}^{Q} \mathbf{h}_{k,q}(l) e^{j\omega_q n}, \tag{4.19}$$

where $\mathbf{h}_{k,q}(l)$ is the N-column time-invariant BEM coefficient vector for k-th user and one chooses (Λ is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{4.20}$$
$$Q \geq 2\lceil f_d T T_s \rceil + 1, \tag{4.21}$$
$$\omega_q := \frac{2\pi}{T}\left[q - (Q+1)/2\right], \quad q = 1, 2, \cdots, Q \tag{4.22}$$
$$L := \lfloor \tau_d / T_s \rfloor, \tag{4.23}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration.

Now extend the proposed RLS subblock-wise approach in Section 4.2 to MIMO systems. Let $h_{rk,q}(l)$ denote the $r-$th component of the column $\mathbf{h}_{k,q}(l)$. Stack the BEM coefficients in (4.19) into vectors as

$$\mathbf{h}_{rk}^{(l)} := \begin{bmatrix} h_{rk,1}(l) & h_{rk,2}(l) & \cdots & h_{rk,Q}(l) \end{bmatrix}^T \tag{4.24}$$
$$\mathbf{h}_{rk} := \begin{bmatrix} \mathbf{h}_{rk}^{(0)\,T} & \mathbf{h}_{rk}^{(1)\,T} & \cdots & \mathbf{h}_{rk}^{(L)\,T} \end{bmatrix}^T \tag{4.25}$$
$$\mathbf{h} := \begin{bmatrix} \mathbf{h}_{11}^T & \cdots & \mathbf{h}_{N1}^T & \cdots & \mathbf{h}_{1K}^T & \cdots & \mathbf{h}_{NK}^T \end{bmatrix}^T \tag{4.26}$$

of size Q, $M := Q(L+1)$ and $NKQ(L+1)$, respectively. The coefficient vectors in (4.24), (4.25) and (4.26) of the p-th subblock will be denoted by $\mathbf{h}_{rk}^{(l)}(p)$, $\mathbf{h}_{rk}(p)$, and $\mathbf{h}(p)$.

Based on CE-BEM given by (4.19), define (where $\mathbf{s}(n)$ is K-column vector)

$$\begin{aligned}\boldsymbol{\mathcal{E}}(n) &:= \left[e^{-j\omega_1 n} \quad e^{-j\omega_2 n} \quad \cdots \quad e^{-j\omega_Q n}\right]^T, \\ \mathbf{S}(n) &:= \left[\mathbf{s}^T(n) \quad \mathbf{s}^T(n-1) \quad \cdots \quad \mathbf{s}^T(n-L)\right]^T,\end{aligned}$$

and then the received signal can be written as

$$\mathbf{y}(n) = \left[\mathbf{S}(n) \otimes \mathbf{I}_N\right]^T \left[\mathbf{I}_{NK(L+1)} \otimes \boldsymbol{\mathcal{E}}(n)\right]^H \mathbf{h}(p) + \mathbf{v}(n) \tag{4.27}$$

when the p-th subblock is being received at time n, by (4.18), (4.19)-(4.23) and (4.24)-(4.26).

We will employ the time-multiplexed training scheme in Section 2.3 [30]. Then by design, a given user's training does not affect that of any other user during training session. Hence the received signal between k-th input and r-th output can be written as (assuming timing synchronization)

$$y_r(n_{k,p}+l) = \gamma h_{rk}(n_{k,p}+l; l) + v_r(n_{k,p}+l) \tag{4.28}$$

for $l = 0, 1, \cdots, L$, $k = 1, 2, \cdots, K$ and $r = 1, 2, \cdots, N$. The (unknown) MIMO channel between K inputs and N outputs can be tracked independently one by one. Using (4.28), in order to track every (sub-)channel between K inputs and N outputs respectively, we can simplify (4.27) at time $n_{k,p}+l$ as ($p = 0, 1, \cdots$ and $l = 0, 1, \cdots, L$)

$$y_r(n_{k,p}+l) = \gamma \boldsymbol{\mathcal{E}}^H(n_{k,p}+l)\, \mathbf{h}_{rk}^{(l)}(p) + v_r(n_{k,p}+l) \tag{4.29}$$

We intend to use only training sessions for subblock-wise channel tracking regardless of data sessions. Further define

$$\begin{aligned}\tilde{\mathbf{y}}_r(p) &:= \left[y_r(n_{k,p}) \quad y_r(n_{k,p}+1) \quad \cdots \quad y_r(n_{k,p}+L)\right]^T, \\ \mathbf{v}_r(p) &:= \left[v_r(n_{k,p}) \quad v_r(n_{k,p}+1) \quad \cdots \quad v_r(n_{k,p}+L)\right]^T,\end{aligned}$$

and

$$\boldsymbol{\Psi}(p) := \begin{bmatrix} \boldsymbol{\mathcal{E}}(n_{k,p}) & & & \\ & \boldsymbol{\mathcal{E}}(n_{k,p}+1) & & \\ & & \ddots & \\ & & & \boldsymbol{\mathcal{E}}(n_{k,p}+L) \end{bmatrix}^H.$$

Then by (4.29),

$$\tilde{\mathbf{y}}_r(p) = \gamma \boldsymbol{\Psi}(p)\, \mathbf{h}_{rk}(p) + \mathbf{v}_r(p). \tag{4.30}$$

Given an $(L+1)\times 1$ measurement vector $\tilde{\mathbf{y}}_r(p)$, a training impulse γ and an $(L+1)\times M$ basis function matrix $\mathbf{\Psi}(p)$ in (4.30), we can apply exponentially-weighted regularized RLS (EW-RLS) algorithm [2, Chapter 12] to track an unknown $M\times 1$ BEM coefficient vector $\mathbf{h}_{rk}(p)$ between r-th receiver and k-th user. Choose $\mathbf{h}_{rk}(p)$ to minimize the cost function

$$\lambda^{p+1}\beta\left\|\mathbf{h}_{rk}\right\|^2+\sum_{i=0}^{p}\lambda^{p-i}\left\|\tilde{\mathbf{y}}_r(i)-\gamma\mathbf{\Psi}(i)\mathbf{h}_{rk}\right\|^2 \tag{4.31}$$

where $\beta>0$ is a regularization parameter and $0<\lambda<1$ is the forgetting factor.

EW-RLS tracking is initialized with $\hat{\mathbf{h}}_{rk}(-1)=\mathbf{0}_{M\times 1}$ and $\mathbf{P}_{h_{rk}}(-1)=\beta^{-1}\mathbf{I}_M$. Mimicking [2, Algorithm 12.3.1], EW-RLS recursion (for $p=0,1,\cdots$) is applied one-by-one for each pair between r-th receiver and k-th user independently via the same steps in Section 4.2.1. After RLS recursion for every p, we generate the channel for the p-th subblock by the estimate $\hat{\mathbf{h}}(p)$ via the CE-BEM (4.19) as

$$\hat{h}_{rk}(n;l)=\boldsymbol{\mathcal{E}}^H(n)\,\hat{\mathbf{h}}_{rk}^{(l)}(p) \tag{4.32}$$

for $n=pm_b,pm_b+1,\cdots,(p+1)m_b-1$. The definition of $\hat{\mathbf{h}}_{rk}^{(l)}(p)$ is similar to (4.24). Thus we obtain the entire $N\times K$ channel matrix for the current subblock.

Similarly, the cost function for SW-RLS is given by

$$\beta\left\|\mathbf{h}_{rk}\right\|^2+\sum_{i=p-W+1}^{p}\left\|\tilde{\mathbf{y}}_r(p)-\gamma\mathbf{\Psi}(i)\mathbf{h}_{rk}\right\|^2 \tag{4.33}$$

where if $p-W+1<0$, we set $i=0$.

SW-RLS tracking is initialized using EW-RLS algorithm with $\lambda=1$ to get the first sliding window for $p=0,1,\cdots,W-1$. Then set $\hat{\mathbf{h}}_{rk,u}(W-1)=\hat{\mathbf{h}}_{rk}(W-1)$ and $\mathbf{P}_{h_{rk},u}(W-1)=\mathbf{P}_{h_{rk}}(W-1)$. Mimicking [2, Problem 12.7], SW-RLS recursion (for $p=W,W+1,\cdots$) has the same algorithm in Section 4.2.2 since it is applied for each pair rk independently. Now $\hat{\mathbf{h}}_u(p)$ is the estimate of $\mathbf{h}(p)$ based on the observations $\{\mathbf{y}(i)\}_{i=p-W+1}^{p}$. The channel estimates for every p-th subblock are also generated using (4.32) by setting $\hat{\mathbf{h}}(p)=\hat{\mathbf{h}}_u(p)$.

Computational Complexity

We compare computational complexity of each scheme using flops counting for the channel estimation algorithm of each simulation program. We use the same flops counting for multiplication and inverse as in Section 3.3.1. The detailed flops for one iteration of EW-RLS tracking algorithm is shown in Table 4.1 and the overall flop counts for each scheme are compared in Table 4.2 ($M=Q(L+1),\bar{P}=T_B/m_b,L_p=L+1$). We also calculate

the relative complexity for specific examples in Table 4.3. Note that subblock-wise EW-RLS scheme appears to provide a good complexity-vs-performance trade-off at higher SNR's at the cost of an increase of flop counts whereas at lower SNR values such an increase in flop count may not warrant the use of subblock-wise tracking schemes.

Table 4.1: Subblock-wise EW-RLS channel estimation: flop count over one subblock of m_b symbols.

Operation	Number of flops
$\mathbf{\Gamma}(p) = \lambda \mathbf{I}_{L+1} + \gamma^2 \mathbf{\Psi}(p)\mathbf{P}(p-1)\mathbf{\Psi}^H(p)$	$M^2 L_p + M(L_p^2 + L_p) + L_p^2$
$\mathbf{G}(p) = \gamma \mathbf{P}(p-1)\mathbf{\Psi}^H(p)\mathbf{\Gamma}^{-1}(p)$	$M^2(L_p+1) + ML_p^2 + L_p^3/3$
$\hat{\mathbf{h}}(p) = \hat{\mathbf{h}}(p-1) + \mathbf{G}(p)\left[\mathbf{y}(p) - \gamma \mathbf{\Psi}(p)\hat{\mathbf{h}}(p-1)\right]$	$M(3L_p)$
$\mathbf{P}(p) = \lambda^{-1}\left[\mathbf{I}_M - \gamma \mathbf{G}(p)\mathbf{\Psi}(p)\right]\mathbf{P}(p-1)$	$M^3 + M^2(L_p+1) + ML_p$
Total : $M^3 + M^2(3L_p+2) + M(2L_p^2+5L_p) + L_p^3/3 + L_p^2$	

Table 4.2: Block-wise least-square and subblock-wise RLS: flop count for channel estimation over one block of T_B symbols.

Scheme	Number of flops
BA-LS	$M^3/3 + M^2(2\bar{P}L_p + 1) + M(\bar{P}L_p + 1)$
EW-RLS	$\bar{P}\left\{M^3 + M^2(3L_p+2) + M(2L_p^2+5L_p) + L_p^3/3 + L_p^2\right\}$
SW-RLS	$2\bar{P}\left\{M^3 + M^2(3L_p+1) + M(2L_p^2+5L_p) + L_p^3/3\right\}$

4.3.2 Simulation Examples

Example 1

A random time- and frequency-selective Rayleigh fading MIMO channel is considered. We assume $\mathbf{h}(n;l)$ are zero-mean, complex Gaussian, and spatially white. We take $L = 2$ (3 taps) in (4.18), and $\sigma_h^2 = E\{h_{rk}(n;l)h_{rk}^*(n;l)\} = 1/(L+1)$. For different l's, $\mathbf{h}(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75] (with a correction in the appendix of [63]).

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 0 to 400Hz, or the normalized Doppler spread $f_d T_s$ from 0 to 0.01 (corresponding to a maximum mobile velocity from 0 to 216km / h). The additive noise is zero-mean complex

Table 4.3: Block-wise least-square and subblock-wise RLS: comparative flop count for channel estimation over one block of T_B symbols.

Example 1: $T = 400, Q = 9, T_B = 200, m_b = 20, \bar{P} = 10, L = 2$			
Number of flops	BA-LS	EW-RLS	SW-RLS
actual	51,867	286,110	557,460
relative	1.00	5.52	10.75

Example 2: $T = 800, Q = 5, T_B = 400, m_b = 80, \bar{P} = 5, L = 8$			
Number of flops	BA-LS	EW-RLS	SW-RLS
actual	216,720	797,445	1,573,830
relative	1.00	3.68	7.26

white Gaussian. The (receiver) SNR refers to the average energy per symbol over one-sided noise spectral density.

The TM training scheme of [30], which is optimal for channels satisfying critically sampled CE-BEM representation, is adopted, where each subblock of equal length m_b symbols consists of an information session of m_d symbols and a succeeding training session of m_t symbols ($m_b = m_d + m_t$). We assume that each information symbol has unit power, while at every training session, we set $\gamma = \sqrt{K(L+1)+L}$ so that the average power per symbol at training sessions is equal to that of information sessions. In the simulations, we consider a simple two-receiver and two-user scenario, i.e., $N = 2, K = 2$ with the same transmitted power. We set $m_b = 20$ or 40 with $m_t = 8$ and $\gamma = \sqrt{8}$ for every user following the TM training scheme. For CE-BEM, we select the period $T = 400$ and hence $Q = 9$ by (4.21).

We compared the following four schemes:

1. The block-adaptive channel estimation in [30], where the transmitted symbols are segmented into consecutive blocks of T_B symbols each. Every block consists of $\bar{P}$ ($\geq Q$) subblocks as in Section 2.3. We use an oversampled CE-BEM with $T_B = T/2$ for $m_b = 20$ since the over-sampled CE-BEM approximates WSSUS channels much better than critically sampled CE-BEM [14]; whereas for $m_b = 40$, since an oversampled CE-BEM is not possible to satisfy $\bar{P} \geq Q$, we take $T_B = T$. Considering the oversampled CE-BEM basis functions are not orthogonal, we apply a regularized least-square approach with a regularization parameter $\beta = 1$; so it is fair to compare with our proposed schemes. In the figures, this scheme is denoted by "BA-LS".

2. Our subblock-wise Kalman filtering in Chapter 3, which uses the first order AR model as *a priori* model for BEM coefficients. We take the AR coefficient $\alpha = 0.9944$ and 0.9667 for $m_b = 20$ and 40 respectively. In the figures, this scheme is denoted by "SB-KF".

3. Proposed subblock-wise EW-RLS algorithm with $\beta = 1$. For $m_b = 20$ and 40, we take the forgetting factor $\lambda = 0.65$ and 0.5 respectively (those values were determined empirically : see the Section 4.4 for the choices). In the figures, this scheme is denoted by "SB-EWRLS".

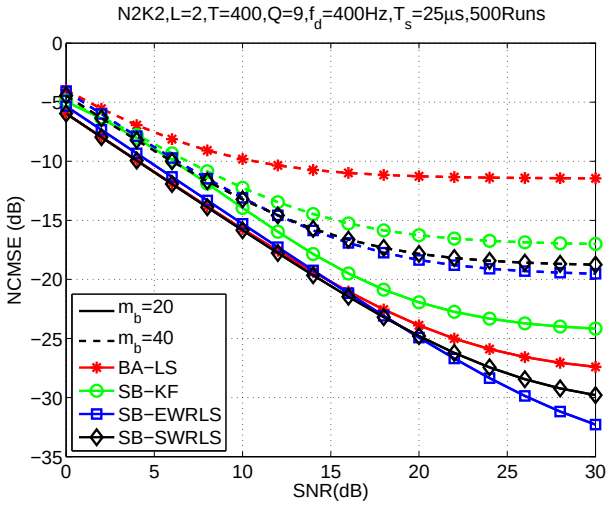
4. Proposed subblock-wise SW-RLS algorithm with $\beta = 1$. We take $T_B = T/2 = 200$, so that for $m_b = 20$ and 40, the window size $W = 10$ and 5 respectively. In the figures, this scheme is denoted by "SB-SWRLS".

We evaluate the performances of those schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

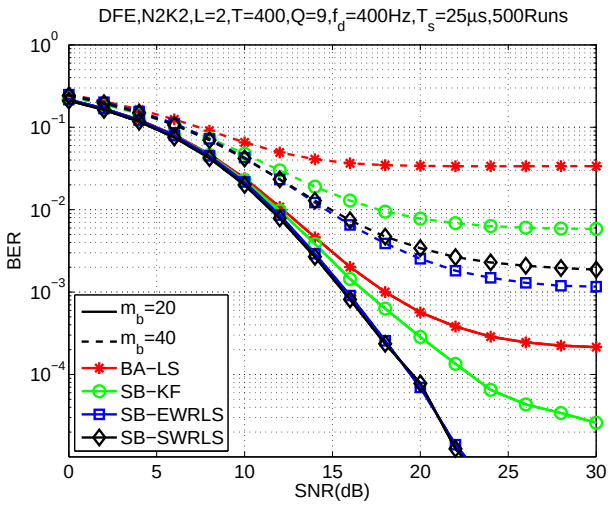
$$\text{NCMSE} := \frac{\sum_{i=1}^{M_r} \sum_{n=0}^{T_N-1} \sum_{l=0}^{L} \left\| \hat{\mathbf{h}}^{(i)}(n;l) - \mathbf{h}^{(i)}(n;l) \right\|^2}{\sum_{i=1}^{M_r} \sum_{n=0}^{T_N-1} \sum_{l=0}^{L} \left\| \mathbf{h}^{(i)}(n;l) \right\|^2}$$

where $\mathbf{h}^{(i)}(n;l)$ is the true channel and $\hat{\mathbf{h}}^{(i)}(n;l)$ is the estimated channel at the i-th Monte Carlo run, among total M_r runs, and T_N is the total observation length in symbols. For each of the above four schemes, the MMSE-DFE described in Section 2.4.2 [40] is employed at the receiver, using the obtained channel estimates, with $l_f = 8$, $l_b = 2$, and the delay $d = 5$ symbols. In each run, a symbol sequence of length, $T_N = 4000$ for each user is modulated by QPSK and the first 200 symbols are discarded in evaluations. All the simulation results are based on 500 runs.

In Fig. 4.4, the performances of the four schemes are compared for different SNR's with the normalized Doppler spread $f_d T_s = 0.01$ fixed. Since we use only training session for channel estimation, the smaller subblock with more training ($m_b = 20$) results in better performance than the larger subblock ($m_b = 40$). All the subblock-wise schemes including Kalman tracking outperform the block-wise one in [30] and the proposed SB-EWRLS and SB-SWRLS approach have superior performances to the subblock-wise Kalman tracking that has an arbitrary model for BEM coefficients (e.g., AR model) and this may incur modeling errors. The proposed subblock-wise EW-RLS and SW-RLS schemes track the channel subblock-by-subblock using the regularized least-square solution, assuming no *a priori* models for the BEM coefficients.



(a) NCMSE vs SNR



(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and delay $d = 5$

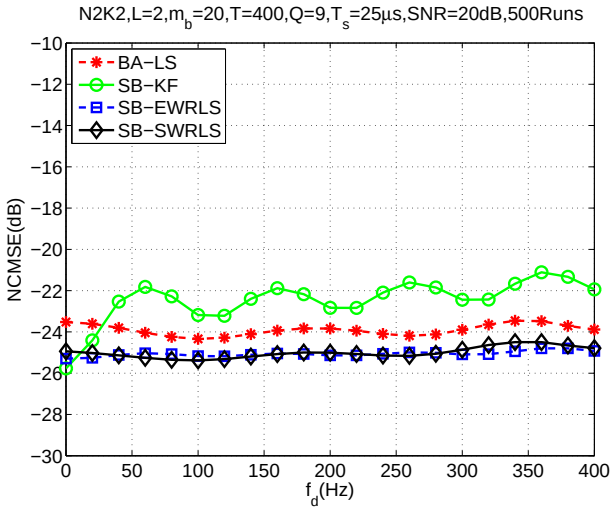
Figure 4.4: Subblock-wise RLS MIMO channel estimation: performance comparison for SNR's under $N = K = 2, f_d T_s = 0.01$.

In Fig. 4.5, we compare the same schemes with the subblock size $m_b = 20$ over the wide range of Doppler spread f_d's under SNR $= 20\,\text{dB}$. It is cleat that the subblock-wise RLS approaches have more stable and better performance than BA-LS and SB-KF over all the range of Doppler spread.

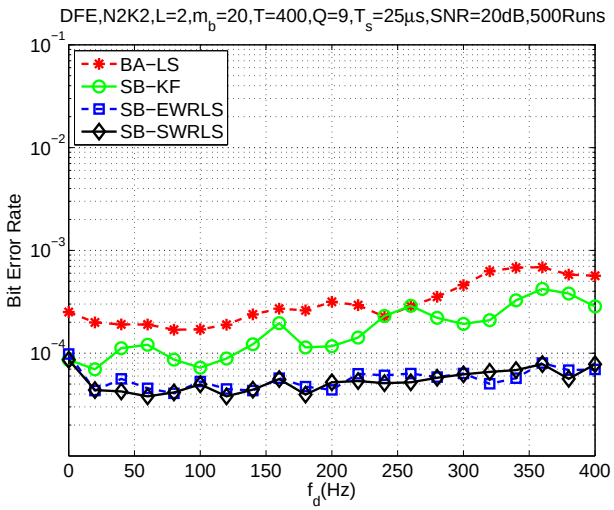
Example 2

The random Rayleigh fading MIMO channel in this example is as in Example 1, except that we change the channel length L from 2 to 8 leading to 9 taps, and the power delay profile is exponential satisfying $E\{|h_{rk}(n:l)|^2\} = \sigma_h^2(l) = ae^{-l/L}$ where the constant a is pick to satisfy $\sum_{l=0}^{L} \sigma_h^2(l) = 1$. We consider a communication system with carrier frequency of 2GHz, data rate of 0.1MBd (mega-Bauds), therefore $T_s = 10\,\mu\text{s}$, and a varying Doppler spread $f_d = 250\,\text{Hz}$, or the normalized Doppler spread $f_d T_s = 0.0025$ (corresponding to a maximum mobile velocity of 135km / h). The training session is described by (2.24) of length $m_t = K(L+1) + L = 26$ symbols with $\gamma = \sqrt{K(L+1)+L}$ so that the average symbol power of training sessions is equal to that of information sessions. We select the period of the CE-BEM $T = 800$ symbols, and hence $Q = 5$ by (4.21) and a block size $T_B = 400$ for oversampled CE-BEM.

Five estimation and tracking schemes are compared: block-adaptive channel estimation in [30] using the regularized LS approach ("BA-LS"), the subblock-wise Kalman tracking



(a) NCMSE vs f_dT_s

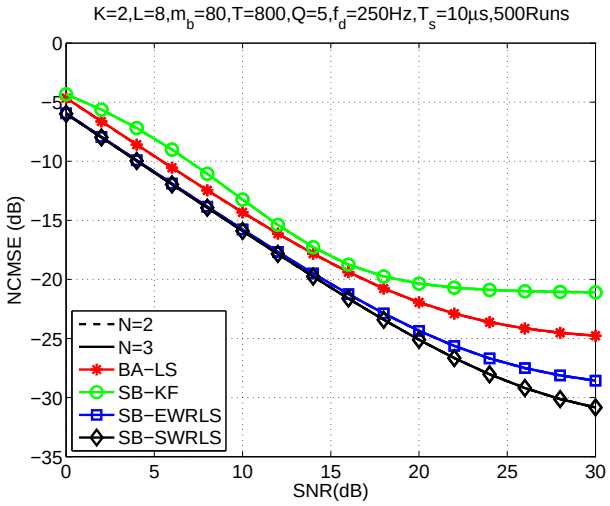


(b) BER vs f_dT_s, with DFE of $l_f = 8, l_b = 2$ and delay $d = 5$

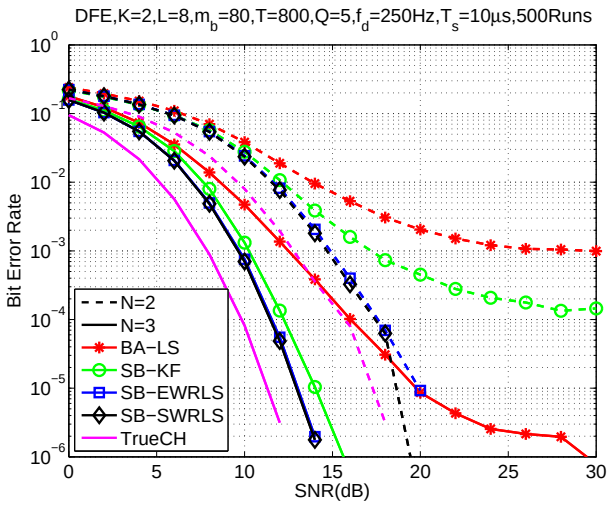
Figure 4.5: Subblock-wise RLS MIMO channel estimation: performance comparison for Doppler spread under $N = K = 2, \text{SNR} = 20\,\text{dB}$.

("SB-KF") introduced in Chapter 3, the proposed subblock-wise EW-RLS tracking ("SB-EWRLS") and SW-RLS tracking ("SB-SWRLS"). With the subblock size $m_b = 80$, we take the AR coefficient $\alpha = 0.993$ for SB-KF, the forgetting factor $\lambda = 0.5$ for SB-EWRLS and the window size $W = T_B/m_b = 5$ for SB-SWRLS. [See Section 3.4 and 4.4 for the choices of the AR coefficient α and the forgetting factor λ respectively.] We also consider the perfect channel estimates ("TrueCH"). The BER's are evaluated by employing the MMSE-DFE with $l_f = 14$, $l_b = 8$ and the delay $d = 10$, using the channel estimates obtained by each scheme.

In Figs. 4.6, the performances of the five schemes are compared for different SNR's. The NCMSE curves for $N = 2$ and 3 coincide with each other since we assume the channel is spatially uncorrelated for different outputs. The subblock-wise Kalman filtering and RLS tracking schemes outperform the block-wise LS estimation in [30]. In particular, the proposed two finite-memory RLS schemes, SB-EWRLS and SB-SWRLS, have better performances in NCMSE and BER than the subblock-wise Kalman tracking scheme. Note that SB-KF assumes the BEM coefficients follow a first-order AR process, which is not necessarily true in the real channel environment. This AR modeling for the BEM coefficients may introduce the modeling errors in channel estimation. The subblock-wise EW-RLS and SW-RLS schemes update the channel estimates subblock-by-subblock using the regularized least-square solution without any *a priori* models, which makes the channel tracking more practical and effective.



(a) NCMSE vs SNR



(b) BER vs SNR, with DFE of $l_f = 14, l_b = 8$ and delay $d = 10$

Figure 4.6: Subblock-wise RLS MIMO channel estimation: performance comparison for SNR's under $K = 2$, $f_d T_s = 0.0025$, $m_b = 80$.

4.4 Forgetting Factor λ for Subblock-wise EW-RLS Tracking

Here we analyze the theoretical choices of λ for subblock-wise channel estimation using EW-RLS algorithm in this chapter. Consider only one pair of channel $\mathbf{h}_{rk}$ between r-th input and k-th output, since we track the channel of every pair independently for N receivers and K users by design of time-multiplexed (TM) training in Section 2.3 [30]. The cost function for EW-RLS algorithm is given as (4.14), which we can rewrite for "large" ($p \to \infty$) as

$$\mathcal{C} = \sum_{i=0}^{\infty} \lambda^i \left\| \tilde{\mathbf{y}}_r(p-i) - \gamma \mathbf{\Psi}(p-i)\mathbf{h}_{rk} \right\|^2 . \tag{4.34}$$

Let $\bar{L}$ the number of subblocks in which we would like to have the estimate of BEM coefficient $\mathbf{h}_{rk}$. It is clear that $\mathbf{\Psi}(i)$ is periodic with period T/m_b as CE-BEM is periodic with period T, that is $\mathbf{\Psi}(p \pm jT/m_b) = \mathbf{\Psi}(p)$ for any integer j. In practice, we would like to have the memory length (in symbols) to be less than the model period (recall that the channel is by no means periodic) so that there are no deleterious effects due to the use of (4.2) for all time, i.e., $m_b \bar{L} \leq T$ in order to avoid this periodicity. Let us pick $\bar{L} = T/m_b$. What other restriction should we impose on $\bar{L}$?

A least-square solution for $\mathbf{h}_{rk}$ to minimize $\mathcal{C}$ (with $p \to \infty$) is given by [50, p. 796]

$$\hat{\mathbf{h}}_{rk} = \mathbf{A}^{-1}\mathbf{B}, \tag{4.35}$$

where

$$\mathbf{A} = \sum_{i=0}^{\infty} \lambda^i \gamma^2 \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i), \tag{4.36}$$

$$\mathbf{B} = \sum_{i=0}^{\infty} \lambda^i \gamma \mathbf{\Psi}^H(p-i)\tilde{\mathbf{y}}_r(p-i). \tag{4.37}$$

In order to analyze the behavior of $\mathbf{h}_{rk}$ we need a model for $\tilde{\mathbf{y}}_r(i)$ for all subblocks, $i = p, p-1, \cdots$. To this end, for the analysis presented in this section, we assume the following "simplified" model (recall (4.13)):

$$\tilde{\mathbf{y}}_r(p-i) = \gamma \mathbf{\Psi}(p-i)\mathbf{h}_{rk}^{(tr)}(p-i) + \mathbf{v}_r(p-i), \tag{4.38}$$

where $\mathbf{h}_{rk}^{(tr)}(p-i)$ is the "true" BEM coefficient vector satisfying ($m = 1, 2, \cdots$)

$$\mathbf{h}_{rk}^{(tr)}(p-i) := \mathbf{h}_{rk,m}^{(tr)} \text{ for } (m-1)\bar{L} \leq p-i < m\bar{L}. \tag{4.39}$$

That is, there exist some true BEM parameters that are "fixed" over one BEM period of T symbols (therefore, $\bar{L} = T/m_b$ subblocks), and are allowed to change over non-overlapping periods. In this set-up, we are trying to estimate the most recent true BEM coefficient vector $\mathbf{h}_{rk,1}^{(tr)}$ via $\hat{\mathbf{h}}_{rk}$ without it being unduly influenced by $\mathbf{h}_{rk,m}^{(tr)}, m \geq 2$.

Using the periodicity, $\mathbf{\Psi}(p-(i+(m-1)\bar{L})) = \mathbf{\Psi}(p-i)$ for $i = 0, 1, \cdots, \bar{L}-1$ and $m = 1, 2, \cdots$, we can rewrite (4.36) as (for $\lambda < 1$)

$$\begin{aligned}\mathbf{A} &= \gamma^2 \sum_{m=1}^{\infty} \lambda^{(m-1)\bar{L}} \left[\sum_{i=0}^{\bar{L}-1} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)\right] \\ &= \frac{\gamma^2}{1-\lambda^{\bar{L}}} \sum_{i=0}^{\bar{L}-1} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i). \end{aligned} \tag{4.40}$$

Similarly, using (4.38) and (4.39), after some manipulations, we obtain

$$\begin{aligned}
\mathbf{B} &= \gamma \sum_{i=0}^{\infty} \lambda^i \mathbf{\Psi}^H(p-i) \left[\gamma \mathbf{\Psi}(p-i)\mathbf{h}_{rk}^{(tr)}(p-i) + \mathbf{v}_r(p-i)\right] \\
&= \gamma^2 \sum_{i=0}^{\infty} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)\mathbf{h}_{rk}^{(tr)}(p-i) + \gamma \sum_{i=0}^{\infty} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{v}_r(p-i) \\
&= \gamma^2 \sum_{m=1}^{\infty} \lambda^{(m-1)\bar{L}} \sum_{i=0}^{\bar{L}-1} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)\mathbf{h}_{rk,m}^{(tr)} + \gamma \sum_{i=0}^{\infty} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{v}_r(p-i) \\
&= \gamma^2 \sum_{m=1}^{\infty} \lambda^{(m-1)\bar{L}} \mathbf{h}_{rk,m}^{(tr)} \underbrace{\left[\sum_{i=0}^{\bar{L}-1} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)\right]}_{=:\mathbf{D}} + \underbrace{\gamma \sum_{i=0}^{\infty} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{v}_r(p-i)}_{=:\tilde{\mathbf{w}}}. \quad (4.41)
\end{aligned}$$

Then using (4.40) and (4.41), we can rewrite (4.35) as

$$\begin{aligned}
\hat{\mathbf{h}}_{rk} &= \left[\frac{1-\lambda^{\bar{L}}}{\gamma^2}\mathbf{D}^{-1}\right]\left[\gamma^2 \mathbf{D} \sum_{m=1}^{\infty} \lambda^{(m-1)\bar{L}} \mathbf{h}_{rk,m}^{(tr)} + \tilde{\mathbf{w}}\right] \\
&= (1-\lambda^{\bar{L}}) \sum_{m=1}^{\infty} \lambda^{(m-1)\bar{L}} \mathbf{h}_{rk,m}^{(tr)} + (1-\lambda^{\bar{L}})(\gamma^2\mathbf{D})^{-1}\tilde{\mathbf{w}}. \quad (4.42)
\end{aligned}$$

For memory length considerations, we will consider only the $\mathbf{noise-free}$ case, and hence we set $\tilde{\mathbf{w}} = 0$ in (4.40). With this restriction together with $\lambda^{\bar{L}} \ll 1$, we obtain

$$\begin{aligned}
\hat{\mathbf{h}}_{rk} &= (1-\lambda^{\bar{L}}) \sum_{m=1}^{\infty} \lambda^{(m-1)\bar{L}} \mathbf{h}_{rk,m}^{(tr)} \\
&= (1-\lambda^{\bar{L}}) \left[\mathbf{h}_{rk,1}^{(tr)} + \lambda^{\bar{L}} \mathbf{h}_{rk,2}^{(tr)} + \lambda^{2\bar{L}} \mathbf{h}_{rk,3}^{(tr)} + \cdots\right] \\
&\approx \mathbf{h}_{rk,1}^{(tr)} + \lambda^{\bar{L}} \mathbf{h}_{rk,2}^{(tr)}. \quad (4.43)
\end{aligned}$$

For WSSUS channel, $\mathbf{h}_{rk,1}^{(tr)}$ and $\mathbf{h}_{rk,2}^{(tr)}$ have the same statistics, $E\left\{\left\|\mathbf{h}_{rk,1}^{(tr)}\right\|^2\right\} = E\left\{\left\|\mathbf{h}_{rk,2}^{(tr)}\right\|^2\right\}$. The normalized error norm in estimating $\mathbf{h}_{rk,1}^{(tr)}$ via $\hat{\mathbf{h}}_{rk}$ is therefore given by

$$e_{norm} := \frac{\sqrt{E\left\{\left\|\lambda^{\bar{L}}\mathbf{h}_{rk,2}^{(tr)}\right\|^2\right\}}}{\sqrt{E\left\{\left\|\mathbf{h}_{rk,1}^{(tr)}\right\|^2\right\}}} = \lambda^{\bar{L}}. \quad (4.44)$$

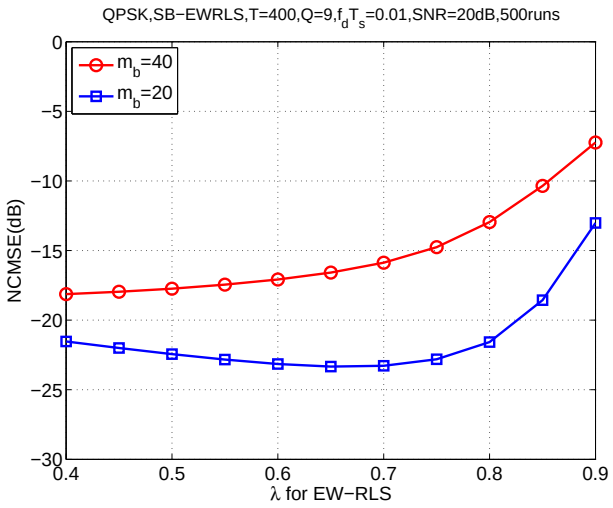
We would like to keep $e_{norm} \leq \delta$ where δ is pre-specified for the desired performance. This leads to

$$\lambda \leq \delta^{1/\bar{L}} = \delta^{m_b/T}. \quad (4.45)$$

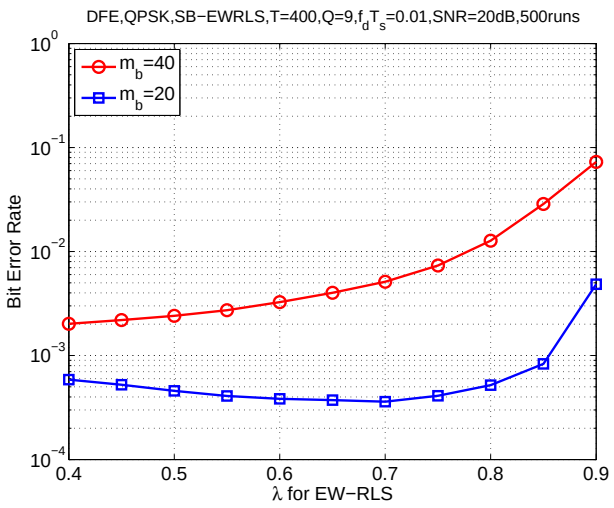
Given the CE-BEM period $T = 400$, some theoretical choices of λ are shown in Table 4.4. In Figs. 4.7a and 4.7b, the performances of EW-RLS algorithm are shown for different values of λ based on 500 Monte Carlo runs with the same environment as the simulation examples in Section 4.2.3 and 4.3.2. In our simulations, we picked the forgetting factor $\lambda = 0.65$ and 0.5 for $m_b = 20$ and 40 respectively, which satisfy the theoretical condition in (4.45). Indeed, $\delta = 0.001$ yields values of $\lambda = \delta^{m_b/T}$ that are close to the values suggested by the minima in Fig. 4.7.

Table 4.4: Theoretical λ for subblock-wise EW-RLS channel estimation.

δ	T	m_b	λ
0.01	400	20	≤ 0.794
		40	≤ 0.631
0.001	400	20	≤ 0.71
		40	≤ 0.50



(a) NCMSE vs λ



(b) BER vs λ, with DFE of $l_f = 8, l_b = 2$ and delay $d = 5$

Figure 4.7: Subblock-wise EW-RLS channel estimation: performances for forgetting factor λ's, under $f_d T_s = 0.01, \text{SNR} = 20\text{dB}, m_b = 20$ and 40.

4.5 Conclusions

The subblock-wise finite-memory recursive least-squares (RLS) tracking approach is presented, exploiting the oversampled complex exponential basis expansion model (CE-BEM) for the overall channel variations. We apply the exponentially-weighted and sliding-window recursive least-squares (RLS) algorithms to track the BEM coefficients, using time-multiplexed periodically transmitted training symbols. The proposed subblock-wise RLS schemes, which have no *a priori* model for the BEM coefficients, outperform the subblock-wise Kalman tracking in Chapter 3, since the AR assumption for the BEM coefficients used in the latter scheme may introduce additional modeling errors. Simulation examples illustrate the superior performance of our subblock-wise scheme to the conventional block-wise channel estimator and the subblock-wise Kalman tracking in Chapter 3.

Chapter 5
Turbo Equalization for Doubly-Selective Channels Using Nonliner Kalman Filtering and Basis Expansion Models

We present a turbo equalization receiver with nonlinear Kalman filtering, exploiting the complex exponential basis expansion model (CE-BEM) for the overall channel variations and the autoregressive (AR) model to update the BEM coefficients. 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. In the proposed receiver, an adaptive equalizer using nonlinear Kalman filters with delay is coupled with a soft-in soft-out (SfiSfo) module to perform the reception process iteratively. The proposed adaptive equalizer jointly optimizes the estimation of BEM channel coefficients and data symbols, thereby automatically accounting for the correlation between the channel estimates and data symbols in the equalization process. Simulation examples demonstrate our CE-BEM based approach has superior performance over an existing AR-only-based turbo equalizer.

5.1 Introduction

Among various models for channel time-variations, the 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 [6, 16, 68]. In fast time-varying channel environments, however, channel prediction using an AR model may lead to high estimation variance resulting in erroneous symbol decisions [6]. The BEM depicts 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 [70].

In Chapter 3, a subblock-wise tracking approach was investigated for doubly-selective channels using time-multiplexed (TM) training. It exploits the complex exponential BEM for the overall channel variations of each (overlapping) block, and a first-order AR model to describe the evolutions of the BEM coefficients. 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. This approach achieves better performance in fast-fading environments, than using conventional symbol-wise AR models.

In this chapter, we propose BEM-based approach to coded modulation communication systems using turbo equalization receiver. Turbo (iterative) equalization is a powerful sub-optimal technique used in place of the computationally prohibitive but maximum-likelihood (ML) or maximum *a posteriori* (MAP) sequence detection based on a super trellis. Although originally proposed for parallel concatenated error correction codes, the turbo principle is shown to be applicable to the detection problem for coded systems with ISI in [5]. 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 much lower complexity [1,5]. In [71], 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 CDMA. Based on this work, a variety of SfiSfo equalizers employing linear MMSE and decision feedback equalization (DFE) are proposed in [38, 39].

For doubly-selective channels, an adaptive SfiSfo equalizer has been presented in [68], using 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 SfiSfo decoder as its *a priori* information, and performs equalization process iteratively. With such an approach, the proposed scheme jointly optimizes the estimates of the channel and data symbols in each iteration. This avoids the common drawback in separate channel estimation and equalization/detection approach in that the correlation between channel estimate and data symbol decision is considered. The complexity of [68] is comparable to that of the turbo equalizers using linear filters [36, 48, 57], and is usually much lower than that of the ML/MAP based joint channel estimation and data detection schemes.

Based on the turbo equalization approach proposed in [68] and CE-BEM, we present an adaptive turbo equalizer with nonlinear Kalman filtering. 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 channel. This adaptive SfiSfo equalizer takes the decision of data symbols provided by SfiSfo decoder as its *a priori* information and the performance can be improved iteratively. The proposed adaptive equalizer jointly optimizes the estimates of BEM channel coefficients and data symbols in each equalization process since the correlation between the estimates of the channel and data symbols is considered. Simulation examples demonstrate our CE-BEM based scheme has superior performance over the turbo equalizer in [68] that relies on the AR modeling of channel. [It has been shown in [68] that their approach has better performance than any other turbo approaches of [36, 48, 57]; hence we compare our approach only with [68].]

5.2 Turbo Equalization using Extended Kalman Filter (EKF) and CE-BEM

Unlike the prior works [11,14,70], 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 (5.2) 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.

5.2.1 System Model and Turbo Equalization Receiver

Consider a doubly-selective (time- and frequency-selective) single-input single-output (SISO), finite impulse response (FIR) linear channel. Let $\{s(n)\}$ denote a scalar sequence that is input to the time-varying channel with discrete-time response $\{h(n;l)\}$ (channel response at time n to a unit input at time $n-l$). Then the symbol-rate noisy channel output is given by $(n = 0, 1, \ldots)$

$$y(n) = \sum_{l=0}^{L} h(n;l)\, s(n-l) + v(n) \tag{5.1}$$

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 [60]. We also assume that $s(n)$ is mutually independent and identically distributed (i.i.d.) with zero mean and variance $E\{s(n)\, s^*(n)\} = \sigma_s^2$.

We use the oversampled CE-BEM for channel modeling comparing AR model in [68]. In CE-BEM [11,14,70], over the i-th block consisting of an observation window of T_B symbols, the channel is represented as

$$h(n;l) = \sum_{q=1}^{Q} h_q^{(l)} e^{j\omega_q n}, \tag{5.2}$$

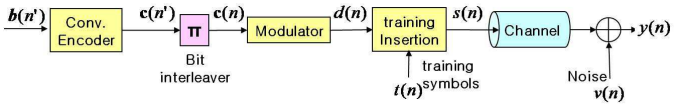


Figure 5.1: Bit-interleaved coded modulation system model for doubly-selective fading channel

where $n = \bar{n}_i, \bar{n}_i + 1, \cdots, \bar{n}_i + T_B - 1$ with $\bar{n}_i := (i-1)\, T_B$, $l = 0, 1, \cdots, L$ and one chooses (Λ is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{5.3}$$

$$Q := 2\left\lceil f_d T T_s \right\rceil + 1, \tag{5.4}$$

$$\omega_q := \frac{2\pi}{T}\left[q - (Q+1)/2\right], \quad q = 1, 2, \cdots, Q \tag{5.5}$$

$$L := \lfloor \tau_d / T_s \rfloor \tag{5.6}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration. The BEM coefficients $h_q^{(l)}$'s remain invariant during this block, but are allowed to change at the next consecutive block; the Fourier basis functions $\{e^{j\omega_q n}\}$ $(q = 1, 2, \cdots, Q)$ are common for every block. If the delay spread and the Doppler spread (or at least their upper bounds) are known, one can infer the basis functions of the CE-BEM [70]. Treating the basis functions as known, estimation of a time-varying process is reduced to estimating the invariant coefficients over a block of T_B symbols.

Bit-Interleaved Coded Modulation (BICM)

We consider a BICM transmitter (as in [67]) for a doubly-selective fading channel as shown in Fig. 5.1. A sequence of independent data vector $b(n') = [b^1(n'), b^2(n'), \cdots, b^{k_0}(n')] \in \{1, 0\}^{k_0}$ are 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 signal sequence $d(n)$ over a 2-dimensional signal constellation $\mathcal{X}$ of cardinality $\mathcal{M} = 2^m$ by a $\mathcal{M}$-ary modulator with an one-to-one binary map $\mu : \{1, 0\}^m \to \mathcal{X}$. In this section, we only consider the case of phase-shift keying (PSK) or quadrature amplitude modulation (QAM) with the average energy of the constellation $\mathcal{X}$ to be unity. That is, the signal $d(n)$ drawn from $\mathcal{X}$ 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

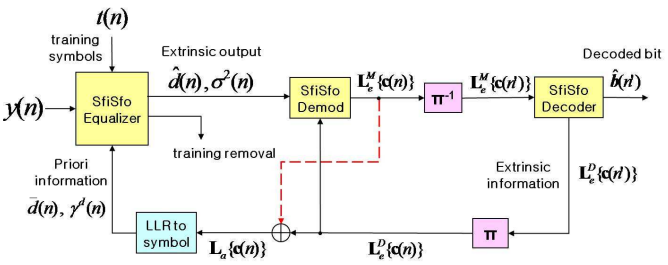


Figure 5.2: Turbo equalization receiver. Following [10, 49, 68] and contrary to the original turbo principle, *a posteriori* LLR $\mathbf{L}_a\{\mathbf{c}(n)\} = \mathbf{L}_e^M\{\mathbf{c}(n)\} + \mathbf{L}_e^D\{\mathbf{c}(n)\}$ instead of the extrinsic LLR $\mathbf{L}_e^D\{\mathbf{c}(n)\}$ can be input 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 [10, 49, 68].

signal constellation $\mathcal{X}$ with equal probabilities. The symbol $\{s(n)\}$ will be used to denote the symbol sequence after training($\{t(n)\}$) insertion into data symbol sequence($\{d(n)\}$).

Receiver Structure

A turbo equalization structure, as depicted in Fig. 5.2, is employed in the receiver, as in [68] except that [68] uses symbol-wise AR models. The adaptive SfiSfo equalizer is embedded into the iterative decoding (ID) process of the BICM transmission system (BICM-ID) [67]. In each decoding iteration, the equalizer takes the training symbols and the soft decision information about data symbols supplied by the SfiSfo decoder from the previous iteration as its *a priori* 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 SfiSfo equalizer in Section 5.2.2, we have extrinsic information for the data symbols $d(n)$. The training symbols are removed at the SfiSfo equalizer output and the iterative process that follows is only for data symbols. The SfiSfo equalizer based on the CE-BEM is described in Section 5.2.2. The SfiSfo demodulator follows [67] whereas the SfiSfo decoder follows the MAP decoding algorithm ("BCJR") [72, Section 6.2].

The data symbol estimates $\hat{d}(n)$ and its error variance $\sigma^2(n)$ are passed to the SfiSfo 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_i received symbols $\{y(l),\ 0 \le l < T_i\}$ (only for the information symbols, not the training), denoted by

$$\mathbf{L}_e^M\{\mathbf{c}(n)\} = \left[L_e^M\left\{c^i(n)\right\},\ i = 1, 2\cdots, n_0\right], \tag{5.7}$$

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

$$L_e^M\left\{c^i(n)\right\} := \ln \frac{P\{c^i(n)=1 \mid y(l),\ 0 \le l < T_i\}}{P\{c^i(n)=0 \mid y(l),\ 0 \le l < T_i\}} - \underbrace{\ln \frac{P\{c^i(n)=1\}}{P\{c^i(n)=0\}}}_{=:L\{c^i(n)\}}, \tag{5.8}$$

and $L\{c^i(n)\}$ is the *a priori* LLR. In (5.8), $P\{c^i(n)=b \mid y(l),\ 0 \le l < T_i\}$, $b \in \{0,1\}$, is approximated as

$$P\{c^i(n)=b \mid y(l),\ 0 \le l < T_i\} \approx P\{c^i(n)=b \mid \hat{d}(n)\} \tag{5.9}$$

by replacing the data $\{y(l),\ 0 \le l < T_i\}$ 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 (5.8) and (5.9) that

$$L_e^M\left\{c^i(n)\right\} = \ln \frac{P\{\hat{d}(n) \mid c^i(n)=1\}}{P\{\hat{d}(n) \mid c^i(n)=0\}}. \tag{5.10}$$

The soft estimate $\hat{d}(n)$ of $d(n)$ follows from the fixed-lag SfiSfo Kalman equalizer discussed later in Section 5.2.2 (following [68]), and it and its variance are given by (5.51) and (5.52), respectively. We assume that $\hat{d}(n)$ is complex Gaussian distributed with mean $d \in \mathcal{X}$ and variance $\sigma^2(n)$ and follow [67] to calculate (5.10). Let $c^i\langle d\rangle$ denote the i-th coded bit in the block of coded bits $[c^1\langle d\rangle, c^2\langle d\rangle, \cdots, c^{n_0}\langle d\rangle]$ that is mapped to the symbol d; therefore $\mu([c^1, c^2, \cdots, c^{n_0}]) = d$. It then follows that

$$P\left\{d \mid c^i\right\} = \prod_{j=1, j\neq i}^{n_0} P\{c^j = c^j\langle d\rangle\}. \tag{5.11}$$

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

$$P\left\{\hat{d}(n) \mid d\right\} = \frac{1}{\pi\sigma^2(n)} \exp\left(-\frac{|\,\hat{d}(n)-d\,|^2}{\sigma^2(n)}\right). \tag{5.12}$$

Recall that we used $\mathcal{X}$ to denote the set of all possible data symbols. Let $\mathcal{X}(i,b) = \{\mu([c^1, c^2, \cdots, c^{n_0}]) \mid c^i = b\}$, 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 (5.10)-(5.12) one obtains

$$L_e^M\left\{c^i(n)\right\} = \ln \frac{\sum_{d\in\mathcal{X}(i,1)}\left[\exp\left(-\frac{|\hat{d}(n)-d|^2}{\sigma^2(n)}\right)\prod_{j=1,j\neq i}^{n_0} P\{c^j(n)=c^j\,\langle d\rangle\}\right]}{\sum_{d\in\mathcal{X}(i,0)}\left[\exp\left(-\frac{|\hat{d}(n)-d|^2}{\sigma^2(n)}\right)\prod_{j=1,j\neq i}^{n_0} P\{c^j(n)=c^j\,\langle d\rangle\}\right]}. \tag{5.13}$$

The output extrinsic bit LLR's of the SfiSfo demodulator are bit-wise de-interleaved as $\mathbf{L}_e^M\{\mathbf{c}(n')\} = \pi^{-1}[\mathbf{L}_e^M\{\mathbf{c}(n)\}]$, which are then input to the SfiSfo convolutional decoder. In SfiSfo decoder, the MAP decoding algorithm for convolutional codes (see Section 2.5.2 [72, Section 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

$$\mathbf{L}_e^D\{\mathbf{c}(n')\} = \left[L_e^D\left\{c^i(n')\right\},\ \ i=1,2\cdots,n_0\right], \tag{5.14}$$

where

$$L_e^D\left\{c^i(n')\right\} := \ln\frac{P\{c^i(n')=1\mid \mathbf{L}_e^M\{\mathbf{c}(l)\},\ 0\le l<T_i\}}{P\{c^i(n')=0\mid \mathbf{L}_e^M\{\mathbf{c}(l)\},\ 0\le l<T_i\}} - \underbrace{L_e^M\{c^i(n')\}}_{=:L\{c^i(n')\}}. \tag{5.15}$$

The output bit LLR's of SfiSfo decoder are bit-wise interleaved as $\mathbf{L}_e^D\{\mathbf{c}(n)\} = \pi[\mathbf{L}_e^D\{\mathbf{c}(n')\}]$. The SfiSfo demodulator performs symbol-by-symbol MAP demodulation using LLR's $\mathbf{L}_e^D\{\mathbf{c}(n)\}$ for the coded bits generated by SfiSfo decoder in the previous iteration as its *a priori* information: $L\{c^i(n)\} = L_e^D\{c^i(n)\}$. We set $L_e^D\{c^i(n)\} = 0$ for the initial step (first iteration). The LLR for use in LLR-to-symbol block is computed via $\mathbf{L}_a\{\mathbf{c}(n)\} = \mathbf{L}_e^M\{\mathbf{c}(n)\} + \mathbf{L}_e^D\{\mathbf{c}(n)\}$, which is the the *a posteriori* LLR. [It is claimed in [10] that, unlike the original turbo principle where one takes $\mathbf{L}_a\{\mathbf{c}(n)\} = \mathbf{L}_e^M\{\mathbf{c}(n)\}$, usage of the full SfiSfo decoder's soft information embodied in the *a posteriori* LLR $\mathbf{L}_a\{\mathbf{c}(n)\}$ enhances performance compared to using only $\mathbf{L}_e^D\{\mathbf{c}(n)\}$; [68] also uses this set-up. This has also been our experience in the simulations presented; therefore, we have followed this approach.] The bit probabilities (converted from the corresponding LLR $\mathbf{L}_a\{\mathbf{c}(n)\}$) at time n are used (following [67]) to compute the mean $\bar{d}(n)$ and variance $\gamma^d(n)$ for data symbols $d(n)$ as

$$\bar{d}(n) = E[d(n)] = \sum_{d\in\mathcal{X}} d\,P\{d(n)=d\} = \sum_{d\in\mathcal{X}} d\prod_{j=1}^{n_0} P_a\left\{c^j(n)=c_d^j\right\}, \tag{5.16}$$

$$\gamma^d(n) = \mathrm{var}[d(n)] = \sum_{d\in\mathcal{X}} |d(n)-\bar{d}(n)|^2 P\{d(n)=d\} = 1-\bar{d}^2(n), \tag{5.17}$$

where

$$P_a\left\{c^j(n)=1\right\}=\frac{1}{1+\exp\left(-L_a\{c^j(n)\}\right)},\qquad P_a\left\{c^j(n)=0\right\}=\frac{1}{1+\exp\left(L_a\{c^j(n)\}\right)}. \tag{5.18}$$

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

5.2.2 Adaptive Soft-In Soft-Out Nonlinear Kalman Equalizer

Using a symbol-wise AR-model for channel variations, an adaptive SfiSfo equalizer using fixed-lag EKF was presented in [68] for joint channel estimation and equalization where their correlation was (implicitly) considered. In this section, we present a CE-BEM model-based SfiSfo nonlinear Kalman equalizer for turbo equalization.

State-Space Model using CE-BEM and *a Priori* Information

Stack the channel coefficients in (5.2) into vectors

$$\mathbf{h}_l:=\begin{bmatrix}h_1(l) & h_2(l) & \cdots & h_Q(l)\end{bmatrix}^T, \tag{5.19}$$

$$\mathbf{h}:=\begin{bmatrix}\mathbf{h}_0^T & \mathbf{h}_1^T & \cdots & \mathbf{h}_L^T\end{bmatrix}^T \tag{5.20}$$

of size Q and $Q(L+1)$ respectively. We will allow $\mathbf{h}$ in (5.20) 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, but we seek a "simple" AR(1) model given by

$$\mathbf{h}(n)=\mathbf{A}_1\mathbf{h}(n-1)+\mathbf{w}(n) \tag{5.21}$$

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

$$\tilde{\mathbf{h}}_B(n):=\Big[\; h(n;0) \quad h(n-1;0) \quad \cdots \quad h(n-T_B+1;0) \quad h(n;1) \quad h(n-1;1)$$

$$\cdots \quad h(n-T_B+1;1) \quad \cdots \quad h(n;L) \quad \cdots \quad h(n-T_B+1;L)\;\Big]^T. \tag{5.22}$$

Define

$$\boldsymbol{\mathcal{E}}(n):=\begin{bmatrix}e^{-j\omega_1 n} & e^{-j\omega_2 n} & \cdots & e^{-j\omega_Q n}\end{bmatrix}^T. \tag{5.23}$$

Using (5.23), we further define

$$\mathbf{B}\left(n\right) := \left[\boldsymbol{\mathcal{E}}\left(n\right) \quad \boldsymbol{\mathcal{E}}\left(n-1\right) \quad \cdots \quad \boldsymbol{\mathcal{E}}\left(n-T_B+1\right)\right]^H, \quad \boldsymbol{\Gamma} := \mathrm{diag}\left\{e^{j\omega_1}, e^{j\omega_2}, \ldots, e^{j\omega_Q}\right\}, \tag{5.24}$$

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

$$\tilde{\mathbf{h}}_B(n) = \tilde{\mathbf{B}}(n)\mathbf{h}(n), \quad \tilde{\mathbf{h}}_B(n+1) = \tilde{\mathbf{B}}(n)\tilde{\boldsymbol{\Gamma}}\mathbf{h}(n+1) \tag{5.25}$$

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

$$\tilde{\mathbf{B}}(n) := \mathrm{diag}\{\mathbf{B}(n),\ \mathbf{B}(n),\ \cdots,\ \mathbf{B}(n)\},\ \ \tilde{\boldsymbol{\Gamma}} := \mathrm{diag}\{\boldsymbol{\Gamma},\ \boldsymbol{\Gamma},\ \cdots,\ \boldsymbol{\Gamma}\}. \tag{5.26}$$

If (5.21) holds, then using the Yule-Walker equation we have

$$\mathbf{A}_1 = E\left\{\mathbf{h}(n+1)\mathbf{h}^H(n)\right\}\left[E\left\{\mathbf{h}(n)\mathbf{h}^H(n)\right\}\right]^{-1} \tag{5.27}$$

where using (5.25) we have

$$E\left\{\mathbf{h}(n+1)\mathbf{h}^H(n)\right\} = \tilde{\boldsymbol{\Gamma}}^{-1}\tilde{\mathbf{B}}^{\dagger}(n)E\left\{\tilde{\mathbf{h}}_B(n+1)\tilde{\mathbf{h}}_B^H(n)\right\}\tilde{\mathbf{B}}^{\dagger H}(n), \tag{5.28}$$

$$E\left\{\mathbf{h}(n)\mathbf{h}^H(n)\right\} = \tilde{\mathbf{B}}^{\dagger}(n)E\left\{\tilde{\mathbf{h}}_B(n)\tilde{\mathbf{h}}_B^H(n)\right\}\tilde{\mathbf{B}}^{\dagger H}(n), \tag{5.29}$$

and $E\left\{\tilde{\mathbf{h}}_B(n)\tilde{\mathbf{h}}_B^H(n)\right\}$ and $E\left\{\tilde{\mathbf{h}}_B(n+1)\tilde{\mathbf{h}}_B^H(n)\right\}$ can be calculated using (5.22) if we know the channel correlation function $\mathbf{R}_h(\tau)$. This procedure results in matching the correlation function of $\mathbf{h}\left(n\right)$ 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), and $E\{\mathbf{w}(n)\mathbf{w}^H(n)\} = \sigma_w^2\mathbf{I}_{Q(L+1)}$, leading to

$$\mathbf{h}\left(n\right) = \alpha\mathbf{h}\left(n-1\right) + \mathbf{w}\left(n\right). \tag{5.30}$$

If the channel is stationary (WSSUS) and coefficients $h_q(l)$'s are independent (as assumed in [70]), then by (5.30) and Yule-Walker equations, we can estimate α as

$$\alpha = \left(\frac{E\left\{\mathbf{h}^H(n+1)\mathbf{h}(n)\right\}}{E\left\{\mathbf{h}^H(n)\mathbf{h}(n)\right\}}\right)^* = \frac{\mathrm{tr}\left\{\tilde{\boldsymbol{\Gamma}}^{-1}\tilde{\mathbf{B}}^{\dagger}(n)E\left\{\tilde{\mathbf{h}}_B(n+1)\tilde{\mathbf{h}}_B^H(n)\right\}\tilde{\mathbf{B}}^{\dagger H}(n)\right\}}{\mathrm{tr}\left\{\tilde{\mathbf{B}}^{\dagger}(n)E\left\{\tilde{\mathbf{h}}_B(n)\tilde{\mathbf{h}}_B^H(n)\right\}\tilde{\mathbf{B}}^{\dagger H}(n)\right\}}, \tag{5.31}$$

and $\sigma_w^2 = E\{|h(n;l)|^2\}(1-|\alpha|^2)/Q$ where $\sigma_h^2 = E\{|h(n;l)|^2\}$. Note that (5.31) 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. [28] (in a different but similar context). Besides, for tracking, one needs $\alpha < 1$ [28]. To gain more insight, let us consider a mutually independent channel tap $h(n;l)$ for different l's following the Jakes' spectrum (also used in Section 5.2.3 in simulation examples). When $T = 200$, $T_B = 100$, $Q = 5$, and $f_d T_s = 0.01$, one gets α =0.99989 using (5.31). We compared it with $\mathbf{A}_1$ obtained via (5.27)-(5.29), 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 Section 5.2.3 (Fig. 5.8 and 5.9), this value of α is too close to one to permit tracking; we used $\alpha = 0.996$ in Section 5.2.3.] 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, not the block size T_B. Later we use (5.2) 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 (5.2) is periodic with period T 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, there are no deleterious effects due to the use of (5.2) for all time.

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

$$\bar{\delta} := \max\{\delta + 1, L + 1\} \tag{5.32}$$

and the data vector

$$\mathbf{z}(n) := \left[s(n) \quad s(n-1) \quad \cdots \quad s\left(n - \bar{\delta} + 1\right)\right]^T. \tag{5.33}$$

Consider (5.30). In order to apply (extended) Kalman filtering to joint channel estimation and equalization, we stack $\mathbf{h}(n)$ and data vector $\mathbf{z}(n)$ together into a $J \times 1$ state vector $\mathbf{x}(n)$ at time n as

$$\mathbf{x}(n) := \left[\mathbf{z}^T(n) \quad \mathbf{h}^T(n)\right]^T, \quad J := \bar{\delta} + Q(L+1). \tag{5.34}$$

As in [68] (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 (5.16) and (5.17), 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

$$\mathbf{x}(n) = \mathcal{T}\mathbf{x}(n-1) + \mathbf{e}_0\bar{s}(n) + \mathbf{u}(n), \tag{5.35}$$

where

$$\mathcal{T} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0}_{\bar{\delta}\times Q(L+1)} \\ \mathbf{0}_{Q(L+1)\times\bar{\delta}} & \mathbf{F} \end{bmatrix}_{J\times J}, \quad \mathbf{F} = \alpha\mathbf{I}_{Q(L+1)}, \tag{5.36}$$

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{0}_{1\times(\bar{\delta}-1)} & 0_{1\times 1} \\ \mathbf{I}_{(\bar{\delta}-1)} & \mathbf{0}_{(\bar{\delta}-1)\times 1} \end{bmatrix}_{\bar{\delta}\times\bar{\delta}}, \qquad \mathbf{e}_0 = \begin{bmatrix} 1 & \mathbf{0}_{1\times(J-1)} \end{bmatrix}^T, \tag{5.37}$$

the vector

$$\mathbf{u}(n) := \begin{bmatrix} \mathbf{e}_{\bar{\delta}}^T\tilde{s}(n) & \mathbf{w}^T(n) \end{bmatrix}^T \tag{5.38}$$

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 (5.21) and

$$\mathbf{Q}(n) := E[\mathbf{u}(n)\mathbf{u}^H(n)] = \tilde{\mathbf{Q}} + \gamma(n)\mathbf{e}_0\mathbf{e}_0^T, \quad \tilde{\mathbf{Q}} := \begin{bmatrix} \mathbf{0}_{\bar{\delta}\times\bar{\delta}} & \mathbf{0}_{\bar{\delta}\times Q(L+1)} \\ \mathbf{0}_{Q(L+1)\times\bar{\delta}} & \sigma_w^2\mathbf{I}_{Q(L+1)} \end{bmatrix}_{J\times J}. \tag{5.39}$$

The channel output $y(n)$ in (5.1) can be rewritten by CE-BEM given in (5.2) as

$$y(n) = \mathbf{s}^T(n)\left[\mathbf{I}_{(L+1)} \otimes \boldsymbol{\mathcal{E}}(n)\right]^H \mathbf{h}(n) + v(n), \tag{5.40}$$

where $\mathbf{s}(n) = [s(n)\ s(n-1)\ \cdots\ s(n-L)]^T$ (and $\boldsymbol{\mathcal{E}}(n)$ is as defined in (5.23)). Using the state vector that comprises the information symbols and channel coefficients, the measurement equation can be given as

$$y(n) = f[\mathbf{x}(n)] + v(n), \tag{5.41}$$

where

$$f[\mathbf{x}(n)] := \mathbf{x}^T(n)\underbrace{\begin{bmatrix} \mathbf{I}_{(L+1)} & \mathbf{0}_{(L+1)\times(J-L-1)} \end{bmatrix}^T \left[\mathbf{I}_{(L+1)} \otimes \boldsymbol{\mathcal{E}}(n)\right]^H \begin{bmatrix} \mathbf{0}_{[Q(L+1)]\times\bar{\delta}} & \mathbf{I}_{Q(L+1)} \end{bmatrix}}_{=:\mathbf{D}}\mathbf{x}(n). \tag{5.42}$$

With (5.35) and (5.41) as the state and measurement equations, respectively, nonlinear Kalman filtering is applied to track $\mathbf{x}(n)$ for joint channel estimation and equalization.

Fixed-Lag Soft Input Extended Kalman Filtering

In EKF, the state transition and observation models need not be linear functions of the state but may instead be (differentiable) functions. The state function can be used to compute the predicted state from the previous estimate and similarly the measurement function can be used to compute the predicted measurement from the predicted state. However, these functions cannot be applied to the covariance directly. Instead a matrix of partial derivatives (Jacobian) is computed. At each time-step the Jacobian is evaluated with current predicted states. Those matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

For the nonlinear system represented by (5.35) and (5.41), EKF is applied to track the channel BEM coefficients and to decode data symbols jointly. The EKF is initialized with

$$\hat{\mathbf{x}}\left(-1 \mid -1\right) = \mathbf{0} \;\text{ and }\; \mathbf{P}\left(-1 \mid -1\right) = \tilde{\mathbf{Q}} \tag{5.43}$$

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

$$\mathbf{P}\left(p \mid m\right) := E\{[\hat{\mathbf{x}}\left(p \mid m\right) - \mathbf{x}\left(p\right)][\hat{\mathbf{x}}\left(p \mid m\right) - \mathbf{x}\left(p\right)]^{H}\}. \tag{5.44}$$

Extended Kalman recursive filtering (for $n = 0, 1, 2, \cdots$) is applied as in [68] but with a different state and measurement equations, to generate $\hat{\mathbf{x}}\left(n \mid n\right)$ and $\mathbf{P}\left(n \mid n\right)$. The following steps are executed:

1. Time update:

$$\hat{\mathbf{x}}\left(n \mid n-1\right) = \mathcal{T}\hat{\mathbf{x}}\left(n-1 \mid n-1\right) + \mathbf{e}_0\bar{s}(n), \tag{5.45}$$

$$\mathbf{P}\left(n \mid n-1\right) = \mathcal{T}\mathbf{P}\left(n-1 \mid n-1\right)\mathcal{T}^{T} + \tilde{\mathbf{Q}} + \gamma(n)\mathbf{e}_0\mathbf{e}_0^{T}. \tag{5.46}$$

2. Kalman gain:

$$\eta(n) = \left.\frac{\partial f\left[\mathbf{x}\right]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(n|n-1)} = \hat{\mathbf{x}}^{T}(n \mid n-1)\left(\mathbf{D} + \mathbf{D}^{T}\right) \quad \ldots \text{ Jacobian matrix}, \tag{5.47}$$

$$\mathbf{K}(n) = \mathbf{P}(n \mid n-1)\eta^{H}(n) / \left[\sigma_v^2 + \eta(n)\mathbf{P}(n \mid n-1)\eta^{H}(n)\right]. \tag{5.48}$$

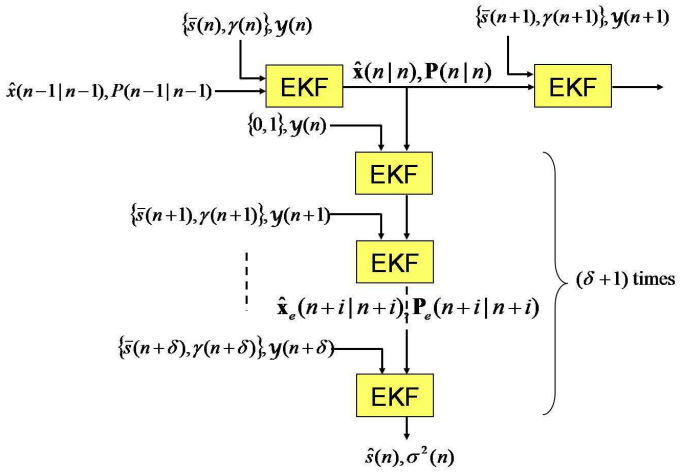


Figure 5.3: Structure of the adaptive SfiSfo equalizer proposed in [68]

3. Measurement update:

$$\hat{\mathbf{x}}\left(n \mid n\right) = \hat{\mathbf{x}}\left(n \mid n-1\right) + \mathbf{K}\left(n\right)\left(y\left(n\right) - f\left[\hat{\mathbf{x}}(n \mid n-1)\right]\right), \tag{5.49}$$

$$\mathbf{P}\left(n \mid n\right) = \left[\mathbf{I}_J - \mathbf{K}\left(n\right)\eta\left(n\right)\right]\mathbf{P}\left(n \mid n-1\right). \tag{5.50}$$

The *a priori* information $\{\bar{s}(n), \gamma(n)\}$ is the soft input at time n acquired via (5.16) and (5.17), while δ-th element of the estimate $\hat{\mathbf{x}}(n+\delta \mid n+\delta)$ is the delayed *a posteriori* estimate of data symbol. [Note that, in order to compensate the estimation errors, we take the noise variance in (5.48) to be $\sigma_v^2 + 0.01\sigma_s^2$ instead of σ_v^2 in our simulations, which is similar as Kalman detector in Section 2.4.1 and MMSE-DFF in Section 2.4.2. One can also take this increase as the stabilizing noise that is useful to solve a problem of EKF underestimating the true covariance matrix.]

Structure of Adaptive soft-in soft-out 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. 5.3 is used for SfiSfo equalization, just as in [68]. 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 $(i+1)$-th vertical filtering branch. Then the extrinsic

estimate $\hat{s}(n)$ of $s(n)$ and its error variance $\sigma^2(n)$ are given by

$$\hat{s}(n) = \delta\text{-th component of vector } \hat{\mathbf{x}}_e(n+\delta \mid n+\delta), \tag{5.51}$$

$$\sigma^2(n) = (\delta,\delta)\text{-th component of matrix } \mathbf{P}_e(n+\delta \mid n+\delta). \tag{5.52}$$

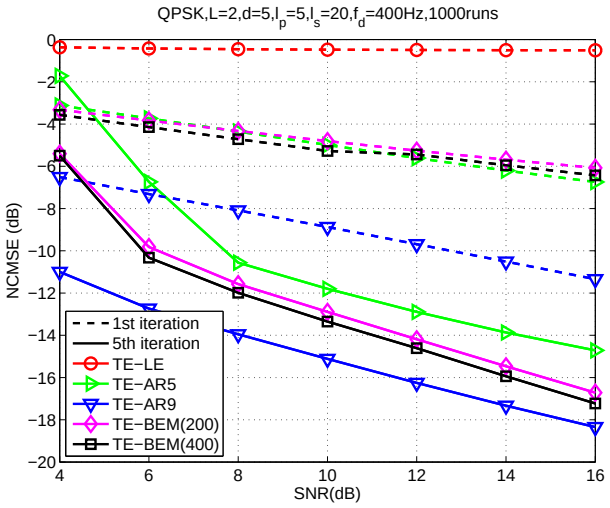
Note that the extrinsic outputs $\hat{s}(n)$ and $\sigma^2(n)$ are computed for data symbol $d(n)$, not for training symbol $t(n)$, and then used in the later parts of the turbo-equalization receiver (see Fig. 5.2). Further details regarding generation of extrinsic estimates can be found in [68].

Computational Complexity

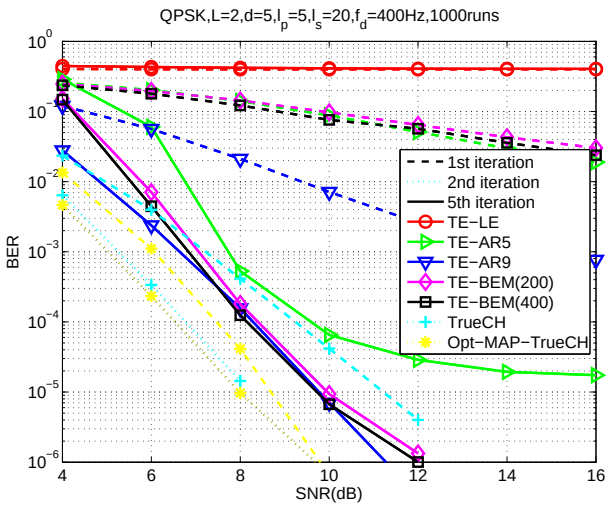
Here we consider computational complexity using the floating point operation (flop) counting for turbo equalization using EKF and CE-BEM. The computational complexity of the approach of [68] is $\mathcal{O}((\delta+2)[\bar{\delta}+P(L+1)]^2)$ where δ is the equalization delay, $\bar{\delta}$ is given by (5.32) and an AR(P) channel model is used. Note that it is independent of the constellation size $\mathcal{M}$. As we follow [68] 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 [68] have comparable computational complexity if one takes $P = Q$. As in [68], the computational complexity of our proposed approach is independent of the constellation size $\mathcal{M}$. In the simulations presented in Section 5.2.3, we have $\delta = 5$, $L = 2$ and $\bar{\delta} = 6$. For CE-BEM, we take $Q = 5$ or $Q = 9$, therefore, corresponding values of the AR model order P in the approach of [68] were picked as 5 or 9 to attain comparable computational requirements for a fair performance comparison.

5.2.3 Simulation Examples

A random time- and frequency-selective Rayleigh fading channel is considered. We assume $h(n;l)$ are zero-mean, complex Gaussian, and spatially white with autocorrelation σ_h^2. We take $L = 2$ (3 taps) in (5.1), and $\sigma_h^2 = 1/(L+1)$. For different l's, $h(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75](with a correction in the appendix of [63]). 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 $f_d T_s$ 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.



(a) NCMSE vs SNR



(b) BER vs SNR, with equalization delay $d = 5$

Figure 5.4: Turbo equalization: performance comparison for SNR's under $f_d T_s = 0.01, l_p = 5, l_s = 20$ (20% training overhead).

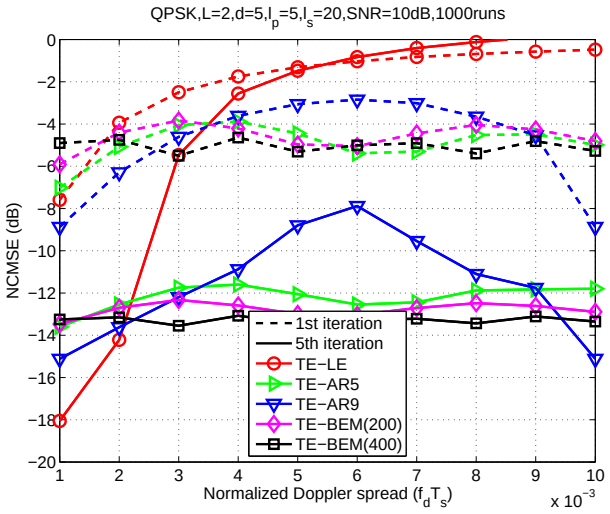
In the simulations, we use a 4-state convolutional code of rate $R_c = 1/2$ with octal generators $(5, 7)$. The information block size is set to 3000 bits (T_i=3000) leading to a coded block size of 6000 bits, and the interleaver size is equal to the coded block size. In the modulator, the QPSK constellation with Gray mapping is used, which gives $\mathcal{M} = 4$ 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).

We compared the following schemes:

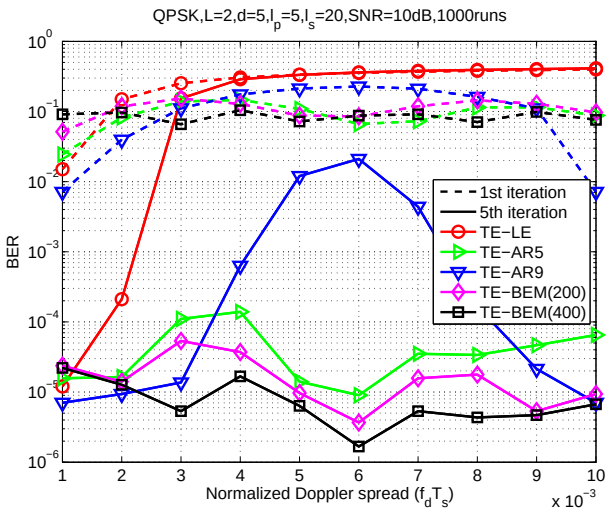
1. The approach of [48] that uses the linear MMSE equalizer (e.g. [38]) coupled with modified RLS channel estimation, where we set the linear filter length = 3 (6 pre-cursor taps and 3 post-cursor taps are used). This scheme is denoted by "TE-LE".

2. The AR(P) model-based scheme in [68]. The AR(P) model is as described in Section 2.2.1 and is fitted using [27] to Jakes' spectrum with $f_d T_s$=0.01 (the maximum anticipated normalized Doppler spread), denoted by "TE-AR5" for AR(5) model and "TE-AR9" for AR(9) model.

3. The proposed BEM-based turbo equalization schemes, where we consider BEM period $T = 200$ and 400 respectively, so that $Q = 5$ and 9, respectively, by (5.2). For the channel BEM coefficients, we take the AR-coefficient in (5.30) as $\alpha = 0.996$ for $T = 200$ and $\alpha = 0.998$ for $T = 400$. This scheme is denoted by "TE-BEM(200)" for $T = 200$ and "TE-BEM(400)" for $T = 400$.



(a) NCMSE vs f_dT_s



(b) BER vs f_dT_s, with equalization delay $d = 5$

Figure 5.5: Turbo equalization: performance comparison for normalized Doppler spread(f_dT_s)'s under SNR = 10dB, $l_p = 5, l_s = 20$ (20% training overhead).

4. The turbo equalizer based on the fixed-lag Kalman filter with perfect knowledge of the true channel, denoted by "TrueCH".

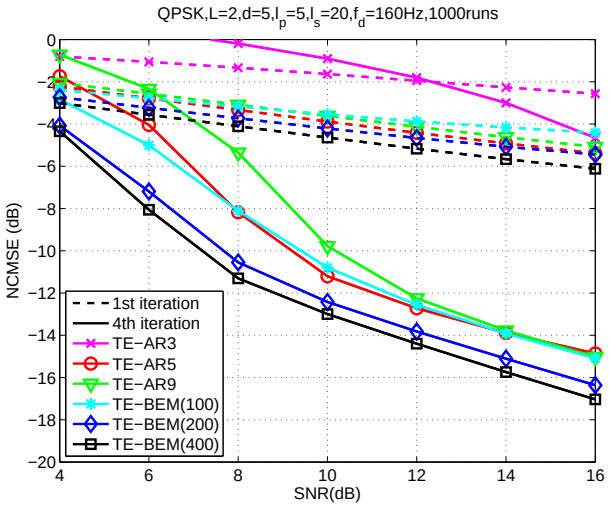
5. The turbo equalizer based on the optimum trellis-based MAP (BJCR) method [46] with perfect knowledge of the true channel, denoted by "Opt-MAP-TrueCH".

We evaluate the performances of various schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

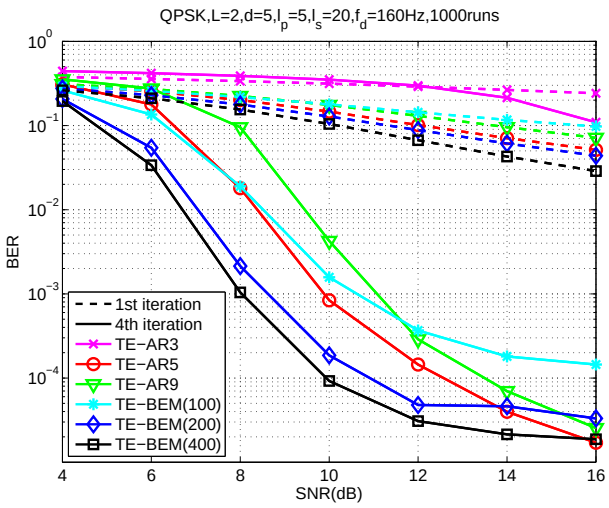
$$\text{NCMSE} := \frac{\sum_{i=1}^{M_r}\sum_{n=0}^{T_N-1}\sum_{l=0}^{L}\left\|\hat{h}^{(i)}(n;l)-h^{(i)}(n;l)\right\|^2}{\sum_{i=1}^{M_r}\sum_{n=0}^{T_N-1}\sum_{l=0}^{L}\left\|h^{(i)}(n;l)\right\|^2}$$

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_r runs. The BER's are evaluated by employing the equalization delay $\delta = 5$, using the decoded information symbol sequences at the turbo-equalization receiver. All the simulation results are based on 1000 runs.

In Fig. 5.4, the performance of all the above schemes, under normalized Doppler spread $f_dT_s = 0.01$, are compared for different SNR's. In Fig. 5.5, those schemes are compared over varying Doppler spread f_d's, under SNR = 10dB. Other settings of the simulation are same as Fig. 5.4, including the fact that $Q = 5$ (for $T = 200$) or $Q = 9$ (for $T = 400$), regardless of the actual f_d. It is clear 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.



(a) NCMSE vs SNR

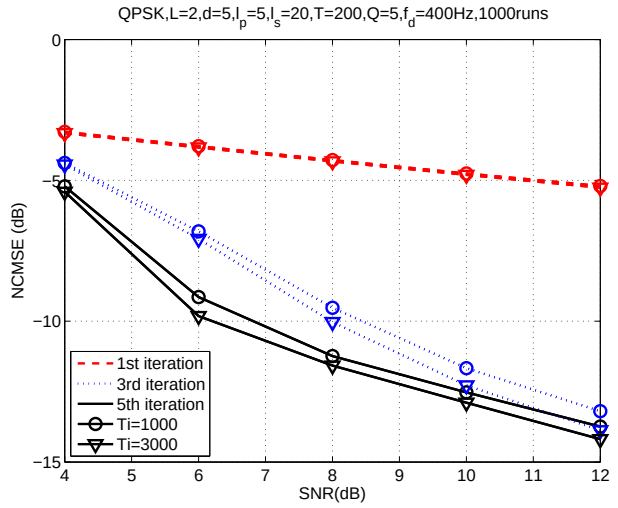


(b) BER vs SNR, with equalization delay $d = 5$

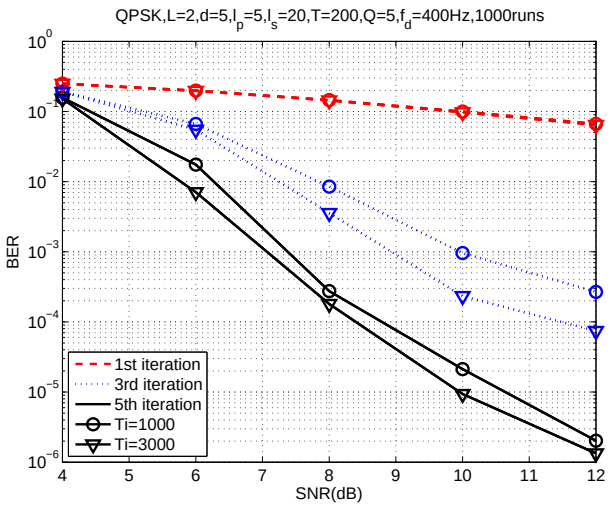
Figure 5.6: Turbo equalization: performance comparison for SNR's under $f_d T_s = 0.004, l_p = 5, l_s = 20$ (20% training overhead).

We can expect to save the signal power to get the same BER performance. Note that TE-BEM with larger block size ($T = 400$) has a little better performance than with smaller one ($T = 200$), since CE-BEM with $T = 400$ has the more basis functions in (5.2) and Kalman filtering utilize the past data implicitly(see Section 2.2.2). The NCMSE and BER for TE-BEM (especially, with $T = 400$) vary "slightly" along the range of normalized Doppler spread, so that given the CE-BEM representation used in channel tracking, its performances are not sensitive to the actual Doppler spread. Therefore, we do not have to know the exact Doppler spread of the channel — an upper bound of that 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. 5.4 with increasing SNR for a fixed $f_d T_s = 0.01$, and is slightly worse in Fig. 5.5 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. 5.4 with increasing SNR for a fixed $f_d T_s = 0.01$, it is significantly worse in Fig. 5.5 for a fixed SNR of 10dB and varying Doppler spreads. While increasing the BEM period T improves performance, increasing the AR model order does not necessarily do so: we get inconsistent performance. A possible reason is that, as noted in [27], 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 [27] in choosing the regularization parameter for the matrix inversion involved. Such inconsistent behavior is also seen in Fig. 5.6 where we compare performance of various schemes (including



(a) NCMSE vs SNR

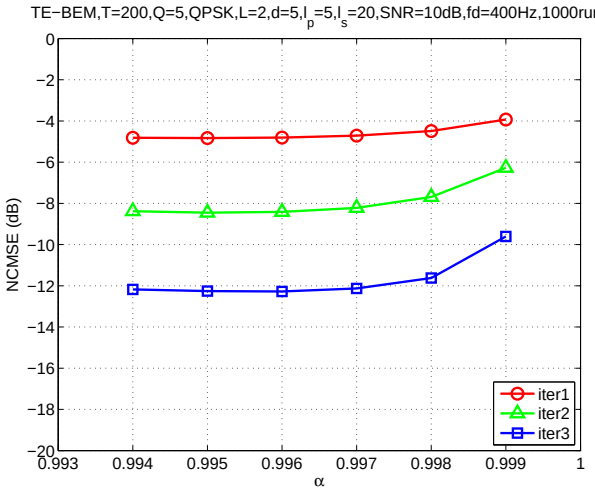


(b) BER vs SNR, with equalization delay $d = 5$

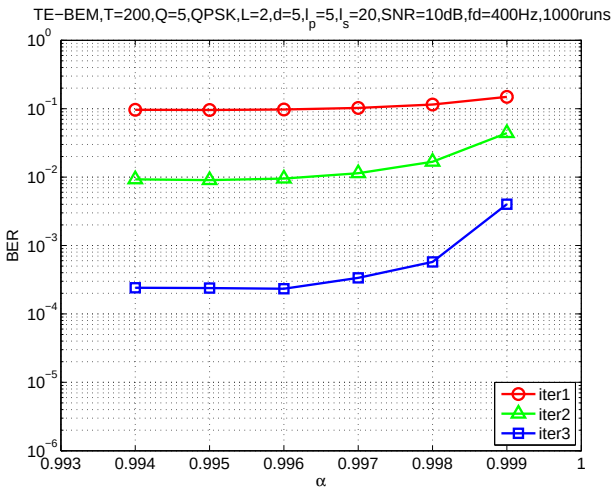
Figure 5.7: Turbo equalization: performance comparison of TE-BEM(200) with different interleaver lengths for SNR's under $f_d T_s = 0.01, l_p = 5, l_s = 20$ (20% training overhead).

TE-BEM(100) with $T = 100$ and $Q = 3$, and AR3 with order $P = 3$) for different SNR's under normalized Doppler spread $f_d T_s = 0.004$. It is seen that increasing the BEM period T 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. 5.4 and 5.5, it is seen that the linear MMSE equalizer coupled with modified RLS channel estimation (TE-LE) only works for normalized Doppler spread values of ≤ 0.002. Note that turbo equalization using linear MMSE equalizer with separate channel estimation only works well for slow fading channels, no matter what value of the filter length to choose [68]. In Fig. 5.4, we present the performance of the turbo equalizer based on the fixed-lag Kalman filter with knowledge of the true channel (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. 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 [46]. We also present the performance of the turbo equalizer based on the optimum BCJR method with knowledge of the true channel (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. It is seen that while there is a large difference in performance initially (see 1st iteration results for TrueCH and Opt-MAP-TrueCH), just one turbo iteration yields very close performance (see the two dotted curves).



(a) NCMSE vs α



(b) BER vs α with equalization delay $d = 5$

Figure 5.8: Turbo equalization: performances of TE-BEM(200) for AR(1) coefficient α's under $f_d T_s = 0.01, \text{SNR} = 10\text{dB}, l_p = 5, l_s = 20$ (20% training overhead).

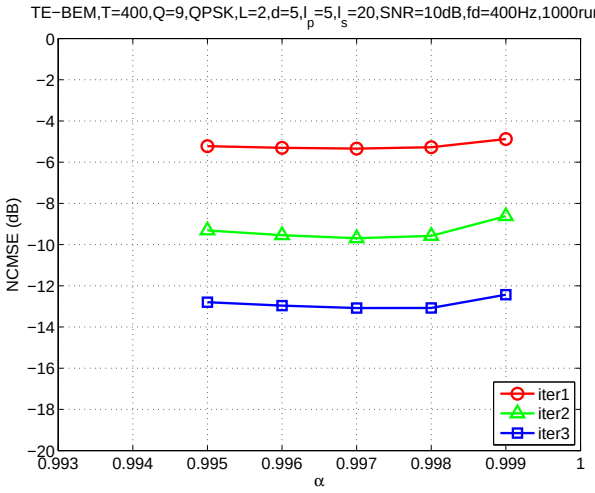
That is, at least for this example, performance loss in using Kalman equalizer instead of the BCJR equalizer is quite negligible.

In Fig. 5.7, a smaller information block size ($T_i = 1000$) in BICM transmitter is considered leading to a coded block size 2000 bits and 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 $T = 200, Q = 5$ in CE-BEM for different SNR's, under normalized Doppler spread $f_d T_s = 0.01$ with two different interleaver size (equivalently different information block size). It is seen that a smaller interleaver length results in a "small" deterioration in NCMSE and BER (when five iterations are considered).

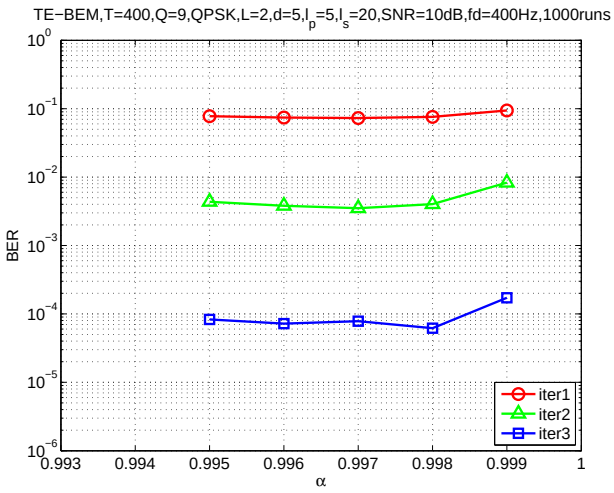
In Fig. 5.8 and 5.9 we show the BER performance of schemes TE-BEM(200) and TE-BEM(400) for different values of α respectively. It is seen that while the performance is not sensitive to the choice α over a relatively wide range of values, it does deteriorate as α approaches one. Note that $\alpha = 1$ in (5.30) 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 (5.31) also increases.

5.2.4 EXIT Chart Analysis

The extrinsic information transfer (EXIT) chart is a useful semi-analytic tool [37, 47, 59] to analyze the exchange of mutual information between the equalizer and the decoder and



(a) NCMSE vs α



(b) BER vs α, with equalization delay $d = 5$

Figure 5.9: Turbo equalization: performances of TE-BEM(400) for AR(1) coefficient α's under $f_d T_s = 0.01, \text{SNR} = 10\text{dB}, l_p = 5, l_s = 20$ (20% training overhead).

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 [37, 59]

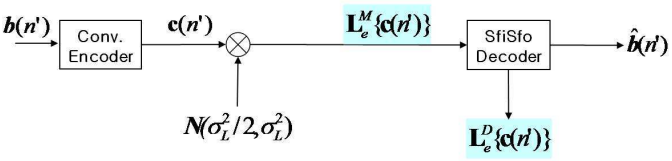
$$I(L;c) = 1 - E\left[\log_2(1+e^{-L}) \mid c = +1\right]. \tag{5.53}$$

Under ergodicity, for a large sample of size T_r, we have [37]

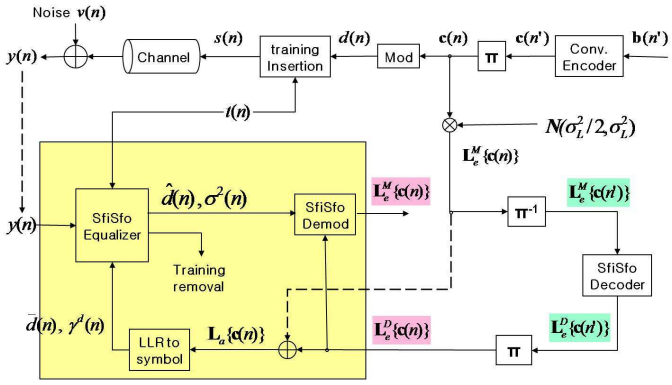
$$I(L;c) \approx 1 - T_r^{-1} \sum_{t=1}^{T_r} \log_2(1+e^{-c(t)L\{c(t)\}}). \tag{5.54}$$

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. 5.10. Following [59] (and others), $L_e^M\{c^i(n')\}$ (input to the SfiSfo decoder) is modeled as independent and identically distributed (i.i.d.) Gaussian with mean $c^i(n')\sigma_L^2/2$ and variance



(a) Decoder



(b) Equalizer

Figure 5.10: Simulation setup for generating extrinsic information transfer functions

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 (5.54). The interleaved random extrinsic LLR's $L_e^D\{c^i(n)\}$ are input to the "LLR to symbol" block in Fig. 5.10b 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 given SfiSfo 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. 5.11 for instance).

Using the set-up and parameters of Fig. 5.5 but with the information block size set to 30000 bits (the coded block size $T_r = 60000$), the normally distributed LLR's were generated with values of $\sigma_L^2 \in [10^{-2}, 10^2]$, and then I_e^M and I_e^D were calculated. We analyze the EXIT charts of our CE-BEM based approach with $T = 200$ and $Q = 5$ (TE-BEM(200) scheme) and the symbol-wise AR-model-based approach in [68] using AR(5) model (TE-AR5 scheme). In

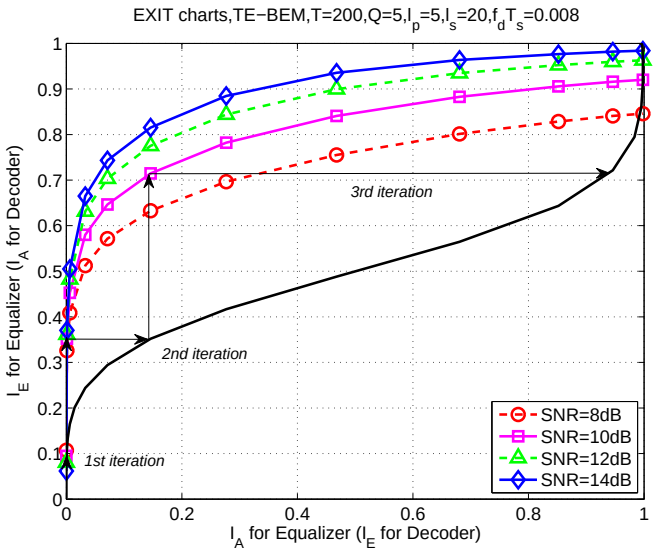
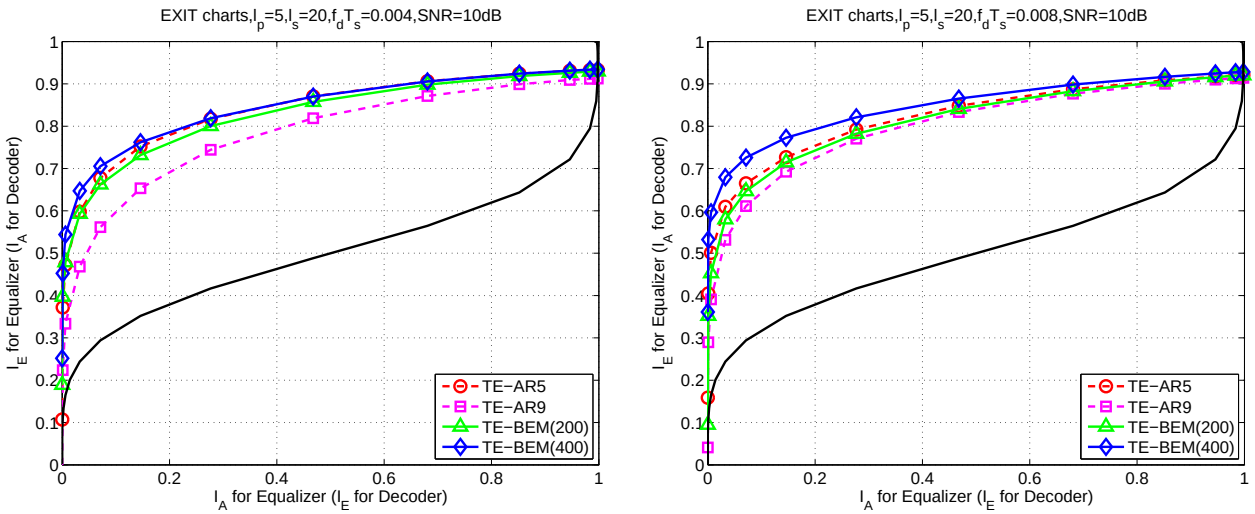


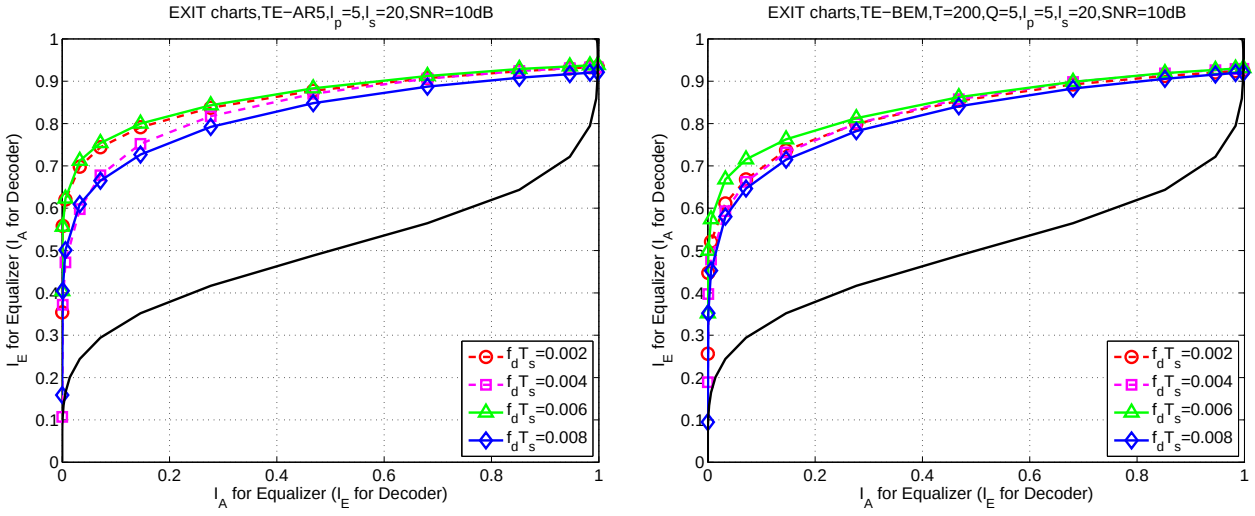
Figure 5.11: EXIT charts of the proposed turbo equalizer using CE-BEM with $T = 200, Q = 5$ for different SNR's under $f_dT_s = 0.008, l_p = 5, l_s = 20$ (20% training overhead).

Fig. 5.11, EXIT charts for TE-BEM(200) are shown under a fixed normalized Doppler spread $f_dT_s = 0.008$ for different SNR's. 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 SfiSfo equalizer increases. In Fig. 5.12, EXIT charts for the symbol-wise AR(P)-based turbo equalizations (TE-AR5, TE-AR9) and the proposed ones using CE-BEM (TE-BEM(200), TE-BEM(400)) are compared under fixed $f_dT_s = 0.004$ and 0.008, SNR = 10dB. In Fig. 5.13, EXIT charts for TE-BEM(200) and TE-AR5 schemes are depicted under SNR = 10dB for different normalized Doppler spreads. One can compare EXIT charts with the actual simulation performances for every scheme in Section 5.2.3. Table 5.1 compares the BER's obtained via full Monte Carlo simulations (as in Section 5.2.3, Fig. 5.13, 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.



(a) $f_dT_s = 0.004$ (b) $f_dT_s = 0.008$

Figure 5.12: EXIT charts for TE-AR5, TE-AR9, TE-BEM(200) and TE-BEM(400) under fixed f_dT_s, SNR $= 10\text{dB}, l_p = 5, l_s = 20$ (20% training overhead).



(a) TE-AR5: turbo equalization of [68] using symbol-wise AR(5) channel model

(b) TE-BEM(200): turbo equalization using CE-BEM with $T = 200, Q = 5$

Figure 5.13: EXIT charts for different f_dT_s's under SNR $= 10\text{dB}, l_p = 5, l_s = 20$ (20% training overhead).

Table 5.1: Turbo equalization: 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})

5.3 MIMO Turbo Equalization using EKF and CE-BEM

In this section, we extend the approach of Section 5.2 to multi-input multi-output (MIMO) systems. Several variants of DFE and linear transversal equalizer (LTE), such as the delayed decision feedback sequence estimator along with schemes like ordered successive interference cancellation has been proposed in [3, 73]. Considering a coded system, Partial response equalizers, which effectively shorten the original MIMO channel so as to follow it by a trellis based equalizers and the SfiSfo iterative Kalman equalizer pursuing low complexity for MIMO frequency selective fading channels have been proposed in [12] and [56] respectively. In contrast to [6] based on AR model only, our adaptive turbo equalizer takes advantage of BEM in addition to the conventional turbo processing and results in superior performance, even with smaller SNR's and over a wide range of Doppler spreads as demonstrated in simulation examples.

5.3.1 MIMO System Model and Turbo Equalization Receiver

Consider a doubly-selective (time- and frequency-selective) MIMO, finite impulse response (FIR) linear channel with K inputs and N outputs. Let $\{\mathbf{s}_k(n)\}$ denote k-th user's information sequence that is input to the time-varying channel with discrete-time response $\{\mathbf{h}_k(n;l)\}$ (channel response for the k-th user at time instance n to a unit input at time instance $n-l$). Then the symbol-rate noisy N-column channel output vector is given by $(n = 0, 1, \ldots)$

$$\mathbf{y}(n) = \sum_{k=1}^{K} \sum_{l=0}^{L} \mathbf{h}_k(n;l)\, s_k(n-l) + \mathbf{v}(n) \tag{5.55}$$

where the N-column vector $\mathbf{v}(n)$ is zero-mean, white, uncorrelated with $s_k(n)$, complex Gaussian noise, with the autocorrelation $E\{\mathbf{v}(n+\tau)\,\mathbf{v}^H(n)\} = \sigma_v^2 \mathbf{I}_N \delta(\tau)$. We assume that $\{\mathbf{h}_k(n;l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [60], independent for different k's. We assume that $s_k(n)$ is mutually independent and identically distributed (i.i.d.) with zero mean and variance $E\{s_k(n)\, s_k^*(n)\} = \sigma_{s_k}^2 = \sigma_s^2$ for $k = 1, 2, \cdots, K$. Define

$$\mathbf{s}(n) := \begin{bmatrix} s_1(n) & s_2(n) & \cdots & s_K(n) \end{bmatrix}^T$$
$$\mathbf{h}(n;l) := \begin{bmatrix} \mathbf{h}_1(n;l) & \mathbf{h}_2(n;l) & \cdots & \mathbf{h}_K(n;l) \end{bmatrix}$$

and then we may rewrite (5.55) as

$$\mathbf{y}(n) = \sum_{l=0}^{L} \mathbf{h}(n;l)\, \mathbf{s}(n-l) + \mathbf{v}(n). \tag{5.56}$$

In our approach, we use an oversampled CE-BEM for the overall channel variations and first-order AR model for the BEM coefficients. These two models are described in this subsection. In CE-BEM [11, 14, 70], over the i-th block consisting of an observation window of T_B symbols, the channel is represented as ($n = (i-1)T_B, (i-1)T_B + 1, \cdots, iT_B - 1$ and $l = 0, 1, \cdots, L$)

$$\mathbf{h}_k(n; l) = \sum_{q=1}^{Q} \mathbf{h}_{k,q}(l) e^{j\omega_q n}, \tag{5.57}$$

where $\mathbf{h}_{k,q}^{(l)}$ is the N-column time-invariant BEM coefficient vector for k-th user and one chooses (Λ is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{5.58}$$

$$Q := 2\lceil f_d T T_s \rceil + 1, \tag{5.59}$$

$$\omega_q := \frac{2\pi}{T}\left[q - (Q+1)/2\right], \quad q = 1, 2, \cdots, Q \tag{5.60}$$

$$L := \lfloor \tau_d / T_s \rfloor, \tag{5.61}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration. The BEM coefficients $\mathbf{h}_{k,q}(l)$'s remain invariant during this block, but are allowed to change at the next consecutive block; the Fourier basis functions $\{e^{j\omega_q n}\}$ ($q = 1, 2, \cdots, Q$) are common for every block. If the delay spread and the Doppler spread (or at least their upper bounds) are known, one can infer the basis functions of the CE-BEM [70]. Treating the basis functions as known, estimation of a time-varying process is reduced to estimating the invariant coefficients over a block of T_B symbols.

Bit-Interleaved Coded Modulation (BICM) for multiple users

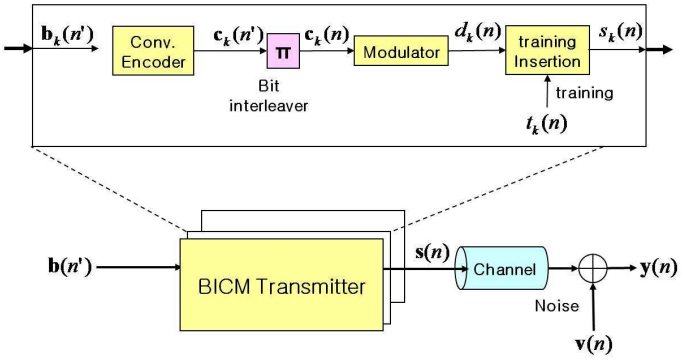


Figure 5.14: Bit-interleaved coded modulation system model for doubly-selective fading MIMO channels

We consider a BICM transmitter (as in [67]) for a doubly-selective MIMO channel as shown in Fig. 5.14. At the Transmitter, the input bit stream $\{\mathbf{b}(n)\}$ for K users are processed independently one another. A sequence of information data vector $\mathbf{b}_k(n') = [b_k^1(n'), b_k^2(n'), \cdots, b_k^{k_0}(n')] \in \{1,0\}^{k_0}$ for k-th input are fed into a convolutional encoder with a code rate $R_c = k_0/n_0$. The coded output $\mathbf{c}_k(n') = [c_k^1(n'), c_k^2(n'), \cdots, c_k^{n_0}(n')] \in \{1,0\}^{n_0}$ is passed through a bit-wise random interleaver π, generating the interleaved coded bit sequence $\mathbf{c}_k(n) = \pi[\mathbf{c}_k(n')]$. The binary coded bits are then mapped to a signal sequence $d_k(n)$ over a 2-dimensional signal constellation $\mathcal{X}$ of cardinality $\mathcal{M} = 2^m$ by a $\mathcal{M}$-ary modulator with an one-to-one binary map $\mu : \{1,0\}^m \to \mathcal{X}$. In this section, we only consider the case of phase-shift keying (PSK) or quadrature amplitude modulation (QAM) with the average energy of the constellation $\mathcal{X}$ to be unity. That is, the signal $d_k(n)$ drawn from $\mathcal{X}$ has mean $E\{d_k(n)\} = 0$ and variance $E\left\{|d_k(n)|^2\right\} = 1$. After modulation, we periodically insert short training sequences into the data symbol sequence. The training symbols $t_k(n)$, which are known to the receiver, are randomly drawn from the signal constellation $\mathcal{X}$ with equal probabilities. The $\{s_k(n)\}$ will be used to denote the symbol sequence after training($\{t_k(n)\}$) insertion into data symbol sequence($\{d_k(n)\}$).

MIMO Turbo Equalization Receiver

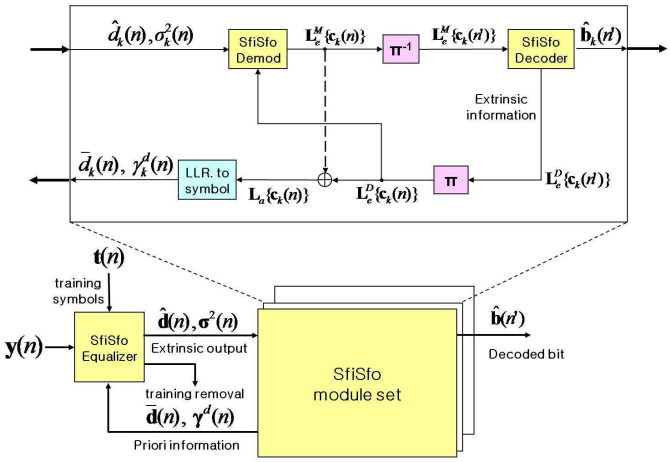


Figure 5.15: MIMO turbo equalization receiver. Following [10, 49, 68], *a posteriori* LLR $\mathbf{L}_a\{\mathbf{c}_k(n)\} = \mathbf{L}_e^M\{\mathbf{c}_k(n)\} + \mathbf{L}_e^D\{\mathbf{c}_k(n)\}$ is input to the LLR-to-symbol block. Inclusion of $\mathbf{L}_e^M\{\mathbf{c}(n)\}$ to create *a posteriori* LLR is shown via dashed line.

A turbo equalization structure, as depicted in Fig. 5.15, is employed in the receiver as used in [68] except that [68] uses symbol-wise AR models. The adaptive SfiSfo equalizer is embedded into the iterative decoding (ID) process of the BICM transmission system

(BICM-ID) [67]. In each decoding iteration, the equalizer takes the training symbols and the soft decision information about data symbols supplied by the SfiSfo decoder from the previous iteration as its *a priori* 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 $\hat{\mathbf{d}}(n)$ and its error covariance $\boldsymbol{\sigma}^2(n)$ for K users. Using the adaptive SfiSfo equalizer in Section 5.3.2, we have an extrinsic information for the data symbols $\mathbf{d}(n)$. The training symbols in the received signal are removed at the SfiSfo equalizer and the following iterative process is only for data symbols since training symbols are known to the receiver. Note that the extrinsic output of SfiSfo equalizer $\left\{\hat{d}_k(n), \sigma_k^2(n)\right\}$ for k-th user is processed independently in SfiSfo module set, where SfiSfo demodulation and SfiSfo decoding are implemented. The SfiSfo demodulator follows [67] whereas the SfiSfo decoder follows the MAP decoding algorithm ("BCJR") [72, Section 6.2]. [See the details for the SfiSfo demodulator and the SfiSfo decoder for k-th user in Section 5.2.1.]

5.3.2 Adaptive Soft-In Soft-Out Nonlinear Kalman Equalizer for Doubly-Selective MIMO Channels

In this section, we extend a CE-BEM model-based SfiSfo nonlinear Kalman equalizer in Section 5.2.2 to MIMO systems, implementing joint channel estimation and equalization where their correlation was (implicitly) considered.

State-Space Model using CE-BEM and *a Priori* Information

Stack the channel coefficients in (5.57) into into vectors

$$\mathbf{h}_q^{(l)} := \left[h_{11,q}^{(l)} \cdots h_{N1,q}^{(l)} \quad \cdots \quad h_{1K,q}^{(l)} \cdots h_{NK,q}^{(l)}\right]^T, \tag{5.62}$$

$$\mathbf{h}^{(l)} := \left[\mathbf{h}_1^{(l)T} \quad \mathbf{h}_2^{(l)T} \quad \cdots \quad \mathbf{h}_Q^{(l)T}\right]^T, \tag{5.63}$$

$$\mathbf{h} := \left[\mathbf{h}^{(0)T} \quad \mathbf{h}^{(1)T} \quad \cdots \quad \mathbf{h}^{(L)T}\right]^T \tag{5.64}$$

of size NK, NKQ and $J_1 := NKQ(L+1)$ respectively. We allow the coefficient vector in (5.64) 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, but we seek a "simple" AR(1) model given by

$$\mathbf{h}\,(n) = \mathbf{A}_1 \mathbf{h}\,(n-1) + \mathbf{w}\,(n) \tag{5.65}$$

where $\mathbf{A}_1 = \alpha \mathbf{I}_{J_1}$ is the AR coefficient matrix, and the driving noise vector $\mathbf{w}(n)$ is zero-mean complex Gaussian with variance $\sigma_w^2 \mathbf{I}_{J_1}$ and statistically independent of $\mathbf{h}(n-1)$. Assuming the channel is wide-sense stationary (WSS) and coefficients $h_{rk,q}^{(l)}$'s are independent, we have

$$\sigma_w^2 = \sigma_h^2 (1 - |\alpha|^2)/Q \tag{5.66}$$

where $\sigma_h^2 \mathbf{I}_{J_1} := E\left\{\mathbf{h}(n;l)\,\mathbf{h}^H(n;l)\right\}$. [Typically $\alpha < 1$ but close to one. Strictly speaking one should try a "full" matrix $\mathbf{A}_1$ in (5.64); however, it can only be calculated if the MIMO channel statistics are known – we do not assume such knowledge here, just an upperbound on the Doppler and multipath delay spreads. See details in Section 5.2.2]

Define with the equalization with a delay $\delta > 0$ and K-column vector $\mathbf{s}(n)$,

$$\bar{\delta} := \max\{\delta + 1, L + 1\} \tag{5.67}$$

$$\mathbf{z}(n) := \left[\mathbf{s}^T(n) \quad \mathbf{s}^T(n-1) \quad \cdots \quad \mathbf{s}^T\left(n - \bar{\delta} + 1\right)\right]^T. \tag{5.68}$$

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

$$\mathbf{x}(n) := \left[\mathbf{z}^T(n) \quad \mathbf{h}^T(n)\right]^T, \quad J := K\bar{\delta} + NKQ(L+1) = K\bar{\delta} + J_1. \tag{5.69}$$

As in [68] (and others), we consider the symbol sequence $\{\mathbf{s}(n)\}$ 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 k-th user's information sequence $s_k(n)$ as

$$s_k(n) = \bar{s}_k(n) + \tilde{s}_k(n), \tag{5.70}$$

where $\bar{s}_k(n) = E[s_k(n)]$ is a deterministic sequence and $\tilde{s}_k(n)$ is approximated as a zero-mean uncorrelated stochastic process such that $E[\tilde{s}_k(n)\tilde{s}_k^*(n+j)] = \gamma_k(n)\delta[j]$ assuming an ideal interleaver. Note that $\bar{s}_k(n)$ and $\gamma_k(n)$, the statistical characteristics of $s_k(n)$, are provided via the *a priori* information. We have $\bar{s}_k(n) = \bar{d}_k(n)$ and $\gamma_k(n) = \gamma_k^d(n)$ for a data symbol $d_k(n)$, where $\bar{d}_k(n)$ and $\gamma_k^d(n)$ are given in (5.16) and (5.17) respectively, while $\bar{s}_k(n) = t_k(n)$ and $\gamma_k(n) = 0$ for a training symbol $t_k(n)$. Using $\mathbf{x}(n)$, the state equation turns out to be

Using the state vector, $\mathbf{x}(n)$ in (5.69), the state equation can be written as

$$\mathbf{x}(n) = \mathcal{T}\mathbf{x}(n-1) + \mathbf{\Gamma}\bar{s}(n) + \mathbf{u}(n), \tag{5.71}$$

with the following definitions :

$$\mathcal{T} = \begin{bmatrix} \mathbf{\Phi} & 0_{K\bar{\delta}\times J_1}, \\ 0_{J_1\times K\bar{\delta}} & \mathbf{F} \end{bmatrix}_{J\times J}, \qquad \mathbf{F} = \alpha \mathbf{I}_{J_1}, \tag{5.72}$$

$$\mathbf{\Phi} = \begin{bmatrix} 0_{1\times(\bar{\delta}-1)} & 0_{1\times 1} \\ \mathbf{I}_{(\bar{\delta}-1)} & 0_{(\bar{\delta}-1)\times 1} \end{bmatrix} \otimes \mathbf{I}_K, \qquad \mathbf{\Gamma} = \begin{bmatrix} \mathbf{I}_K & 0_{K\times(J-K)} \end{bmatrix}^T. \tag{5.73}$$

The vector $\mathbf{u}(n)$ is zero-mean uncorrelated process noise, defined as

$$\mathbf{u}(n) = \begin{bmatrix} \mathbf{\Gamma}_{\bar{\delta}}^T \tilde{s}(n) & \mathbf{w}^T(n) \end{bmatrix}^T, \tag{5.74}$$

where $\mathbf{\Gamma}_{\bar{\delta}} = [1\ 0_{1\times(\bar{\delta}-1)}]^T \otimes \mathbf{I}_K$, and $\mathbf{w}(n)$ is given in (5.65). The covariance matrix of $\mathbf{u}(n)$ is given by

$$\mathbf{Q}(n) = E[\mathbf{u}(n)\mathbf{u}^H(n)] = \tilde{\mathbf{Q}} + \mathbf{\Gamma} diag\left\{\boldsymbol{\gamma}(n)\right\} \mathbf{\Gamma}^T, \tag{5.75}$$

where $diag\left\{\boldsymbol{\gamma}(n)\right\}$ is a diagonal matrix composed of $\gamma_k(n)$ and matrix $\tilde{\mathbf{Q}}$ is defined as

$$\tilde{\mathbf{Q}} = \begin{bmatrix} 0_{K\bar{\delta}\times K\bar{\delta}} & 0_{K\bar{\delta}\times J_1} \\ 0_{J_1\times K\bar{\delta}} & \sigma_w^2 \mathbf{I}_{J_1} \end{bmatrix}_{J\times J}. \tag{5.76}$$

Meanwhile, based on CE-BEM given in (5.57), the channel output $\mathbf{y}(n)$ in (5.56) can be rewritten as

$$\mathbf{y}(n) = \left[\mathbf{S}(n) \otimes \mathbf{I}_N\right]^T \left[\mathbf{I}_{(L+1)} \otimes \left(\boldsymbol{\mathcal{E}}(n) \otimes \mathbf{I}_{NK}\right)\right]^H \mathbf{h}(n) + \mathbf{v}(n), \tag{5.77}$$

where

$$\mathbf{S}(n) := \begin{bmatrix} \mathbf{s}^T(n) & \mathbf{s}^T(n-1) & \cdots & \mathbf{s}^T(n-L) \end{bmatrix}^T, \tag{5.78}$$

$$\boldsymbol{\mathcal{E}}(n) := \begin{bmatrix} e^{-j\omega_1 n} & e^{-j\omega_2 n} & \cdots & e^{-j\omega_Q n} \end{bmatrix}^T. \tag{5.79}$$

where $\mathbf{s}\,(n) = [s_1\,(n)\ \ s_2\,(n)\ \cdots\ s_K\,(n)]^T$. Note that the state vector $\mathbf{x}(n)$ comprises $\mathbf{S}(n)$ and $\mathbf{h}(n)$ in (5.69). Hence, the measurement equation using $\mathbf{x}(n)$ can be given as

$$\mathbf{y}(n) = \mathbf{f}[\mathbf{x}(n)] + \mathbf{v}(n), \tag{5.80}$$

where the nonlinear function for $\mathbf{f}[\mathbf{x}(n)]$ is defined as

$$\mathbf{f}[\mathbf{x}(n)] := \left[\mathbf{S}(n) \otimes \mathbf{I}_N\right]^T \left[\mathbf{I}_{(L+1)} \otimes \left(\boldsymbol{\mathcal{E}}(n) \otimes \mathbf{I}_{NK}\right)\right]^H \mathbf{h}(n), \tag{5.81}$$

Treating (5.71) and (5.80) respectively as the state and the measurement equations, nonlinear Kalman filtering can be applied to track the vector $\mathbf{x}(n)$ at each time for the joint channel estimation and equalization.

Fixed-Lag Soft Input Extended Kalman Filtering

The EKF is applied to (5.71) and (5.80) to track the channel BEM coefficients and to decode data symbols jointly. Kalman tracking is initialized with

$$\hat{\mathbf{x}}\left(-1 \mid -1\right) = \mathbf{0} \;\text{ and } \mathbf{P}\left(-1 \mid -1\right) = \tilde{\mathbf{Q}},$$

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

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

Extended Kalman recursive filtering (for $n = 0, 1, 2, \cdots$) is applied one by one via the following steps:

1. Time update:

$$\hat{\mathbf{x}}\left(n \mid n-1\right) = \mathcal{T}\hat{\mathbf{x}}\left(n-1 \mid n-1\right) + \mathbf{\Gamma}\overline{\mathbf{s}}(n), \tag{5.82}$$

$$\mathbf{P}\left(n \mid n-1\right) = \mathcal{T}\mathbf{P}\left(n-1 \mid n-1\right)\mathcal{T}^T + \tilde{\mathbf{Q}} + \mathbf{\Gamma} diag\left\{\boldsymbol{\gamma}(n)\right\}\mathbf{\Gamma}^T; \tag{5.83}$$

2. Kalman gain:

$$\boldsymbol{\eta}(n) = \left.\frac{\partial \mathbf{f}\left[\mathbf{x}\right]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(n|n-1)} \text{(Jacobian matrix)}, \tag{5.84}$$

$$\mathbf{K}(n) = \mathbf{P}(n \mid n-1)\boldsymbol{\eta}^H(n) / \left[\sigma_v^2\mathbf{I}_N + \boldsymbol{\eta}(n)\mathbf{P}(n \mid n-1)\boldsymbol{\eta}^H(n)\right](n). \tag{5.85}$$

3. Measurement update:

$$\hat{\mathbf{x}}\left(n \mid n\right) = \hat{\mathbf{x}}\left(n \mid n-1\right) + \mathbf{K}\left(n\right)\left\{\mathbf{y}\left(n\right) - \mathbf{f}\left[\hat{\mathbf{x}}(n \mid n-1)\right]\right\}, \tag{5.86}$$

$$\mathbf{P}\left(n \mid n\right) = \left[\mathbf{I}_J - \mathbf{K}\left(n\right)\boldsymbol{\eta}\left(n\right)\right]\mathbf{P}\left(n \mid n-1\right). \tag{5.87}$$

The *a priori* information $\{\overline{\mathbf{s}}(n), \boldsymbol{\gamma}(n)\}$ is the soft input at time n, a *a priori* information acquired from LLR-to-symbol block in Fig. 5.15, while $(K(\delta-1)+k)$-th elements (for $k = 1, 2, \cdots, K$) of the estimate $\hat{\mathbf{x}}(n+\delta \mid n+\delta)$ is the delayed *a posteriori* estimate of

data symbol for the k-th user. [Note that we increase the noise variance in (5.85) from σ_v^2 to $\sigma_v^2 + 0.1\sigma_s^2$ in our simulations in Section 5.3.3 to compensate the estimation errors.]

Structure of Adaptive soft-in soft-out MIMO Equalizer

The fixed-lag EKF takes soft inputs and generates a delayed *a posteriori* estimate for $\mathbf{s}(n)$. However, we need an extrinsic estimate as an output of SfiSfo equalizer at turbo-equalization receiver in Fig. 5.15. In order to generate extrinsic estimate independent of the *a priori* information $\{\bar{\mathbf{s}}(n), \boldsymbol{\gamma}(n)\}$, we use a "comb" structure in conjunction with the EKF in Fig. 5.3, where we have K-column vectors $\bar{\mathbf{s}}(n), \boldsymbol{\gamma}(n)$ and N-column vector $\mathbf{y}(n)$ for MIMO systems. At each time n, the vertical branch composed of $(\delta+1)$ EKFs produce the extrinsic estimate $\hat{\mathbf{s}}(n)$, 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 $\{\mathbf{0}_{K\times 1}, \mathbf{1}_{K\times 1}\}$ in place of $\{\bar{\mathbf{s}}(n), \boldsymbol{\gamma}(n)\}$ 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 $(i+1)$-th vertical filtering branch. Then the extrinsic estimate $\hat{s}_k(n)$ and its error covariance $\sigma_k^2(n)$ for $s_k(n)$ are given by

$$\hat{s}_k(n) = (K(\delta-1)+k)\text{-th component of vector } \hat{\mathbf{x}}_e(n+\delta \mid n+\delta), \tag{5.88}$$

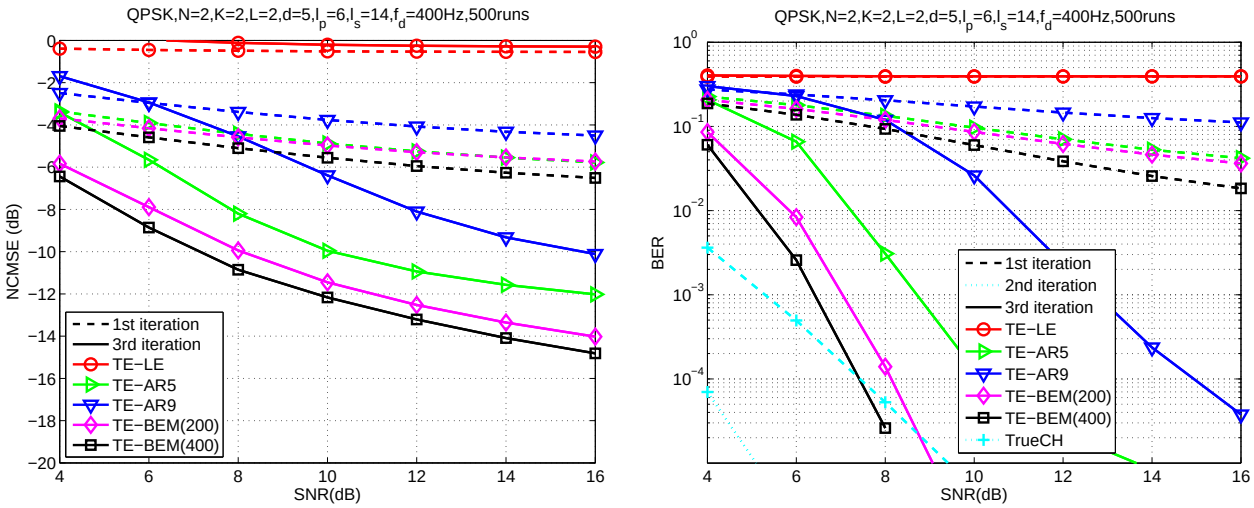
$$\sigma_k^2(n) = (K(\delta-1)+k, K(\delta-1)+k)\text{-th component of matrix } \mathbf{P}_e(n+\delta \mid n+\delta). \tag{5.89}$$

Computational Complexity

Here we consider computational complexity using the floating point operation (flop) counting for MIMO turbo equalization in this section. Since the adaptive SfiSfo equalizer has $(\delta+2)$ EKFs at time n as shown in Fig. 5.3, the computational complexity is given by $\mathcal{O}((\delta+2)J^2)$, where δ is the equalization delay and J is the size of state vector $\mathbf{x}(n)$; the computational complexity of a Kalman filtering is $\mathcal{O}(J^2)$. When based on AR(P) model [68], the computational complexity is $\mathcal{O}((\delta+2)\left[K\bar{\delta} + NKP(L+1)\right]^2)$ where $\bar{\delta}$ is given by (5.67). Note that it is independent of the constellation size $\mathcal{M}$. As we follow [68] 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 $\mathcal{O}((\delta+2)\left[K\bar{\delta} + NKQ(L+1)\right]^2)$ where Q is the number of basis functions in CE-BEM. Therefore, the proposed approach and the approach of [68] have comparable computational complexity if one takes $P = Q$. In addition, since the size of state vector using CE-BEM is given by (5.69), the computational complexity of our proposed approach is also independent of the constellation size $\mathcal{M}$. In the simulations presented in Section 5.3.3, 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 [68] were picked as 5 or 9 to attain comparable computational requirements for a fair performance comparison.

5.3.3 Simulation Examples



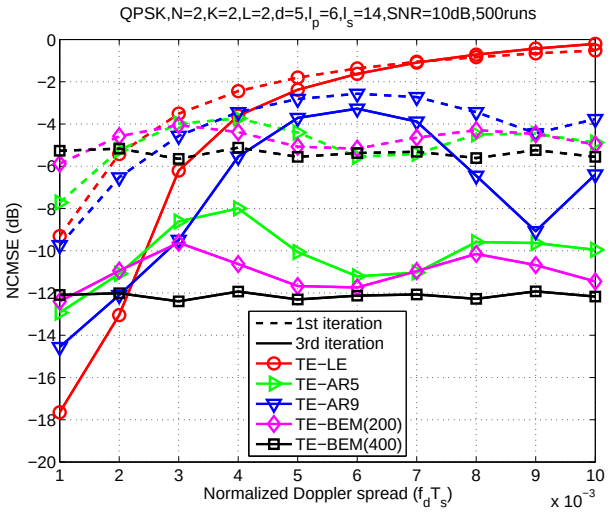
(a) NCMSE vs SNR

(b) BER vs SNR, with equalization delay $d = 5$

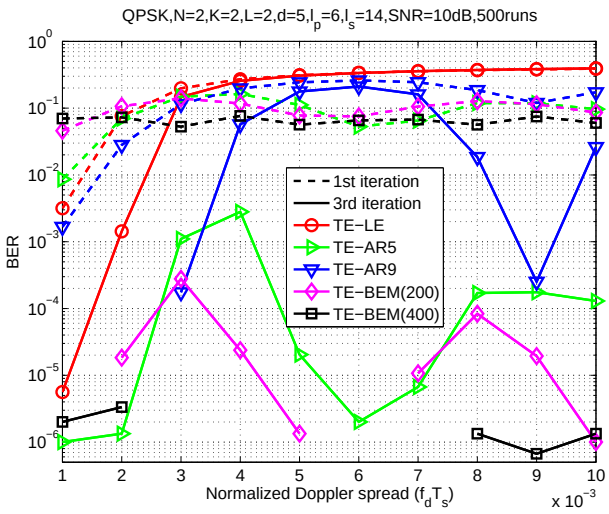
Figure 5.16: MIMO turbo equalization: performance comparison for SNR's under $N = K = 2, f_d T_s = 0.01, l_p = 6, l_s = 14$ (30% training overhead).

A random time- and frequency-selective multi-input multi-output (MIMO) Rayleigh fading channel is considered. We assume $\mathbf{h}(n;l)$ are zero-mean, complex Gaussian, and spatially white. We take $L = 2$ (3 taps) in (5.56), and $\sigma_h^2 = 1/(L+1)$. For different l's, $\mathbf{h}(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75] (with a correction in the appendix of [63]). 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 $f_d T_s$ 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 consider a simple two-receiver and two-user scenario, i.e., $N = 2, K = 2$ with the same transmitted power. 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



(a) NCMSE vs f_dT_s



(b) BER vs f_dT_s, with equalization delay $d = 5$

Figure 5.17: MIMO turbo equalization: performance comparison for normalized Doppler spread(f_dT_s)'s under $N = K = 2, \text{SNR} = 10\text{dB}, l_p = 6, l_s = 14$ (30% training overhead).

$\mathcal{M} = 4$ 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. We set $l_p = 6$ and $l_s = 14$ leading to 30% training overhead.

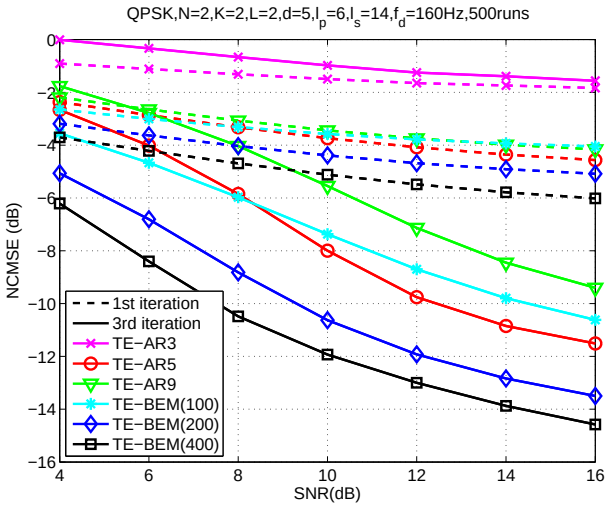
We compared the following schemes:

1. The approach of [48] that uses the linear MMSE equalizer (e.g. [38]) coupled with modified RLS channel estimation, where we set the linear filter length = 3 (6 pre-cursor taps and 3 post-cursor taps are used). This scheme is denoted by "TE-LE".

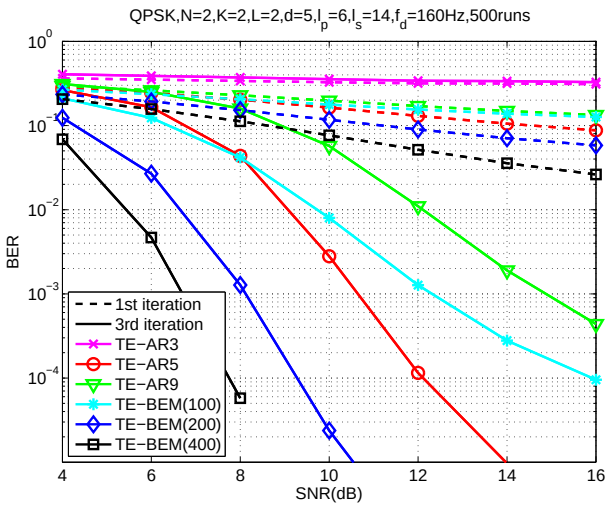
2. The AR(P) model-based scheme in [68]. The AR(P) model is as described in Section 2.2.1 and is fitted using [27] to Jakes' spectrum with f_dT_s=0.01 (the maximum anticipated normalized Doppler spread), denoted by "TE-AR5" for AR(5) model and "TE-AR9" for AR(9) model.

3. The proposed BEM-based turbo equalization schemes, where we consider BEM period $T = 200$ and 400 respectively, so that $Q = 5$ and 9, respectively, by (5.57). For the channel BEM coefficients, we take the AR-coefficient in (5.30) as $\alpha = 0.994$ for $T = 200$ and $\alpha = 0.996$ for $T = 400$. This scheme is denoted by "TE-BEM(200)" for $T = 200$ and "TE-BEM(400)" for $T = 400$.

4. The turbo equalizer based on the fixed-lag Kalman filter with perfect knowledge of the true channel, denoted by "TrueCH".



(a) NCMSE vs SNR



(b) BER vs SNR, with equalization delay $d = 5$

Figure 5.18: MIMO turbo equalization: performance comparison for SNR's under $N = K = 2$, $f_d T_s = 0.004, l_p = 6, l_s = 14$ (30% training overhead).

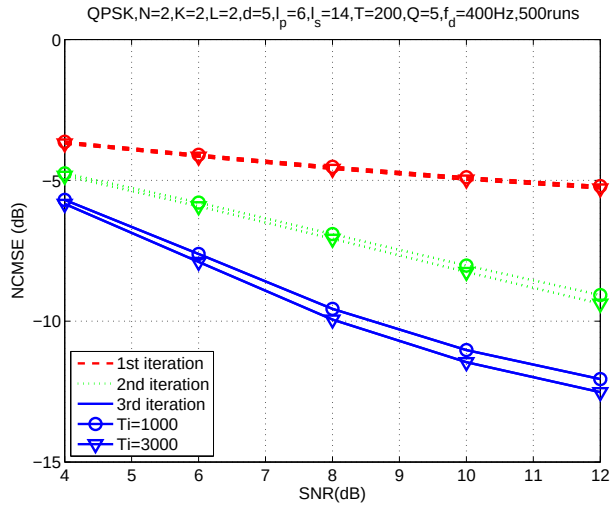
For the MIMO systems, unlike SISO system in Section 5.2.3, we do not have the turbo equalizer based on the optimum trellis-based MAP (BJCR) method [46] with perfect knowledge of the true channel due to the complexity.

We evaluate the performances of various schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

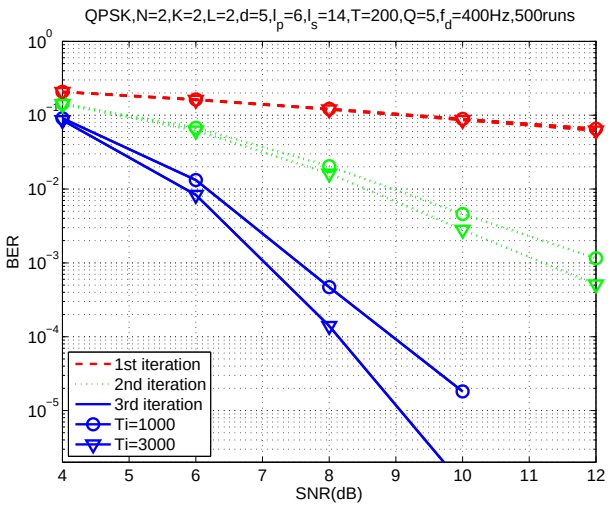
$$\text{NCMSE} := \frac{\sum_{i=1}^{M_r} \sum_{n=0}^{T_N-1} \sum_{l=0}^{L} \left\| \hat{\mathbf{h}}^{(i)}(n;l) - \mathbf{h}^{(i)}(n;l) \right\|^2}{\sum_{i=1}^{M_r} \sum_{n=0}^{T_N-1} \sum_{l=0}^{L} \left\| \mathbf{h}^{(i)}(n;l) \right\|^2}$$

where $\mathbf{h}^{(i)}(n;l)$ is the true channel and $\hat{\mathbf{h}}^{(i)}(n;l)$ is the estimated channel at the i-th Monte Carlo run, among total M_r runs. The BER's are evaluated by employing the equalization delay $\delta = 5$, using the decoded information symbol sequences at the turbo-equalization receiver. All the simulation results are based on 500 runs.

In Fig. 5.16, the performance of all the above schemes, under normalized Doppler spread $f_d T_s = 0.01$, are compared for different SNR's. In Fig. 5.17, they are compared over varying Doppler spread f_d's, under SNR = 10dB. Other settings of the simulation are same as Fig. 5.16, including the fact that $Q = 5$ (for $T = 200$) or $Q = 9$ (for $T = 400$), regardless of the actual f_d. Since the channel variations are well captured by the BEM coefficients, our proposed TE-BEM approach yields good performance even for "low" SNR's, which makes it possible to save the signal power to get the same BER performance. Note that TE-BEM with larger block size ($T = 400$) has a better performance than with smaller one ($T = 200$),



(a) NCMSE vs SNR



(b) BER vs SNR, with equalization delay $d = 5$

Figure 5.19: MIMO turbo equalization: performance comparison of TE-BEM(200) with different interleaver lengths for SNR's under $N = K = 2, f_dT_s = 0.01, l_p = 6, l_s = 14$ (30% training overhead).

since CE-BEM with $T = 400$ has the more basis functions in (5.57) and Kalman filtering utilize the past data implicitly (see Section 2.2.2). It is clear that the NCMSE and BER of turbo equalization using the CE-BEM for the channel tracking (TE-BEM) are not sensitive to the actual Doppler spread. Especially, TE-BEM with $T = 400, Q = 9$ seems so realizable over all the range of normalized Doppler spread. Therefore, we can neglect the actual Doppler spread of the channel for TE-BEM, esp., TE-BEM(400). With the comparable computational complexity, the performance of the symbol-wise AR(P)-based schemes are significantly worse than that of the proposed turbo equalization using CE-BEM. TE-AR5 is than that of TE-BEM(200) in Fig. 5.16 with increasing SNR for a fixed $f_dT_s = 0.01$, and is slightly worse in Fig. 5.17 for a fixed SNR of 10dB and varying Doppler spreads. On the other hand, TE-AR9 is significantly worse than that of TE-BEM(400) in Fig. 5.16 with increasing SNR for a fixed $f_dT_s = 0.01$ and in Fig. 5.17 for a fixed SNR of 10dB and varying Doppler spreads. For the same computational complexity, BEM models outperform AR models.

Note that while increasing the BEM period T (and corresponding Q) in CE-BEM improves performance, increasing the AR model order does not necessarily do so: we get inconsistent performance. A possible reason is that, as noted in [27], 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 [27] in choosing the regularization parameter for the matrix inversion involved. Such inconsistent behavior

is also seen in Fig. 5.18 where we compare performance of various schemes (including TE-BEM(100) with $T = 100$ and $Q = 3$, and AR3 with order $P = 3$) for different SNR's under normalized Doppler spread $f_d T_s = 0.004$. It is seen that increasing the BEM period T improves performance but increasing the AR model order does not necessarily do so.

For the linear MMSE equalizer coupled with modified RLS channel estimation (TE-LE), it performs well only for the small Doppler spread ($f_d T_s \leq 0.002$ in Fig. 5.17), since it uses the linear MMSE equalizer with separate channel estimation (not joint channel estimation and equalization). In Fig. 5.16, we present the performance of the turbo equalizer based on the fixed-lag Kalman filter with knowledge of the true channel (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. 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 [46].

In Fig. 5.19, we compare the performance of TE-BEM(200) for different SNR's, under normalized Doppler spread $f_d T_s = 0.01$ with two different interleaver size (equivalently different information block size). A smaller information block size ($T_i = 1000$) in BICM transmitter is considered leading to a coded block size 2000 bits and 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. It is seen that a smaller interleaver length results in a "small" deterioration in NCMSE and BER.

5.3.4 EXIT Chart Analysis

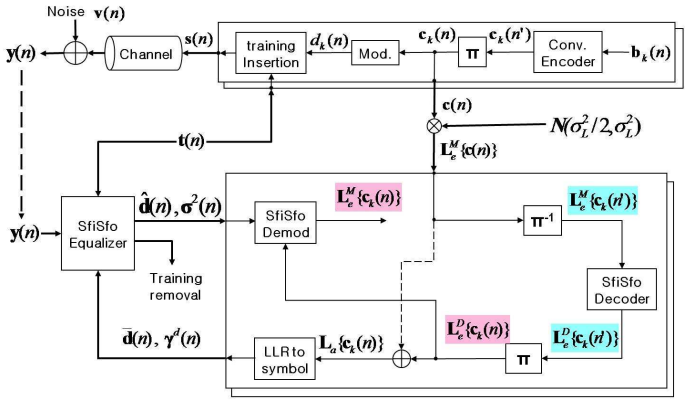


Figure 5.20: Simulation setup for generating extrinsic information transfer functions for MIMO system

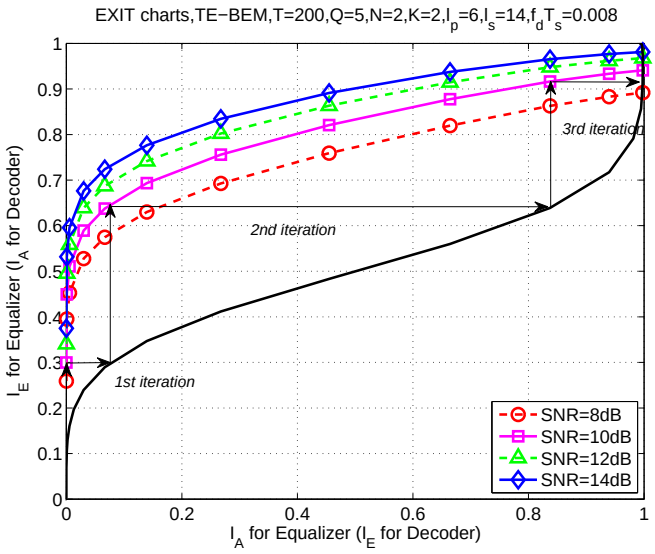
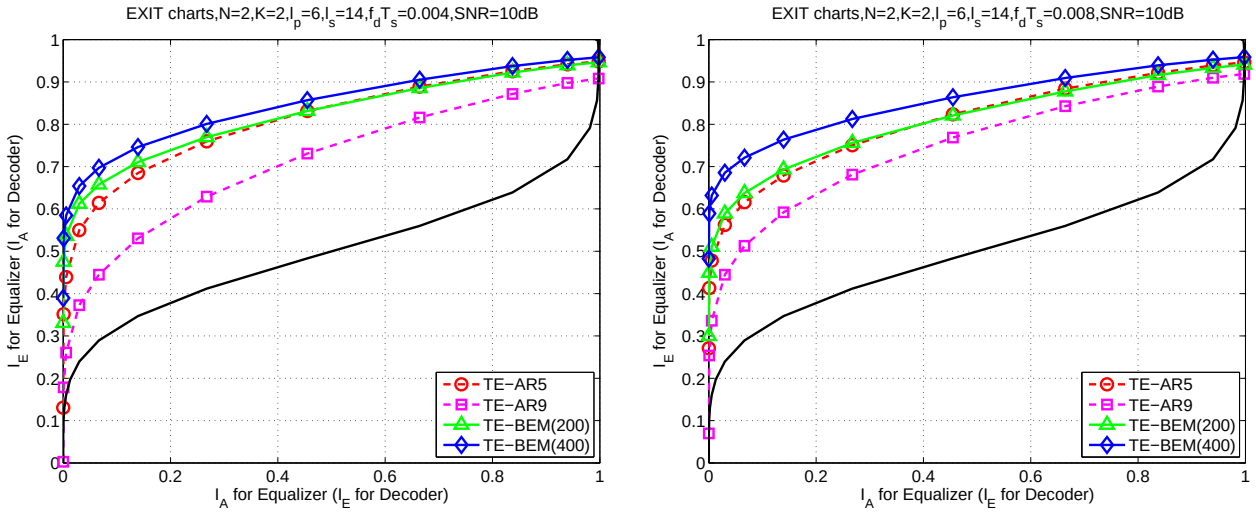


Figure 5.21: EXIT charts for different SNR's under $N = K = 2, f_d T_s = 0.008, l_p = 6, l_s = 14$ (30% training overhead).

The simulation setup to generate the extrinsic information transfer function is shown in Fig. 5.20. Note that the data symbol estimate and its error covariance for the k-th user, $\left\{\hat{d}_k(n), \sigma_k^2(n)\right\}$ is processed independently for the SfiSfo demodulation and the SfiSfo decoding at the receiver, as the input bit stream for k-th user, $\{\mathbf{b}_k(n)\}$ is processed independently of other users at the transmitter. For EXIT chart for the SfiSfo decoder, we observe $L_e^M\{c_k^i(n')\}$ (input LLR's to the SfiSfo decoder) and $L_e^D\{c_k^i(n')\}$ (output LLR's to the SfiSfo decoder). Following [59] (and others), $L_e^M\{c_k^i(n')\}$ is modeled as independent and identically distributed (i.i.d.) Gaussian with mean $c_k(n')\sigma_L^2/2$ 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 $L_e^M\{c_k^i(n')\}$, we can estimate I_e^M and I_e^D (the same for all channel models) via simulations. The interleaved random extrinsic LLR's $L_e^D\{c_k^i(n)\}$ are input to the "LLR to symbol" block in Fig. 5.20 together with the corresponding *a priori* LLR $L_e^M\{c_k^i(n)\}$, the input LLR's of the decoder, in order to obtain the *a posteriori* LLR's $L_a\{c_k^i(n)\}$. 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 $L_e^D\{c_k^i(n)\}$ (not $L_a\{c_k^i(n)\}$) and output LLR's $L_e^M\{c_k^i(n)\}$, respectively, of a given SfiSfo 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



(a) $f_dT_s = 0.004$ (b) $f_dT_s = 0.008$

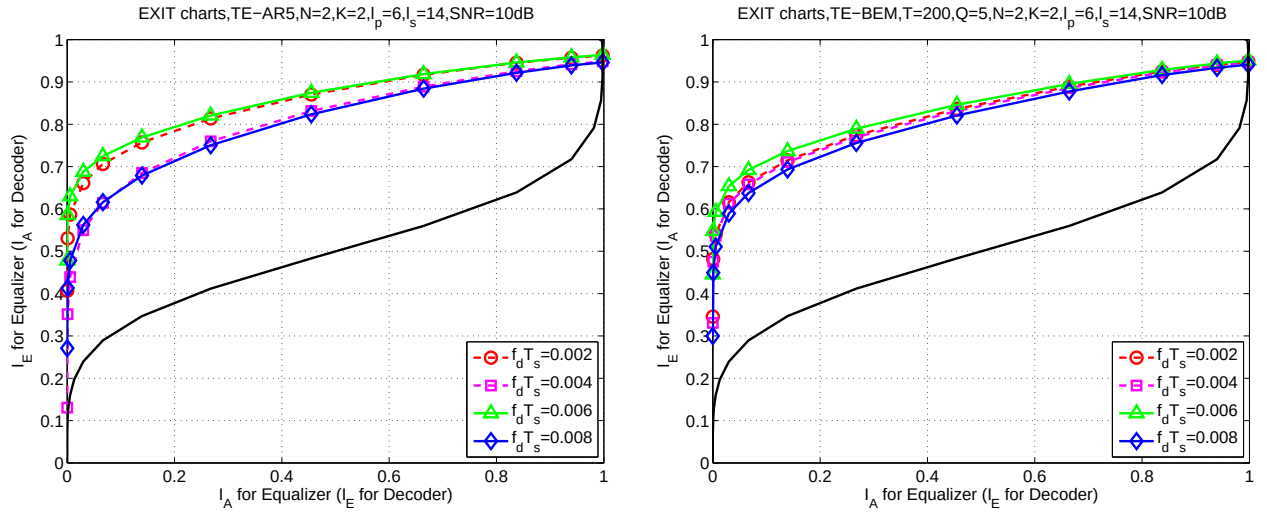
Figure 5.22: EXIT charts for TE-AR5, TE-AR9, TE-BEM(200) and TE-BEM(400) under $N = K = 2$, fixed f_dT_s, SNR = 10dB, $l_p = 6, l_s = 14$ (30% training overhead).

trace represents decoding task and the trajectory starts at the $(0, 0)$ point (see Fig. 5.21 for instance).

Using the set-up and parameters of Fig. 5.5 but with the information block size set to 30000 bits (the coded block size $T_r = 60000$), the normally distributed LLR's were generated with values of $\sigma_L^2 \in [10^{-2}, 10^2]$, and then I_e^M and I_e^D were calculated. We analyze the EXIT charts of our CE-BEM based approach with $T = 200$ and $Q = 5$ (TE-BEM(200) scheme) and the symbol-wise AR-model-based approach in [68] using AR(5) model (TE-AR5 scheme). In Fig. 5.21, EXIT charts for TE-BEM(200) are shown under a fixed normalized Doppler spread $f_dT_s = 0.008$ for different SNR's, where the output mutual information of the SfiSfo equalizer increases as SNR increases. We also show the trajectory trace for SNR=10 dB. In Fig. 5.22, EXIT charts for the symbol-wise AR(P)-based turbo equalizations (TE-AR5, TE-AR9) and the proposed ones using CE-BEM (TE-BEM(200), TE-BEM(400)) are compared under fixed $f_dT_s = 0.004$ and 0.008, SNR = 10dB. In Fig. 5.23, EXIT charts for TE-BEM(200) and TE-AR5 schemes are depicted under SNR = 10dB for different normalized Doppler spreads. One can estimate the actual simulation performances in Section 5.3.3 using the corresponding EXIT charts [Compare Fig. 5.23 and 5.22a with Fig. 5.17 and 5.18 respectively].

5.4 Conclusions

We proposed a turbo equalization receiver based on complex exponential basis expansion model (CE-BEM) for doubly selective fading channels, extending the single-user turbo



(a) TE-AR5: Turbo Equalization of [68] using AR(5) (b) TE-BEM(200): Turbo Equalization using CE-BEM with $T = 200, Q = 5$

Figure 5.23: EXIT charts for different f_dT_s's under $N = K = 2, \text{SNR} = 10\text{dB}, l_p = 6, l_s = 14$ (30% training overhead).

equalization approach of [68] 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 SfiSfo 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 SfiSfo decoder. While the BEM coefficients are updated via the autoregressive (AR) model, the time-varying nature of channels can be well captured by the CE-BEM, since the time-variations of the (unknown) BEM coefficients are likely much slower than those of the real channels. Unlike [68], 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 [68] for comparable computational complexity.

Chapter 6
Decision-Directed Tracking for Doubly-Selective Channel using Basis Expansion Models

We present a decision-directed tracking approach to doubly-selective channel estimation, exploiting the complex exponential basis expansion model (CE-BEM) for the overall channel variations, where we track the BEM coefficients rather than the channel tap gains, aided by the symbol decisions from a decision-feedback equalizer (DFE). 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. Two different schemes are investigated for BEM coefficient tracking based on the first-order autoregressive (AR) model for the BEM coefficients, including the Kalman filtering scheme and the extended exponentially-weighted (EW) recursive least-squares (RLS) algorithm. Simulation examples illustrate the superior performance of our approaches.

6.1 Introduction

Exploiting the detected information symbols as virtual training, decision-directed tracking is often utilized in both training-based and blind channel estimation approaches [6, 9, 21, 53]. Wireless channels, due to multipath propagation and Doppler spread, are characterized by frequency- and time-selectivity [60]. Accurate modeling of temporal evolution of the channel plays a crucial role for estimation and tracking purposes. 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 [16].

The AR model, however, does not perform well in predicting a (fast-varying) channel, which is often required in decision-directed schemes. In fast-fading environments, channel prediction using an AR model may lead to high estimation variance resulting in erroneous symbol decisions [6]. With these incorrect decisions, channel estimation error can be further worsened, inducing error propagation and even a breakdown in symbol detection [53]. Since decision-directed approaches are sensitive to error propagation, more accurate channel modeling becomes necessary for tracking fast-fading channels.

In contrast to AR models that describe channel variations on a symbol-by-symbol update basis, a basis expansion model (BEM) depicts evolutions of the channel over a period (block) of time, where the time-varying channel taps are expressed as superpositions of time-varying

basis functions in modeling Doppler effects, weighted by time-invariant coefficients [11]. Candidate basis functions include complex exponential (Fourier) functions [11, 70] leading to CE-BEM. Since the time-varying nature of the channel can be well captured in a BEM by the known basis functions, the time-variations of the (unknown) BEM coefficients are likely much slower than that of the channel, and thus more convenient to track in a fast-fading environment.

In Chapter 3, a subblock-wise tracking approach was proposed for doubly-selective channels using time-multiplexed (TM) training. It exploits the CE-BEM for the overall channel variations, and a first-order AR model to describe the evolutions of the BEM coefficients. The slow-varying BEM coefficients, rather than the fast-varying channel, are tracked and updated at each training session; during information sessions, channel estimates are generated by the CE-BEM using the estimated BEM coefficients. The BEM coefficients are updated via Kalman filtering at each training session.

Based on the subblock approach (as opposed to blockwise approaches of [8, 26, 41, 44]), decision-directed tracking of doubly-selective channels is investigated. Acting as virtual training symbols, the information symbol decisions, provided by a DFE (with delay $d \geq 0$) utilizing the estimated channel, are used to enhance the estimation of the BEM coefficients, so that much of the spectrum resource allocated to training can be saved. Although a time gap still exists between the available symbol decisions and the channel estimates required by the DFE, it can be successfully bridged by the CE-BEM-based channel prediction, without incurring much estimation variance.

Decision-directed channel tracking using a polynomial BEM has been investigated in [8], where the BEM coefficients are updated via the recursive least-squares (RLS) algorithm within a sliding window; channel estimation using Kalman filtering and polynomial or complex exponential BEM for OFDM systems has been explored in [26, 41, 44], among which decision-directed tracking is considered in [26, 41]. Our decision-directed scheme updates the BEM coefficients of much smaller size of (sub)block than BEM period. The periodic training symbols are still necessary to recover the channel tracking from possible phase ambiguity due to the error propagation.

Two different decision-directed schemes are investigated exploiting CE-BEM for the channel variations: the Kalman filtering and the extended EW-RLS algorithm. We assume that the BEM coefficients (rather than the time-varying channel tap gains) follow a first-order AR model. The BEM coefficients are tracked and used to generate the channel estimates by the CE-BEM. The RLS filter is only a special case of a Kalman filter and an extended RLS scheme is developed with enhanced tracking abilities. These approaches have superior performances in both channel estimation variance and bit error rate (BER) over the scheme

proposed by [6] that relies solely on the AR model. Compared with the approaches in Chapters 3 and 4, the decision-directed tracking scheme in this chapter achieves better BER performances with much less training overhead.

6.2 Decision-Directed Tracking using CE-BEM

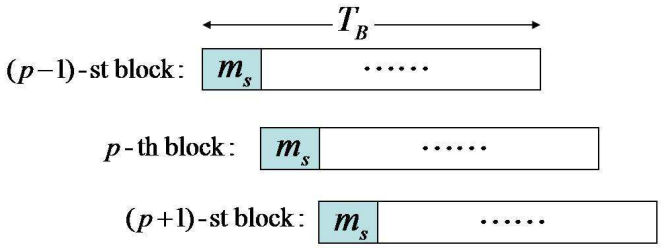


Figure 6.1: Decision-directed tracking: overlapping blocks that differ by a small step size m_s.

Consider two overlapping blocks (each consisting of T_B symbols) that differ by only m_s ($1 \leq m_s \ll T_B$) symbols as Fig. 6.1: 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 two blocks, one can expect the BEM coefficients $\mathbf{h}_q(l)$'s in CE-BEM (6.2) to vary only a little from the past block to the current overlapping one. Therefore, we can estimate channel by updating the BEM coefficients every m_s symbols, rather than the anew at every non-overlapping block as in [70]. Intuitively, with a smaller step size m_s, the BEM coefficients should vary more slowly. At the receiver, we employ a MMSE-DFE in Section 2.4.2 [40] with delay $d \geq 0$ to equalize the estimated channel. In decision-directed tracking, its output symbol decisions are fed-back to the channel estimator as pseudo-trainings. Note that we need to predict the channel up to time n to detect symbol for $\hat{s}(n-d)$ at the DFE. In CE-BEM (6.2), we can use the BEM coefficients $\mathbf{h}_q(l)$'s provided by a channel estimator for the current step size m_s to predict the channel $\hat{\mathbf{h}}(n;l)$ for the following $m_s + d$ symbols since they are within the same p-th overlapping block; the estimated BEM coefficients are shared for T_B symbols starting $n = pm_s$ in CE-BEM. In this chapter, we propose to update the BEM coefficients in CE-BEM every m_s symbols via Kalman filtering or EW-RLS algorithm for the decision-directed tracking.

Consider a doubly-selective (time- and frequency-selective) single-input multi-output (SIMO), finite impulse response (FIR) linear channel with N outputs. Let $\{s(n)\}$ denote a scalar information sequence that is input to the time-varying channel with discrete-time response $\{\mathbf{h}(n;l)\}$ (N-column vector channel response at time instance n to a unit input at

time instance $n-l$). We assume $\{s(n)\}$ is independent and identically distributed (i.i.d.) with zero mean and variance $E\{|s\,(n)\,|^2\} = \sigma_s^2$. Then the symbol-rate noisy N-column channel output vector is given by $(n = 0, 1, \ldots)$

$$\mathbf{y}\,(n) = \sum_{l=0}^{L} \mathbf{h}\,(n; l)\, s\,(n-l) + \mathbf{v}\,(n) \tag{6.1}$$

where the N-column vector $\mathbf{v}\,(n)$ is zero-mean, white, uncorrelated with $s\,(n)$, complex Gaussian noise, with the autocorrelation $E\{\mathbf{v}\,(n+\tau)\,\mathbf{v}^H\,(n)\} = \sigma_v^2 \mathbf{I}_N \delta\,(\tau)$. We assume that $\{\mathbf{h}\,(n; l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [60].

In CE-BEM [11, 14, 70], over the i-th block consisting of an observation window of T_B symbols, the channel is represented as $(n = (i-1)T_B, (i-1)T_B + 1, \cdots, iT_B - 1$ and $l = 0, 1, \cdots, L\,)$

$$\mathbf{h}(n; l) = \sum_{q=1}^{Q} \mathbf{h}_q(l) e^{j\omega_q n}, \tag{6.2}$$

where $\mathbf{h}_q(l)$ is the N-column time-invariant BEM coefficient vector and one chooses (K is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{6.3}$$

$$Q \geq 2\,\lceil f_d T T_s \rceil + 1, \tag{6.4}$$

$$\omega_q := \frac{2\pi}{T}\,[q - (Q+1)\,/2]\,, \quad q = 1, 2, \cdots, Q \tag{6.5}$$

$$L := \lfloor \tau_d / T_s \rfloor\,, \tag{6.6}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration. The BEM coefficients $\mathbf{h}_q(l)$'s remain invariant during this block, but are allowed to change at the next consecutive block; the Fourier basis functions $\{e^{j\omega_q n}\}$ $(q = 1, 2, \cdots, Q)$ are common for every block. If the delay spread and the Doppler spread (or at least their upper bounds) are known, one can infer the basis functions of the CE-BEM [70]. Treating the basis functions as known, estimation of a time-varying process is reduced to estimating the invariant coefficients over a block of T_B symbols.

Stack the BEM coefficients in (6.2) into "tall" vectors as

$$\mathbf{h}^{(l)} := \begin{bmatrix} \mathbf{h}_1^{(l)T} & \mathbf{h}_2^{(l)T} & \cdots & \mathbf{h}_Q^{(l)T} \end{bmatrix}^T \tag{6.7}$$

$$\mathbf{h} := \begin{bmatrix} \mathbf{h}^{(0)T} & \mathbf{h}^{(1)T} & \cdots & \mathbf{h}^{(L)T} \end{bmatrix}^T \tag{6.8}$$

of size NQ and $M := NQ(L+1)$ respectively. The coefficient vectors in (6.7) and (6.8) of the p-th overlapping block will be denoted by $\mathbf{h}^{(l)}(p)$, and $\mathbf{h}(p)$. Again, we emphasize that the p-th block and the $(p+1)$-st block differ by just m_s symbols. Since a fading channel can follow well a Markov model [16], we further assume that the BEM coefficients over each overlapping block are also Markovian. A simplified formulation is given by the first-order AR model, i.e.,

$$\mathbf{h}(p) = \mathbf{A}_1 \mathbf{h}(p-1) + \mathbf{w}(p), \tag{6.9}$$

where $\mathbf{A}_1 = \alpha \mathbf{I}_M$ is the first-order AR coefficient, and the driving noise vector $\mathbf{w}(p)$ is zero-mean complex Gaussian with variance $\sigma_w^2 \mathbf{I}_M$ and statistically independent of $\mathbf{h}(p-1)$. [We have already investigated the theoretical choice for the first-order AR coefficient in Section 3.4, whose results apply here after replacing the subblock size m_b with the smaller step size m_s.] Assuming the channel is wide-sense stationary (WSS) and the BEM coefficients $h_q(l)$'s are independent, we have

$$\sigma_w^2 = \sigma_h^2 (1 - |\alpha|^2)/Q \tag{6.10}$$

where $\sigma_h^2 \mathbf{I}_M := E\left\{\mathbf{h}(n;l)\,\mathbf{h}^H(n;l)\right\}$. [Typically $\alpha < 1$ but close to one. Strictly speaking, one should try a "full" matrix $\mathbf{A}_1$ in (6.9); however, it can only be calculated if all the channel statistics are known — we do not assume such knowledge here, just an upperbound on the Doppler and multipath delay spreads.]

Define

$$\begin{aligned} \boldsymbol{\mathcal{E}}(n) &:= \left[e^{-j\omega_1 n} \quad e^{-j\omega_2 n} \quad \cdots \quad e^{-j\omega_Q n}\right]^T, \\ \mathbf{s}(n) &:= \left[s(n) \quad s(n-1) \quad \cdots \quad s(n-L)\right]^T. \end{aligned} \tag{6.11}$$

For $pm_s \le n < (p+1)m_s$, by (6.1), (6.2), (6.7) and (6.8), the received signal at time n can be written as

$$\mathbf{y}(n) = \left[\mathbf{s}(n) \otimes \mathbf{I}_N\right]^T \left[\mathbf{I}_{L+1} \otimes \left(\boldsymbol{\mathcal{E}}(n) \otimes \mathbf{I}_N\right)\right]^H \mathbf{h}(p) + \mathbf{v}(n).$$

Further defining

$$\mathbf{C}_i(p) := \left[\mathbf{s}(pm_s + i) \otimes \mathbf{I}_N\right]^T \left[\mathbf{I}_{L+1} \otimes \left(\boldsymbol{\mathcal{E}}(pm_s + i) \otimes \mathbf{I}_N\right)\right]^H, \tag{6.12}$$

$$\mathbf{C}(p) := \left[\mathbf{C}_0^T(p) \quad \mathbf{C}_1^T(p) \quad \cdots \quad \mathbf{C}_{m_s-1}^T(p)\right]^T, \tag{6.13}$$

we have

$$\mathbf{y}_{ms}(p) = \mathbf{C}(p)\,\mathbf{h}(p) + \mathbf{v}_{ms}(p) \tag{6.14}$$

where

$$\mathbf{y}_{ms}(p) := \left[\mathbf{y}^T(pm_s) \quad \mathbf{y}^T(pm_s+1) \quad \cdots \quad \mathbf{y}^T((p+1)m_s-1)\right]^T$$

and $\mathbf{v}_{ms}(p)$ is defined likewise.

6.2.1 Decision-Directed Kalman Tracking

The dynamical state-space system model for Kalman filtering is represented by (6.9) and (6.14), where $\mathbf{y}_{ms}(p)$ is the measurement of the state of BEM coefficients $\mathbf{h}(p)$. Kalman filtering can be applied to track the coefficient vector $\mathbf{h}(p)$ at a step size of m_s symbols. Since $\mathbf{s}(n)$ is unknown during information sessions, the receiver then switches to a decision-directed mode, in which the past symbol decisions are assumed to be correct and used in Kalman tracking. Treating (6.9) and (6.14) as the state and the measurement equations respectively, the Kalman tracking in the training mode is initialized with

$$\hat{\mathbf{h}}(-1 \mid -1) = \mathbf{0}_M \text{ and } \mathbf{R}_h(-1 \mid -1) = \sigma_w^2 \mathbf{I}_M,$$

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

$$\mathbf{R}_h(p \mid m) := E\{[\hat{\mathbf{h}}(p \mid m) - \mathbf{h}(p)][\hat{\mathbf{h}}(p \mid m) - \mathbf{h}(p)]^H\}.$$

Kalman recursive filtering (for $p = 0, 1, \cdots$) is applied via the following steps [32]:

1. Time update:

$$\begin{aligned}\hat{\mathbf{h}}(p \mid p-1) &= \alpha \hat{\mathbf{h}}(p-1 \mid p-1),\\ \mathbf{R}_h(p \mid p-1) &= |\alpha|^2 \mathbf{R}_h(p-1 \mid p-1) + \sigma_w^2 \mathbf{I}_M;\end{aligned}$$

2. Kalman gain:

$$\begin{aligned}\mathbf{R}_\eta(p) &= \mathbf{C}(p)\mathbf{R}_h(p \mid p-1)\mathbf{C}^H(p) + \sigma_v^2 \mathbf{I}_{Nm_s},\\ \mathbf{G}(p) &= \mathbf{R}_h(p \mid p-1)\mathbf{C}^H(p)\mathbf{R}_\eta^{-1}(p);\end{aligned}$$

3. Measurement update:

$$\begin{aligned}\hat{\mathbf{h}}(p \mid p) &= \hat{\mathbf{h}}(p \mid p-1) + \mathbf{G}(p)\left[\mathbf{y}_{ms}(p) - \mathbf{C}(p)\hat{\mathbf{h}}(p \mid p-1)\right],\\ \mathbf{R}_h(p \mid p) &= [\mathbf{I}_M - \mathbf{G}(p)\mathbf{C}(p)]\mathbf{R}_h(p \mid p-1).\end{aligned}$$

In our simulations, we take the noise variance to be $\sigma_v^2 + 0.01\sigma_s^2$ instead of σ_v^2 to compensate the symbol decision errors.

6.2.2 Decision-Directed EW-RLS Tracking

Since the CE-BEM (6.2) is periodic with a period T, the memory of the algorithm should be less than T to avoid periodicity of the model influencing the estimation results as the actual channel is not periodic. Therefore, adaptive algorithms with finite memory are preferred. A "finite-memory" algorithm is considered in decision-directed tracking: the extended exponentially-weighted recursive least-square (EW-RLS) algorithm. Assuming the BEM coefficients follow the first-order AR model, a state-space model for the extended EW-RLS algorithm is given by (6.9) and (6.14). [As in Chapter 4, one can use the "normal" EW-RLS algorithm with no *a priori* model for BEM coefficients. However, we adopt the extended EW-RLS algorithm because it turns out AR(1) modeling for the BEM coefficients limits the error propagation caused by the incorrect decision-feedbacks which are more severe than the modeling errors (especially in low SNR's). Note that subblock-wise channel estimation in Chapter 4 uses only the training symbols for the channel tracking.]

Based on (6.9) and (6.14), we can apply the extended exponentially-weighted regularized RLS (EW-RLS) algorithm [2, Section 12.B.(12.B.18)] to track an unknown BEM coefficient vector $\mathbf{h}(p)$. Choose $\mathbf{h}(p)$ to minimize the cost function

$$\lambda^{p+1}\beta \left\|\mathbf{h}\right\|^2 + \sum_{i=0}^{p} \lambda^{p-i} \left\|\mathbf{y}_{m_s}(i) - \mathbf{C}(i)\mathbf{h}\right\|^2, \tag{6.15}$$

where $\beta > 0$ is a regularization parameter and $0 < \lambda < 1$ is the forgetting factor. [This is the general cost function of the exponentially-weighted regularized RLS. See the details in [2, Section 12.B] about the equivalent cost function and its solution for the extended exponentially-weighted regularized RLS that is subject to the state equation in (6.9).] Note that the forgetting factor is used to give more weight to recent data and less weight to past data. We take λ to be close to one for updating the small step size m_s.

Mimicking [2, Algorithm 12.B.1], the extended EW-RLS tracking is applied via the following steps :

1. Initialization:

$$\hat{\mathbf{h}}(-1) = \mathbf{0}_{M\times 1} \text{ and } \mathbf{P}(-1) = \beta^{-1}\mathbf{I}_M$$

2. RLS recursion: For $p = 0, 1, \cdots$

$$\begin{aligned}
\mathbf{\Gamma}(p) &= \lambda \mathbf{I}_{Nm_s} + \mathbf{C}(p)\mathbf{P}(p-1)\mathbf{C}^H(p), \\
\mathbf{G}(p) &= \mathbf{P}(p-1)\mathbf{C}^H(p)\mathbf{\Gamma}^{-1}(p), \\
\hat{\mathbf{h}}(p) &= \alpha\hat{\mathbf{h}}(p-1) + \alpha\mathbf{G}(p)\left[\mathbf{y}_{m_s}(p) - \mathbf{C}(p)\hat{\mathbf{h}}(p-1)\right], \\
\mathbf{P}(p) &= \lambda^{-1}|\alpha|^2\left[\mathbf{I}_M - \mathbf{G}(p)\mathbf{C}(p)\right]\mathbf{P}(p-1) + \sigma_w^2\mathbf{I}_M
\end{aligned}$$

where $\hat{\mathbf{h}}(p)$ denotes the estimate of $\mathbf{h}(p)$ given the observations $\{\mathbf{y}_{m_s}(0), \mathbf{y}_{m_s}(1), \cdots, \mathbf{y}_{m_s}(p)\}$.

After channel estimation for every p using Kalman filtering or RLS algorithm, we can generate the channel by the estimated $\hat{\mathbf{h}}(p)$ via the CE-BEM (6.2). In order to bridge the time gap between the symbol decisions and the channel estimates required by a MMSE-DFE with delay $d \geq 0$, we estimate the channel for the current m_s symbols as well as following $m_s + d$ symbols, i.e., the channel estimates in our receiver are given by

$$\hat{\mathbf{h}}(n; l) = \boldsymbol{\mathcal{E}}^H(n)\hat{\mathbf{h}}^{(l)}(p), \tag{6.16}$$

for $n = pm_s, pm_s + 1, \cdots, (p+2)m_s + d - 1$. The definition of $\hat{\mathbf{h}}^{(l)}(p)$ is similar to (6.7) and $\hat{\mathbf{h}}^{(l)}(p)$ is based on observations up to time $n = (p+1)m_s - 1$.

Using the channel estimates up to time $(p+2)m_s + d - 1$, the symbol detections are made by DFE for the time $n = pm_s, pm_s + 1, \cdots, (p+2)m_s - 1$. Note that the detected symbols for the time $n = (p+1)m_s, (p+1)m_s + 1, \cdots, (p+2)m_s - 1$ are used for the following channel tracking, whereas the "current" decisions for the time $n = pm_s, pm_s + 1, \cdots, (p+1)m_s - 1$ are updated for improved performance.

Computational Complexity

We compare the computational complexity using the floating point operation (flop) count for the decision-directed schemes. The detailed flops counts for one iteration of Kalman filtering are already shown in Section 3.3.1 (EW-RLS algorithm in Section 4.3.1) and DFE in Table 6.1 respectively. We consider three different channel estimation schemes: "SB-KF-BEM" denoting BEM-based subblock-wise Kalman filtering in Section 3, "DD-KF-ARP" denoting AR(P) model-based symbol-by-symbol decision-directed Kalman filtering [6], "DD-KF-BEM" denoting BEM-based decision-directed Kalman filtering with different step sizes $m_s = 1, 2$ and 4. All the above schemes have the same subblock ($m_b = 100$) and hence same training overhead. We take $T = 400$ and $Q = 9$ for CE-BEM. [Since the complexities of Kalman filtering and EW-RLS algorithm are (almost) the same, we consider Kalman

filter only to compare the proposed decision-directed approach with others under the same environment.]

The number of flops for one cycle of subblock-wise Kalman filtering, decision-directed Kalman filtering and DFE turn out to be

$$\begin{aligned} f_1(\text{SB-KF}) =& M_s^3 + M_s^2(3L_p+3) + M_s(2L_p^2+5L_p+1) + L_p^3/3 + L_p^2 \\ f_1(\text{DD-KF}) =& M^3 + M^2(3Nm_s+2) + M[2(Nm_s)^2+3Nm_s+1] + (Nm_s)^3/3 + (Nm_s)^2 \\ f_1(\text{DFE}) =& 2(Nl_f)^3 + (Nl_f)^2(\bar{l}_f+\bar{l}_b+2) + (Nl_f)(\bar{l}_f\bar{l}_b+\bar{l}_b^2+\bar{l}_b+1) + 2(\bar{l}_b^3+\bar{l}_b^2+\bar{l}_b) + l_b \end{aligned}$$

where $M_s = Q(L+1)$ for SB-KF-BEM, $M = NP(L+1)$ for DD-KF-ARP or $NQ(L+1)$ for DD-KF-BEM, and $L_p = L+1, \bar{l}_f = l_f + L, \bar{l}_b = l_b + 1$. The overall flop counts over one block of T_B symbols for the subblock-wise and decision-directed Kalman tracking schemes are then given by

$$\begin{aligned} f_c(\text{SB-KF-BEM}) &= N(T_B/m_b)f_1(\text{SB-KF}) + [T_B - m_t(T_B/m_b)]f_1(\text{DFE}) \\ f_c(\text{DD-KF-AR}P) &= T_B f_1(\text{DD-KF}) + 2[T_B - m_t(T_B/m_b)]f_1(\text{DFE}), m_s = 1 \\ f_c(\text{DD-KF-BEM}) &= (T_B/m_s)f_1(\text{DD-KF}) + 2[T_B - m_t(T_B/m_b)]f_1(\text{DFE}). \end{aligned}$$

The comparative flop counts for each scheme with one receiver ($N = 1$) are compared in Table 6.2 over one block ($T_B = 200$). Note that DD-KF-BEM with $m_s = 1$ has the same complexity as DD-KF-ARP since we select $P = Q = 9$.

Table 6.1: DFE: flop count for one cycle.

Operation	flops (with $\sigma_s^2 = 1$)
$\mathbf{R}_{yy} = \mathbf{H}(n)\mathbf{H}^H(n) + \sigma_v^2\mathbf{I}_{Nl_f}$	$(Nl_f)^2(\bar{l}_f+1)$
$\mathbf{R}_{sy} = \mathbf{\Phi}\mathbf{H}^H(n)$	$(Nl_f)\bar{l}_f\bar{l}_b$
$\mathbf{R}_\delta = \mathbf{I}_{\bar{l}_b} - \mathbf{R}_{sy}(n)\mathbf{R}_{yy}^{-1}(n)\mathbf{R}_{sy}^H(n)$	$(Nl_f)^3 + (Nl_f)^2\bar{l}_b + (Nl_f)\bar{l}_b^2$
$\mathbf{b}_{\text{MMSE}}(n) = \mathbf{R}_\delta^{-1}\mathbf{e}_0\left(\mathbf{e}_0^T\mathbf{R}_\delta^{-1}\mathbf{e}_0\right)^{-1}$	$2(\bar{l}_b^3+\bar{l}_b^2+\bar{l}_b)$
$\mathbf{f}_{\text{MMSE}}(n) = \mathbf{R}_{yy}^{-1}(n)\mathbf{R}_{sy}^H(n)\mathbf{b}_{\text{MMSE}}(n)$	$(Nl_f)^3 + (Nl_f)^2 + (Nl_f)\bar{l}_b$
$\hat{s}(n-d) = \sum_{m=0}^{l_f-1}\mathbf{f}_m^T(n)\mathbf{y}(n-m) - \sum_{k=1}^{l_b}b_k(n)\tilde{s}(n-d-k)$	$Nl_f + l_b$
Total : $2(Nl_f)^3 + (Nl_f)^2(\bar{l}_f+\bar{l}_b+2) + (Nl_f)(\bar{l}_f\bar{l}_b+\bar{l}_b^2+\bar{l}_b+1) + 2(\bar{l}_b^3+\bar{l}_b^2+\bar{l}_b) + l_b$	

Table 6.2: Subblock-wise and decision-directed tracking with DFE: comparative flop count over one block of T_B symbols.

tracking schemes	comparative flops
SB-KF-BEM	1
DD-KF-BEM, $m_s = 4$	4.78
DD-KF-BEM, $m_s = 2$	6.78
DD-KF-BEM, $m_s = 1$	10.86
DD-KF-AR9	10.86

6.2.3 Simulation Examples

A random time- and frequency-selective Rayleigh fading channel is considered. We assume $\mathbf{h}(n;l)$ are zero-mean, complex Gaussian, and spatially white with autocorrelation $E\left\{\mathbf{h}(n;l)\,\mathbf{h}^H(n;l)\right\} = \sigma_h^2\mathbf{I}_N$ for each l. We take $L = 2$ (3 taps) in (6.1), and $\sigma_h^2 = 1/(L+1)$. For different l's, $\mathbf{h}(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75] (with a correction in the appendix of [63]).

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 $f_d = 400\,$Hz, or the normalized Doppler spread $f_dT_s = 0.01$ (corresponding to a maximum mobile velocity 216km / h). 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.

We evaluate the performances of various schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

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

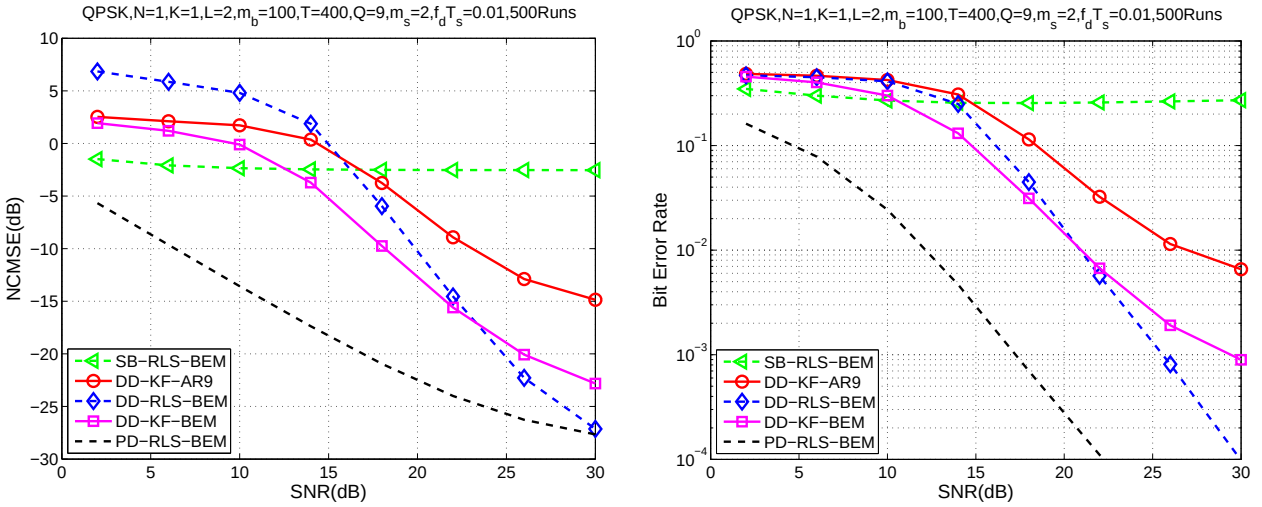
where $\mathbf{h}^{(i)}(n;l)$ is the true channel and $\hat{\mathbf{h}}^{(i)}(n;l)$ is the estimated channel at the i-th Monte Carlo run, among total M_r runs. In each run, a training mode of 200 binary phase-shift keying (BPSK) symbols is followed by a decision-directed mode of quadrature phase-shift keying (QPSK) 4000 symbols ($T_N = 4000$). All the simulation results are based on 500 Monte Carlo 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 [70] is adopted, where each subblock of equal length m_b symbols consists 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 for each user contains

an impulse guarded by zeros (silent periods), which has the structure ($\gamma > 0$)

$$\mathbf{c}_p := \begin{bmatrix} \mathbf{0}_{1\times L} & \gamma & \mathbf{0}_{1\times L} \end{bmatrix}. \tag{6.17}$$

Therefore, $m_t = 2L+1 = 5$, which have to be 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 at every training session given by (6.17), we set $\gamma = \sqrt{2L+1} = \sqrt{5}$ so that the average power per symbol at training sessions is equal to that of information sessions. We consider a large subblock size $m_b = 100$ with less frequent training symbols, comparing a small subblock size $m_b = 40$ with frequent training symbols. For the CE-BEM, we take $T = 400$ and $Q = 9$ with $f_d T_s = 0.01$.



(a) NCMSE vs SNR

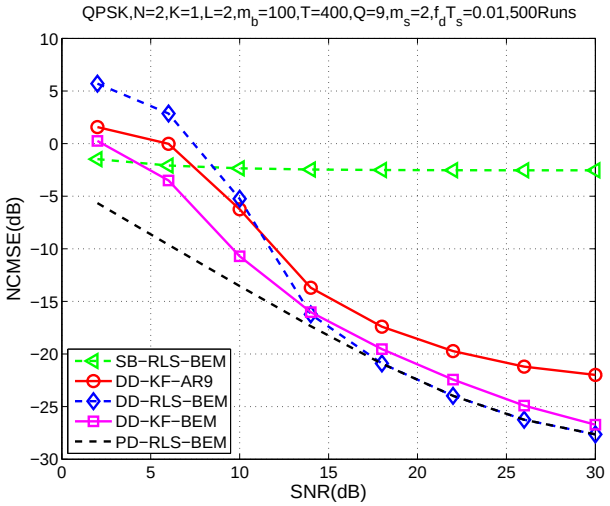
(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.2: Decision-directed tracking: performance comparison for SNR's, under $N = 1, f_d T_s = 0.01, m_b = 100$.

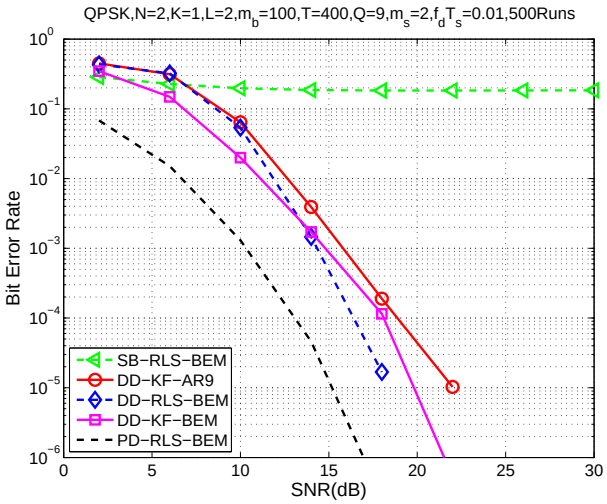
We compared the following schemes:

1. The subblock-wise EW-RLS algorithm with $\beta = 1$, which is one of the best subblock-wise approaches in Chapter 4. [One can pick a different subblock-wise scheme using SW-RLS algorithm or Kalman filter that has similar (or slightly worse) performances.] In the simulations, we take the forgetting factor $\lambda = 0.5$ for $m_b = 100$ in EW-RLS algorithm. This scheme is denoted by "SB-RLS-BEM".

2. The decision-directed channel tracking scheme in [6] using the P-th order AR model, i.e., the time-varying channel follows (assuming each independent channel tap has same



(a) NCMSE vs SNR



(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.3: Decision-directed tracking: performance comparison for SNR's, under $N = 2, f_d T_s = 0.01, m_b = 100$.

Doppler spread)

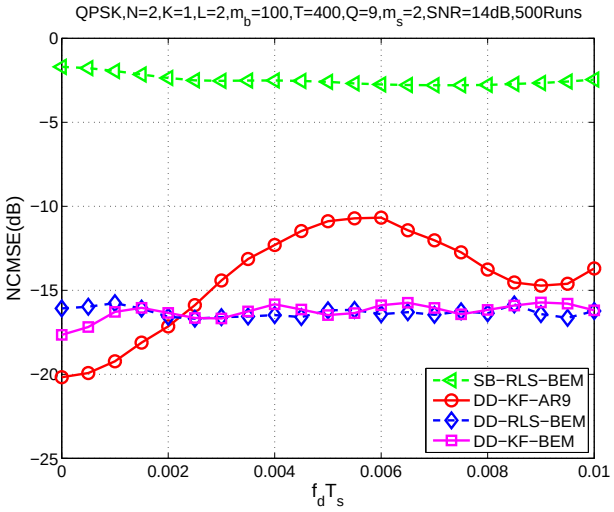
$$\mathbf{h}\left(n;l\right) = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{h}\left(n-i;l\right) + \mathbf{w}\left(n\right), \tag{6.18}$$

where $\mathbf{A}_i = \alpha_i \mathbf{I}$ is the AR coefficients matrix and driving noise $\mathbf{w}(n)$ is zero-mean complex Gaussian with autocorrelation, $E\left\{\mathbf{w}\left(n\right)\mathbf{w}^H\left(n\right)\right\} = \sigma_w^2 \mathbf{I}_N$. Using Yule-Walker equations, we "fit" the AR coefficients and noise variance for $f_d T_s = 0.01$. The channel prediction stage is conducted by using (6.18) omitting the driving noise, i.e.,

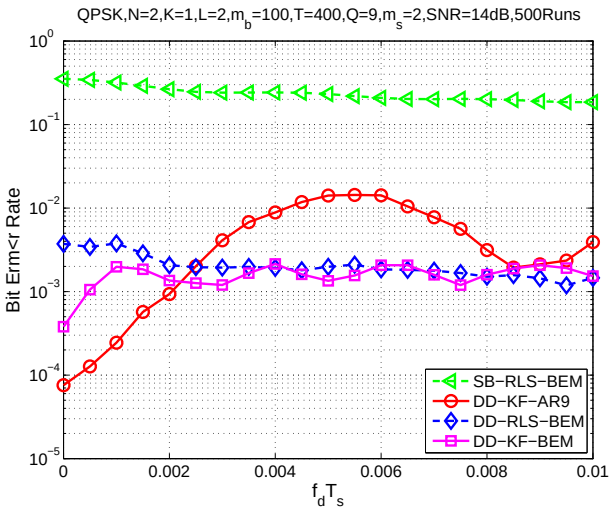
$$\hat{\mathbf{h}}\left(n;l\right) = \sum_{i=1}^{P} \mathbf{A}_i \hat{\mathbf{h}}\left(n-i;l\right). \tag{6.19}$$

We take $P = 9$ to compare CE-BEM with $T = 400, Q = 9$. This scheme is denoted by "DD-KF-AR9".

3. The proposed decision-directed tracking scheme with different step sizes in the Kalman tracking, denoting by "DD-KF-BEM". For the CE-BEM, we take the first-order AR coefficient for the BEM coefficients, $\alpha = 0.992, 0.996$ and 0.998 for $m_s = 4, 2$ and 1 respectively. [We select the α values empirically via simulations to get the best performances; the corresponding theoretical values are $\alpha = 0.9992, 0.9998$ and 0.9999 in (6.9) for $m_s = 4, 2$ and 1 respectively. See the details in Section 3.4 about theoretical



(a) NCMSE vs f_dT_s



(b) BER vs f_dT_s, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

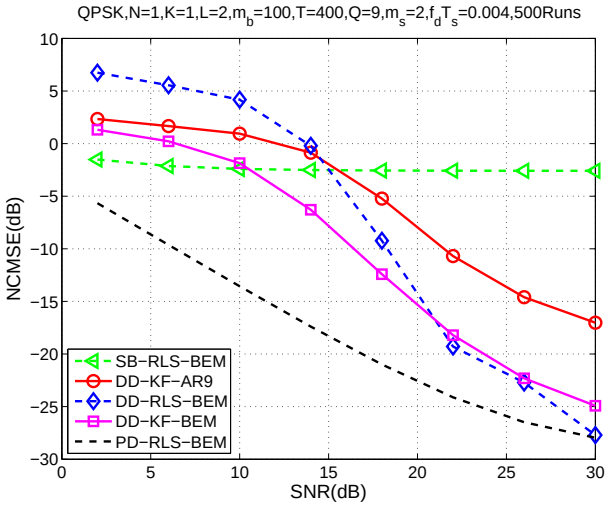
Figure 6.4: Decision-directed tracking: performance comparison for f_dT_s's, under $N = 2, \text{SNR} = 14\text{dB}, m_b = 100$.

choice for AR(1) coefficient, but replacing the subblock size m_b with the smaller step size m_s.]

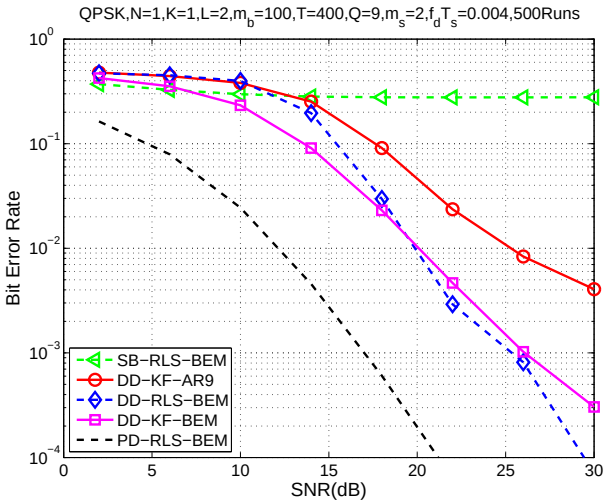
4. The proposed decision-directed tracking scheme with different step sizes in the extended EW-RLS tracking with $\beta = 1$, denoting by "DD-RLS-BEM". We take the same α value for each step size as the decision-directed Kalman tracking. We also take the forgetting factor $\lambda = 0.93, 0.96$ and 0.98 for $m_s = 4, 2$ and 1 respectively. [The theoretical choice for the forgetting factor in EW-RLS tracking is analyzed later in Section 6.4.]

5. Perfect symbol decisions are used as training for the extended EW-RLS channel tracking with step size $m_s = 2$ and the same other setups as decision-directed RLS tracking. First, the channel is estimated using exact knowledge of the information symbols. Then this estimated channel is used to detect the "unknown" information symbols. Therefore the "performance-gap" between "decision-directed tracking" and "perfect decision-directed" in figures shows the performance deterioration by decision-directed approach. [One can also use Kalman tracking with the perfect symbol decisions which has almost the same performance.] This scheme provides the baseline for decision-directed tracking, denoted by "PD-RLS-BEM".

For each of the above schemes, an MMSE-DFE described in Section 2.4.2 is employed at the receiver, using the obtained channel estimates, with $l_f = 8$, $l_b = 2$, and the equalization delay $d = 5$ symbols.



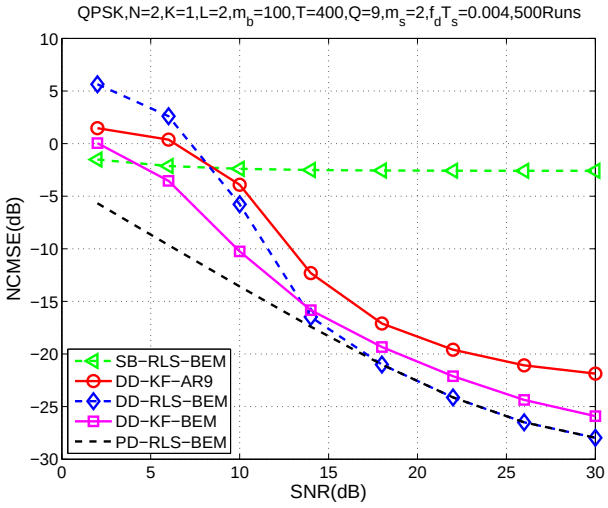
(a) NCMSE vs SNR



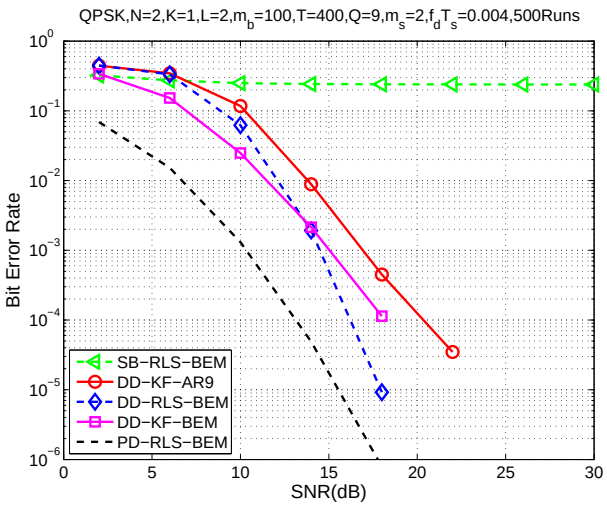
(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.5: Decision-directed tracking: performance comparison for SNR's, under $N = 1, f_dT_s = 0.004, m_b = 100$.

In Figs. 6.2 and 6.3, the performances of the schemes are compared for different SNR's, under normalized Doppler spread $f_dT_s = 0.01$. In Fig. 6.4, the performances are compared over different normalized Doppler spread f_dT_s's for fixed SNR = 14dB with two receivers ($N = 2$); we keep the AR(9) coefficients and $Q = 9$ for CE-BEM with $T = 400$, regardless of the actual f_d (we do not know it in practice). For the subblock-wise RLS tracking scheme, frequent training sessions with a small subblock size (for instance $m_b = 40$) are required in order to track the rapid channel variations; SB with a large subblock size $m_b = 100$ and hence less training overhead, does not work. Note that subblock-wise approach depends only on the training session of size m_t to estimate channel for one subblock of size m_b. On the other hand, the decision-directed tracking performs well with low training overhead for the fast-varying channel. Our decision-directed tracking has the training symbols to reduce the error propagation while subblock tracking uses them for channel estimation. Exploiting the detected information symbols as virtual training, our decision-directed tracking approach requires less training overhead to achieve satisfactory performance, so that much of the spectrum resource can be saved. Since the DFE is also a decision-directed device and thus prone to error propagation, periodic training sessions are still necessary to recover channel tracking from possible ambiguity. It is clear that since the channel variations are well captured by the BEM coefficients, our proposed BEM-based decision-directed approaches yield better performance than the symbol-wise AR-model-based decision-directed approach [6]. The performance deteriorations caused by "not-a-perfect" decision-directed are also shown



(a) NCMSE vs SNR



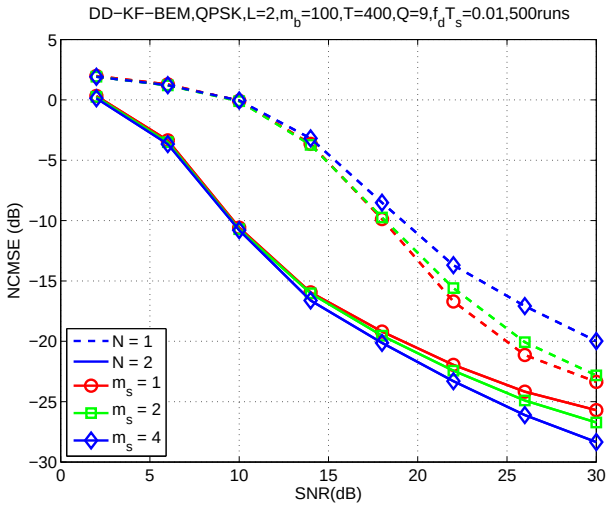
(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.6: Decision-directed tracking: performance comparison for SNR's, under $N = 2, f_dT_s = 0.004, m_b = 100$.

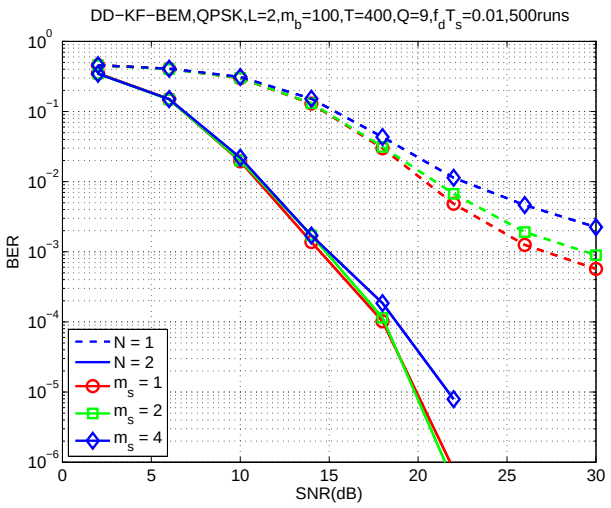
in Figs. 6.2 and 6.3, where better performance in symbol detection (with $N = 2$) leads to less deterioration. In Fig. 6.4, the NCMSE and BER for BEM-based decision-directed approaches vary only "slightly" with increasing normalized Doppler spread implying that its performance is not sensitive to the 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. In Figs. 6.5 and 6.6, the performances are compared for the actual $f_dT_s = 0.004$ (slower fading), with the same AR coefficients and CE-BEM ($T = 400, Q = 9$) for the upper bound $f_dT_s = 0.01$; similar results are shown.

Note that, in Figs. 6.2, 6.3, 6.4, 6.6 and 6.5, the proposed BEM-based decision-directed tracking schemes update the BEM coefficients every step size ($m_s = 2$ symbols), which have less computational complexity than the AR-model-based one; see the flop counts in Table 6.2. As shown in Fig. 6.7, a "finer" channel tracking can be obtained by reducing step size m_s although the computational complexity increases. Fig. 6.7 also shows that the advantage of multiple-receiver system with $N = 2$. At high SNR's with two receivers, the NCMSE for a large step size ($m_s = 4$ symbols) is better than for a small one ($m_s = 1$ symbol) since the former has more measurements with the symbol decisions close to perfect decisions.

It is hard to say which one is better between Kalman filtering and extended EW-RLS algorithm for BEM-based decision-directed tracking. For both decision-directed schemes, we selected an arbitrary value for the first-order AR coefficient α via simulations, although the theoretical choice is given in Section 3.4. In Fig. 6.8, the performances of the BEM-based



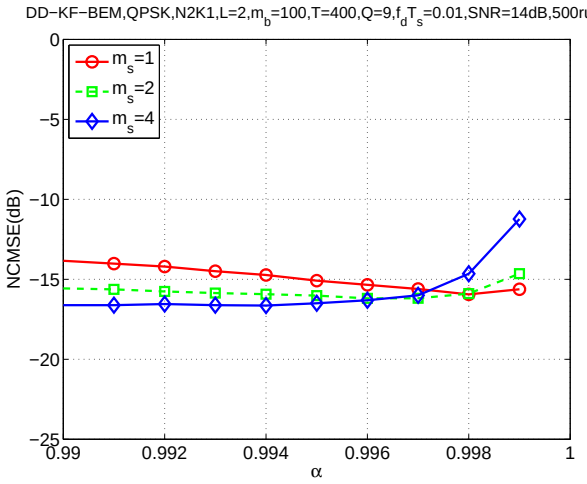
(a) NCMSE vs SNR



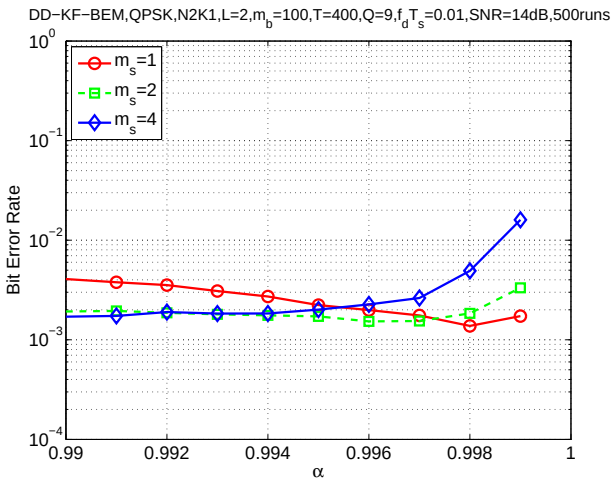
(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.7: Decision-directed Kalman tracking: performances with different step size m_s's for SNR's, under $f_dT_s = 0.01, m_b = 100$.

decision-directed Kalman tracking is shown for different values of α. It is seen that while the performance is not sensitive to the value of α over a relatively wide range of values, it does deteriorate as α approaches one. Typically α is close to one ($\alpha \approx 1$) but not quite one. Note that $\alpha = 1$ in (6.9) 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 theoretical choice of α in Section 3.4 also increases. For the extended EW-RLS algorithm, the sensitivity to α is similar but a little more due to the forgetting factor λ. In Fig. 6.9, the performances of the BEM-based decision-directed extended EW-RLS tracking are shown for different values of λ under $f_dT_s = 0.01$, SNR = 14dB. As in Fig. 6.13 where we show the theoretical MSE analysis for λ values, the minima of NCMSE and BER are similar for the different step sizes. It is also shown that the larger step size leads to more convenience to select λ for the reasonable performance. One can choose the step size m_s considering performance, complexity or convenience. In the following section (Section 6.4), we analyze the mean square error (MSE) of decision-directed EW-RLS tracking and hence obtain the baseline for the theoretical value for the forgetting factor λ in EW-RLS algorithm.



(a) NCMSE vs α



(b) BER vs α with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.8: Decision-directed Kalman tracking: performances for AR(1) coefficient α's, under $N = 2, f_d T_s = 0.01, \text{SNR} = 14\text{dB}, m_b = 100$.

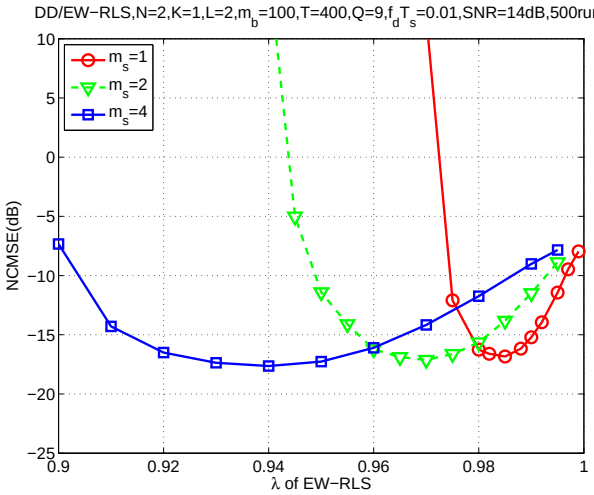
6.3 Decision-Directed MIMO Tracking using CE-BEM

Consider a doubly-selective (time- and frequency-selective) multi-input multi-output (MIMO), finite impulse response (FIR) linear channel with K inputs and N outputs. Let $\{s_k(n)\}$ denote k-th user's information sequence that is input to the time-varying channel with discrete-time response $\{\mathbf{h}_k(n;l)\}$ (channel response for the k-th user at time instance n to a unit input at time instance $n-l$). Then the symbol-rate noisy N-column channel output vector is given by $(n = 0, 1, \ldots)$

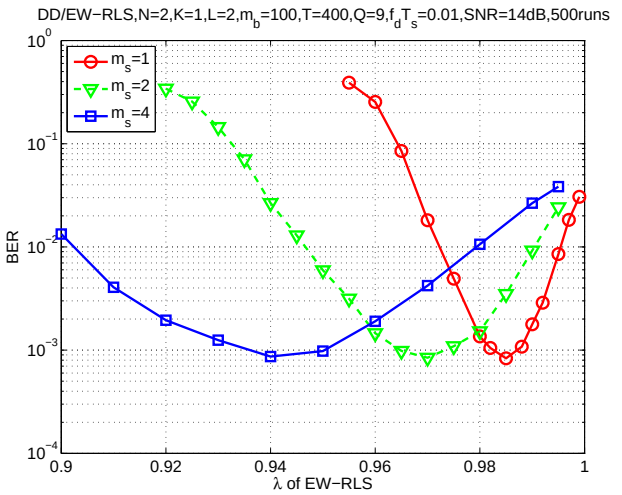
$$\mathbf{y}(n) = \sum_{k=1}^{K} \sum_{l=0}^{L} \mathbf{h}_k(n;l)\, s_k(n-l) + \mathbf{v}(n) \tag{6.20}$$

where the N-column vector $\mathbf{v}(n)$ is zero-mean, white, uncorrelated with $s_k(n)$, complex Gaussian noise, with the autocorrelation $E\{\mathbf{v}(n+\tau)\,\mathbf{v}^H(n)\} = \sigma_v^2 \mathbf{I}_N \delta(\tau)$. We assume that $\{\mathbf{h}_k(n;l)\}$ represents a wide-sense stationary uncorrelated scattering (WSSUS) channel [60], independent for different k's. We assume that $s_k(n)$'s are mutually independent and identically distributed (i.i.d.), with zero mean and variance $E\{s_k(n)\, s_k^*(n)\} = \sigma_{s_k}^2 = \sigma_s^2$ for $k = 1, 2, \cdots, K$. Define

$$\mathbf{s}(n) := \begin{bmatrix} s_1(n) & s_2(n) & \cdots & s_K(n) \end{bmatrix}^T$$
$$\mathbf{h}(n;l) := \begin{bmatrix} \mathbf{h}_1(n;l) & \mathbf{h}_2(n;l) & \cdots & \mathbf{h}_K(n;l) \end{bmatrix}.$$



(a) NCMSE vs λ



(b) BER vs λ with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.9: Decision-directed EW-RLS tracking: performances for forgetting factor λ's, under $N = 2, f_d T_s = 0.01, \text{SNR} = 14\text{dB}, m_b = 100$.

Then we may rewrite (6.20) as

$$\mathbf{y}(n) = \sum_{l=0}^{L} \mathbf{h}(n;l)\,\mathbf{s}(n-l) + \mathbf{v}(n)\,. \tag{6.21}$$

In CE-BEM [11, 14, 70], over the i-th block consisting of an observation window of T_B symbols, the channel is represented as ($n = (i-1)T_B, (i-1)T_B + 1, \cdots, iT_B - 1$ and $l = 0, 1, \cdots, L$)

$$\mathbf{h}_k(n;l) = \sum_{q=1}^{Q} \mathbf{h}_{k,q}(l) e^{j\omega_q n}, \tag{6.22}$$

where $\mathbf{h}_{k,q}(l)$ is the N-column time-invariant BEM coefficient vector for k-th user and one chooses (Λ is an integer)

$$T := \Lambda T_B, \quad \Lambda \geq 1, \tag{6.23}$$

$$Q \geq 2\lceil f_d T T_s \rceil + 1, \tag{6.24}$$

$$\omega_q := \frac{2\pi}{T}\left[q - (Q+1)/2\right], \quad q = 1, 2, \cdots, Q \tag{6.25}$$

$$L := \lfloor \tau_d / T_s \rfloor\,, \tag{6.26}$$

τ_d and f_d are respectively the delay spread and the Doppler spread, and T_s is the symbol duration. The BEM coefficients $\mathbf{h}_{k,q}(l)$'s remain invariant during this block, but are allowed

to change at the next consecutive block; the Fourier basis functions $\{e^{j\omega_q n}\}$ $(q = 1, 2, \cdots, Q)$ are common for every block. If the delay spread and the Doppler spread (or at least their upper bounds) are known, one can infer the basis functions of the CE-BEM [70]. Treating the basis functions as known, estimation of a time-varying process is reduced to estimating the invariant coefficients over a block of T_B symbols.

Now extend the decision-directed tracking using CE-BEM in Section 6.2 to MIMO systems. Consider two overlapping blocks (each consisting K-users' sequences of T_B symbols) that differ by only m_s $(1 \leq m_s \ll T_B)$ symbols. Since the past block starting at time n_0 and the present block starting at time $n_0 + m_s$ overlap significantly, the BEM coefficients representing a block of T_B symbols in CE-BEM vary only a little from the past block to the present one. We wish to estimate $\mathbf{h}_{k,q}(l)$ by updating the BEM coefficients every m_s symbols using TM training or detected symbols, rather than estimating them anew at every non-overlapping block as in [70].

Let $h_{rk,q}^{(l)}$ denote the r-th component of the column $\mathbf{h}_{k,q}^{(l)}$. Stack the BEM coefficients in (6.22) into "tall" vectors as

$$\mathbf{h}_q^{(l)} := \left[h_{11,q}^{(l)} \cdots h_{N1,q}^{(l)} \quad \cdots \quad h_{1K,q}^{(l)} \cdots h_{NK,q}^{(l)}\right]^T \tag{6.27}$$

$$\mathbf{h}^{(l)} := \left[\mathbf{h}_1^{(l)T} \quad \mathbf{h}_2^{(l)T} \quad \cdots \quad \mathbf{h}_Q^{(l)T}\right]^T \tag{6.28}$$

$$\mathbf{h} := \left[\mathbf{h}^{(0)T} \quad \mathbf{h}^{(1)T} \quad \cdots \quad \mathbf{h}^{(L)T}\right]^T \tag{6.29}$$

of size NK, NKQ and $M := NKQ(L+1)$ respectively. The coefficient vectors in (6.27), (6.28) and (6.29) of the p-th overlapping block will be denoted by $\mathbf{h}_q^{(l)}(p)$, $\mathbf{h}^{(l)}(p)$, and $\mathbf{h}(p)$ respectively. Again, we emphasize that the p-th block and the $(p+1)$-st block differ by just m_s symbols. Since a fading channel can follow well a Markov model [16], we further assume that the BEM coefficients over each overlapping block are also Markovian. A simplified formulation is given by the first-order AR model, i.e.,

$$\mathbf{h}(p) = \mathbf{A}_1 \mathbf{h}(p-1) + \mathbf{w}(p), \tag{6.30}$$

where $\mathbf{A}_1 = \alpha \mathbf{I}_M$ is the first-order AR coefficient matrix ($\alpha < 1$ but close to one), and the driving noise vector $\mathbf{w}(p)$ is zero-mean complex Gaussian with variance $\sigma_w^2 \mathbf{I}_M$ and statistically independent of $\mathbf{h}(p-1)$. Assuming the channel is wide-sense stationary (WSS) and the BEM coefficients $h_q(l)$'s are independent, we have

$$\sigma_w^2 = \sigma_h^2 (1 - |\alpha|^2)/Q \tag{6.31}$$

where $\sigma_h^2 \mathbf{I}_M := E\left\{\mathbf{h}(n;l)\,\mathbf{h}^H(n;l)\right\}$.

Define

$$\begin{aligned}\boldsymbol{\mathcal{E}}(n) &:= \left[e^{-j\omega_1 n} \quad e^{-j\omega_2 n} \quad \cdots \quad e^{-j\omega_Q n}\right]^T, \\ \mathbf{S}(n) &:= \left[\mathbf{s}^T(n) \quad \mathbf{s}^T(n-1) \quad \cdots \quad \mathbf{s}^T(n-L)\right]^T, \end{aligned} \tag{6.32}$$

where $\mathbf{s}(n)$ is K-column vector. For $pm_s \le n < (p+1)m_s$, by (6.21), (6.22), and (6.27)–(6.29), the received signal at time n can be written as

$$\mathbf{y}(n) = [\mathbf{S}(n) \otimes \mathbf{I}_N]^T [\mathbf{I}_{L+1} \otimes (\boldsymbol{\mathcal{E}}(n) \otimes \mathbf{I}_{NK})]^H \mathbf{h}(p) + \mathbf{v}(n).$$

Further defining

$$\mathbf{C}_i(p) := [\mathbf{S}(pm_s + i) \otimes \mathbf{I}_N]^T [\mathbf{I}_{L+1} \otimes (\boldsymbol{\mathcal{E}}(pm_s + i) \otimes \mathbf{I}_{NK})]^H, \tag{6.33}$$

$$\mathbf{C}(p) := \left[\mathbf{C}_0^T(p) \quad \mathbf{C}_1^T(p) \quad \cdots \quad \mathbf{C}_{m_s-1}^T(p)\right]^T, \tag{6.34}$$

we have

$$\mathbf{y}_{ms}(p) = \mathbf{C}(p)\,\mathbf{h}(p) + \mathbf{v}_{ms}(p) \tag{6.35}$$

where

$$\mathbf{y}_{ms}(p) := \left[\mathbf{y}^T(pm_s) \quad \mathbf{y}^T(pm_s + 1) \quad \cdots \quad \mathbf{y}^T((p+1)m_s - 1)\right]^T \tag{6.36}$$

and $\mathbf{v}_{ms}(p)$ is defined likewise.

6.3.1 Decision-Directed MIMO Tracking via KF or EW-RLS

The dynamical state-space MIMO system model is represented by (6.30) and (6.35), where $\mathbf{y}_{ms}(p)$ is the measurement of the state of BEM coefficients $\mathbf{h}(p)$. Kalman filtering can be applied to track the coefficient vector $\mathbf{h}(p)$ at a step size of m_s symbols. Since $\mathbf{s}(n)$ is unknown during information sessions, the receiver then switches to a decision-directed mode, in which the symbol decision is assumed to be correct ($\breve{\mathbf{s}}(n) = \mathbf{s}(n)$) and used in Kalman tracking. Treating (6.30) and (6.35) as the state and the measurement equations respectively, the Kalman tracking is applied recursively every step size. [See the details in Section 6.2.1.]

In order to avoid the periodicity of CE-BEM with a period T that has an impact on estimating "non-periodic" channels, a "finite-memory" algorithm is considered in decision-directed tracking via the extended exponentially-weighted RLS (EW-RLS) algorithm. Assuming the BEM coefficients follow the first-order AR model, a state-space model for the extended EW-RLS algorithm is given by (6.30) and (6.35).

Similarly as Kalman filtering, the extended exponentially-weighted regularized RLS (EW-RLS) algorithm [2, Section 12.B.(12.B.18)] is based on (6.30) and (6.35) to track the unknown BEM coefficient vector $\mathbf{h}\,(p)$ recursively every step size. [See the details in Section 6.2.2.]

After Kalman or RLS recursion for every p, we can generate the channel by the estimated $\hat{\mathbf{h}}\,(p)$ via the CE-BEM (6.22). Note that we also need the channel estimate $\hat{\mathbf{h}}(n;l)$ for the following $m_s + d$ symbols to make up the time gap between channel estimation and symbol detection in DFE. Since the estimated BEM coefficients are same for a large block size ($T_B >> m_s$), the channel predictions for the current m_s symbols as well as following $m_s + d$ symbols are given by

$$\hat{\mathbf{h}}(n;l) = \boldsymbol{\mathcal{E}}^H(n)\hat{\mathbf{h}}^{(l)}(p), \tag{6.37}$$

for $n = pm_s, pm_s + 1, \cdots, (p+2)m_s + d - 1$. The definition of $\hat{\mathbf{h}}^{(l)}\,(p)$ is similar to (6.28) and $\hat{\mathbf{h}}^{(l)}\,(p)$ is based on observations up to time $n = (p+1)m_s - 1$.

Using the channel estimates $\left\{\hat{\mathbf{h}}(n;l)\right\}$ in (6.37), the symbol decisions are made by an FIR MMSE-DFE in Section 2.4.2 [40] for the time $n = pm_s, pm_s + 1, \cdots, (p+2)m_s - 1$. Note that the detected symbols for the time $n = (p+1)m_s, (p+1)m_s + 1, \cdots, (p+2)m_s - 1$ are used for the following channel tracking, whereas the "current" decisions for the time $n = pm_s, pm_s + 1, \cdots, (p+1)m_s - 1$ are updated for improved performance.

Computational Complexity

We compare computational complexity using the floating point operation (flop) count for the decision-directed MIMO channel tracking schemes. Here we consider Kalman filtering as the common basis for all the simulation schemes to fairly compare the decision-directed approach with subblock-wise one. We consider three different channel estimation schemes: "SB-KF(BEMT)" denoting BEM-based subblock-wise Kalman filtering with a period T in Chapter 3, "DD-ARP" denoting AR(P) model-based symbol-by-symbol decision-directed Kalman filtering [6], "DD-KF(BEMT)" denoting BEM-based decision-directed Kalman filtering with a period T. All the above schemes have the same subblock ($m_b = 100$) and hence same training overhead. We take $T = 400$ and $Q = 9$ for CE-BEM. [The complexity

of EW-RLS algorithm is almost the same as Kalman filtering. See the detailed flop counts of Kalman filtering in Section 3.3.1 and EW-RLS algorithm in Section 4.3.1.]

The number of flops for one cycle of of subblock-wise Kalman filtering, decision-directed Kalman filtering and DFE turn out to be

$$\begin{aligned}
f_c(\text{SB-KF}) =& M_s^3 + M_s^2(3L_p+3) + M_s(2L_p^2+5L_p+1) + L_p^3/3 + L_p^2 \\
f_c(\text{DD-KF}) =& M^3 + M^2(3Nm_s+2) + M[2(Nm_s)^2+3Nm_s+1] + (Nm_s)^3/3 + (Nm_s)^2 \\
f_c(\text{DFE}) =& 2(Nl_f)^3 + (Nl_f)^2(\bar{l}_f+\bar{l}_b+K+1) + (Nl_f)(\bar{l}_f\bar{l}_b+\bar{l}_b^2+K\bar{l}_b+K) \\
&+ 2(\bar{l}_b^3+K\bar{l}_b^2+K^2\bar{l}_b) + K^3/3 + K^2 l_b
\end{aligned}$$

where $M_s = Q(L+1)$ for SB-KF(BEMT), $M = NKP(L+1)$ for DD-KF(ARP) or $NKQ(L+1)$ for DD-KF(BEMT), and $L_p = L+1, \bar{l}_f = K(l_f+L), \bar{l}_b = K(l_b+1)$. The overall flop counts over one block of T_B symbols for the subblock-wise and decision-directed Kalman tracking schemes are then given by

$$\begin{aligned}
f_c(\text{SB-KF(BEM}T\text{)}) &= N(T_B/m_b)f_c(\text{SB-KF}) + [T_B - m_t(T_B/m_b)]f_c(\text{DFE}) \\
f_c(\text{DD-KF(AR}P\text{)}) &= T_B f_c(\text{DD-KF}) + 2[T_B - m_t(T_B/m_b)]f_c(\text{DFE}), m_s = 1 \\
f_c(\text{DD-KF(BEM}T\text{)}) &= (T_B/m_s)f_c(\text{DD-KF}) + 2[T_B - m_t(T_B/m_b)]f_c(\text{DFE}).
\end{aligned}$$

The comparative flop counts for each scheme with three receivers and two users($N = 3, K = 2$) are compared in Table 6.3 over one block ($T_B = 200$).

Table 6.3: Subblock-wise and decision-directed MIMO channel tracking with DFE: comparative flop count over one block of T_B symbols.

tracking schemes	comparative flops
SB-KF(BEM400)	1
DD-KF(BEM400), $m_s = 4$	30.18
DD-KF(BEM400), $m_s = 2$	52.99
DD-KF(BEM400), $m_s = 1$	98.81
DD-KF(AR9)	98.81

6.3.2 Simulation Examples

A random time- and frequency-selective Rayleigh fading MIMO channel is considered. We assume $\mathbf{h}(n;l)$ are zero-mean, complex Gaussian, and spatially white with autocorrelation $E\left\{\mathbf{h}(n;l)\,\mathbf{h}^H(n;l)\right\} = \sigma_h^2\mathbf{I}_N$ for each l. We take $L = 2$ (3 taps) in (6.21), and $\sigma_h^2 = 1/(L+1)$. For different l's, $\mathbf{h}(n;l)$'s are mutually independent and satisfy Jakes' model. To this end, we simulate each single tap following [75] (with a correction in the appendix of [63]).

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 $f_d = 400\,$Hz, or the normalized Doppler spread $f_dT_s = 0.01$ (corresponding to a maximum mobile velocity 216km / h). 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.

We evaluate the performances of various schemes by considering their normalized channel mean square error (NCMSE) and their bit error rates (BER). The NCMSE is defined as

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

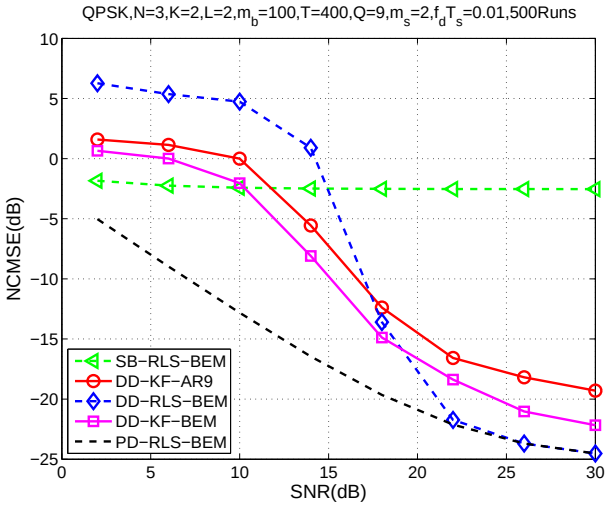
where $\mathbf{h}^{(i)}(n;l)$ is the true channel and $\hat{\mathbf{h}}^{(i)}(n;l)$ is the estimated channel at the i-th Monte Carlo run, among total M_r runs. In each run, a training mode of 200 symbols modulated by binary phase-shift keying (BPSK) is followed by a decision-directed mode of quadrature phase-shift keying (QPSK) 4000 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 [70] is adopted, where each subblock of equal length m_b symbols consists 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 for each user contains an impulse guarded by zeros (silent periods), which for the $k-$th user has the structure ($\gamma > 0$)

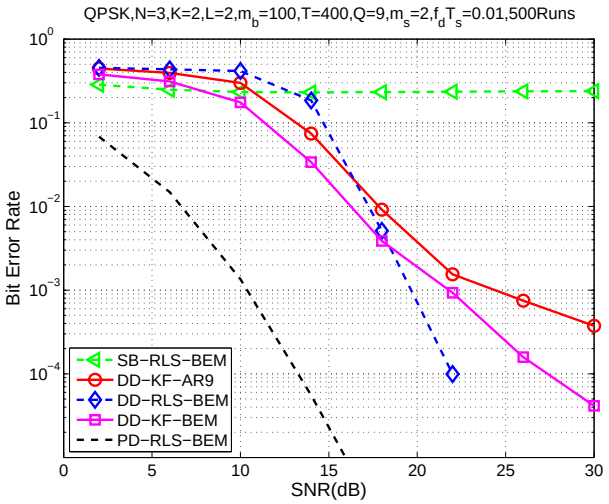
$$\mathbf{c}_{k,p} := \left[\mathbf{0}_{1\times((k-1)(L+1)+L)} \quad \gamma \quad \mathbf{0}_{1\times((K-k)(L+1)+L)}\right] \tag{6.38}$$

where $k = 1, \cdots, K$. Therefore, $m_t = K(L+1) + L$, which have to be devoted for training and the remaining are available for information symbols. Note that $\mathbf{c}_{k,p}$ is the $k-$th row of the matrix $\mathbf{c}_p$. We assume that each information symbol has unit power, while at every training session given by (6.38), we set $\gamma = \sqrt{K(L+1)+L}$ so that the average power per symbol at training sessions is equal to that of information sessions.

We compared the following schemes:



(a) NCMSE vs SNR



(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.10: Decision-directed MIMO tracking: performance comparison for SNR's, under $N = 3, K = 2, f_d T_s = 0.01, m_b = 100$.

1. The subblock-wise EW-RLS algorithm with $\beta = 1$, which is one of the best subblock-wise approaches in Chapter 4. [One can pick a different subblock-wise scheme using SW-RLS algorithm or Kalman filter that has similar (or slightly worse) performances.] In the simulations, we take the forgetting factor $\lambda = 0.5$ for $m_b = 100$ in EW-RLS algorithm. This scheme is denoted by "SB-RLS-BEM".

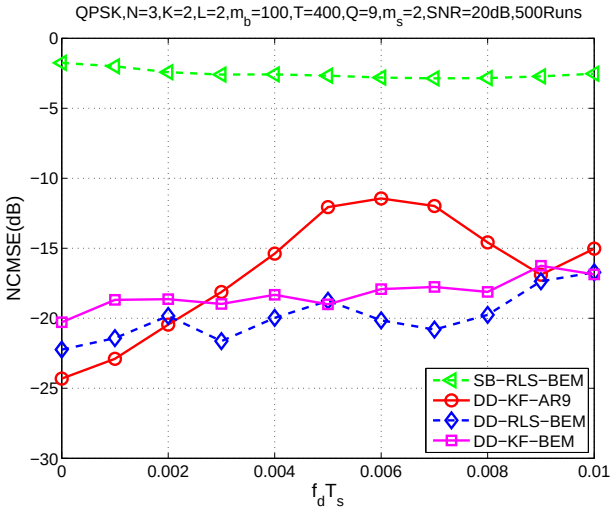
2. The decision-directed channel tracking scheme in [6] using the P-th order AR model, i.e., the time-varying channel follows (assuming each independent channel tap has same Doppler spread)

$$\mathbf{h}(n;l) = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{h}(n-i;l) + \mathbf{w}(n), \tag{6.39}$$

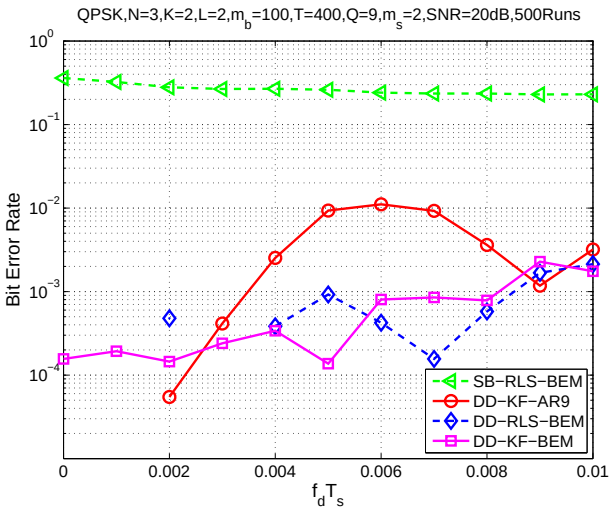
where $\mathbf{A}_i = \alpha_i \mathbf{I}$ is the AR coefficients matrix and driving noise $\mathbf{w}(n)$ is zero-mean complex Gaussian with autocorrelation, $E\left\{\mathbf{w}(n)\mathbf{w}^H(n)\right\} = \sigma_w^2 \mathbf{I}_N$. Using Yule-Walker equations, we take the AR coefficients and noise variance for $f_d T_s = 0.01$. The channel prediction stage is conducted by using (6.39) omitting the driving noise, i.e.,

$$\hat{\mathbf{h}}(n;l) = \sum_{i=1}^{P} \mathbf{A}_i \hat{\mathbf{h}}(n-i;l). \tag{6.40}$$

We take $P = 9$ to compare CE-BEM with $T = 400, Q = 9$. This scheme is denoted by "DD-KF-AR9".



(a) NCMSE vs f_dT_s



(b) BER vs f_dT_s, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

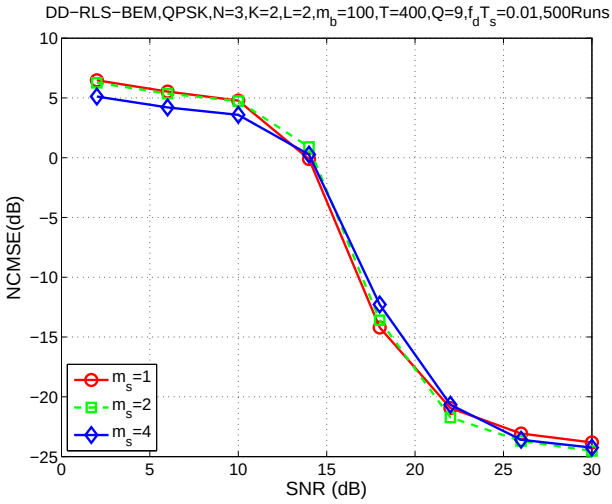
Figure 6.11: Decision-directed MIMO tracking: performance comparison for f_dT_s's, under $N = 3, K = 2, \text{SNR} = 20\text{dB}, m_b = 100$.

3. The proposed decision-directed tracking scheme with different step sizes in the Kalman tracking, denoting by "DD-KF-BEM". For the CE-BEM, we take the first-order AR coefficient for the BEM coefficients empirically via simulations, $\alpha = 0.990, 0.994$ and 0.997 for $m_s = 4, 2$ and 1 respectively.

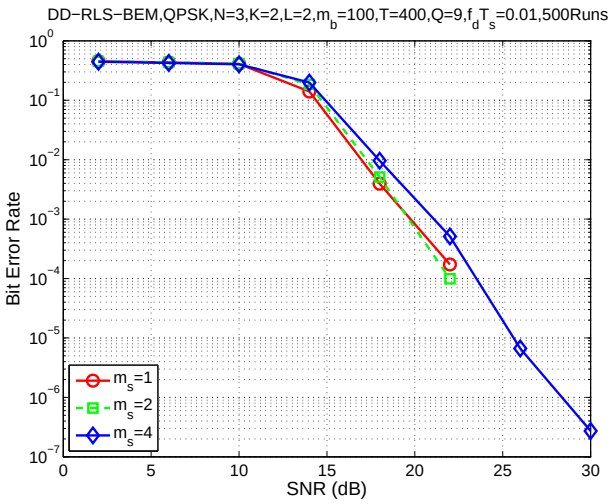
4. The proposed decision-directed tracking scheme with different step sizes in the extended EW-RLS tracking with $\beta = 1$, denoting by "DD-RLS-BEM". We take the same α value for each step size as the decision-directed Kalman tracking. We also take the forgetting factor $\lambda = 0.93, 0.96$ and 0.98 for $m_s = 4, 2$ and 1 respectively.

5. Perfect symbol decisions are used as training for the extended EW-RLS channel tracking with step size $m_s = 2$ and the same other setups as decision-directed RLS tracking. First, the channel is estimated using exact knowledge of the information symbols. Then this estimted channel is used to detect the "unknown" information symbols. Therefore the "performance-gap" between "decision-directed tracking" and "perfect decision-directed" in figures shows the performance deterioration by decision-directed approach. [One can also use Kalman tracking with the perfect symbol decisions which has almost the same performance.] This scheme provides the baseline for decision-directed tracking, denoted by "PD-RLS-BEM".

In the simulations, we consider a three-receiver and two-user scenario, i.e., $N = 3, K = 2$ with the same transmitted power. We have $m_t = 8$ with $\gamma = \sqrt{8}$ for every user following the



(a) NCMSE vs SNR



(b) BER vs SNR, with DFE of $l_f = 8, l_b = 2$ and equalization delay $d = 5$

Figure 6.12: Decision-directed EW-RLS MIMO tracking: performances with different step size m_s's for SNR's, under $N = 3, K = 2, f_dT_s = 0.01, m_b = 100$.

TM training scheme. For each of the above schemes, an MMSE-DFE described in Section 2.4.2 is employed at the receiver, using the obtained channel estimates, with $l_f = 8$, $l_b = 2$, and the equalization delay $d = 5$ symbols.

In Fig. 6.10, the performances of the schemes are compared for different SNR's, under normalized Doppler spread $f_dT_s = 0.01$. In Fig. 6.11, the performances are compared over different normalized Doppler spread f_dT_s's for fixed SNR = 20dB; we keep the AR(9) coefficients and $Q = 9$ for CE-BEM with $T = 400$ for the Doppler spread $f_d = 400$Hz. It is clear the subblock-wise RLS tracking scheme does not work with a large subblock size $m_b = 100$ at all since it depends only on the training to estimate channels. Note that the decision-directed schemes take the symbol-decisions as virtual training and hence perform well with less training overhead. It is shown the CE-BEM-based decision-directed approaches yield better performance than the symbol-wise AR-model-based decision-directed approach [6]. Especially, the extended EW-RLS decision-directed tracking, a finite memory algorithm with a forgetting factor λ, performs closely to the perfect decision-directed scheme with high SNR's. In Fig. 6.11, the performances for DD-RLS-BEM and DD-KF-BEM are not so sensitive to the Doppler spread of the channel as AR-model-based scheme, which means an upper bound, for instance $f_d = 400$Hz, is sufficient in practice with unknown exact Doppler spreads.

The proposed BEM-based decision-directed tracking schemes update the BEM coefficients every step size (m_s symbols) while the AR-model-based one updates the channel estimates every symbol. Note that we take $m_s = 2$ for the BEM-based schemes ($T = 400, Q = 9$) in Fig. 6.10 and 6.11, where they have better performances with less computational complexity than the AR(9)-based one. [See the flop counts in Table 6.3.] As shown in Fig. 6.12, a "finer" channel tracking can be obtained by reducing step size m_s although the computational complexity increases.

6.4 Performance Analysis of Decision-Directed EW-RLS Tracking using CE-BEM

The cost function for the EW-RLS algorithm can be rewritten for "large" $p (p \to \infty)$ as

$$\mathcal{C}_{EW} = \sum_{i=0}^{\infty} \lambda^i \left\| \mathbf{y}_{m_s}(p-i) - \mathbf{C}(p-i)\mathbf{h} \right\|^2 . \tag{6.41}$$

Let N_s denote the multiples of the step size m_s on which we would like to base the estimate of BEM coefficient $\mathbf{h}$. It is clear that $E\left\{\mathbf{C}^H(i)\mathbf{C}(i)\right\}$ is periodic in i with period T/m_s as CE-BEM is periodic with period T. In practice, we would like to have the memory length (in symbols) to be less than the model period (recall that the channel is by no means periodic) so that there are no deleterious effects due to the CE-BEM model for all time, i.e., $m_s N_s \leq T$ in order to avoid this periodicity. Let us pick $N_s = T/m_s$. What other restriction should we impose on N_s? In the following we assume that the detected symbols resulting from MMSE-DFE and used for EW-RLS tracking, are correct (no detection error).

A least-square solution for $\mathbf{h}$ to minimize $\mathcal{C}_{EW}$ is given by [50, p. 796]

$$\hat{\mathbf{h}} = \mathbf{A}^{-1}\mathbf{B} \tag{6.42}$$

where

$$\mathbf{A} := \frac{1}{N_s} \sum_{i=0}^{\infty} \lambda^i \mathbf{C}^H(p-i)\mathbf{C}(p-i), \tag{6.43}$$

$$\mathbf{B} := \frac{1}{N_s} \sum_{i=0}^{\infty} \lambda^i \mathbf{C}^H(p-i)\mathbf{y}_{m_s}(p-i). \tag{6.44}$$

In order to analyze the behavior of $\mathbf{h}$ we need a model for $\mathbf{y}_{m_s}(i)$ for every step, $i = p, p-1, \cdots$. To this end, for the following analysis presented in this section, we assume the

following "simplified" model:

$$\mathbf{y}_{m_s}(p-i) = \mathbf{C}(p-i)\mathbf{h}^{(tr)}(p-i) + \mathbf{v}_{m_s}(p-i). \tag{6.45}$$

where $\mathbf{h}^{(tr)}(p-i)$ is the "true" BEM coefficient vector satisfying $(m = 1, 2, \cdots)$

$$\mathbf{h}^{(tr)}(p-i) := \mathbf{h}_m^{(tr)} \text{ for } (m-1)N_s \leq p-i < mN_s. \tag{6.46}$$

That is, there exist some true BEM parameters that are "fixed" over one BEM period of T symbols (therefore, $N_s = T/m_s$ steps), and are allowed to change over non-overlapping periods. In this set-up, we estimate the most recent true BEM coefficient vector $\mathbf{h}_1^{(tr)}$ via $\hat{\mathbf{h}}$ without it being unduly influenced by $\mathbf{h}_m^{(tr)}, m \geq 2$. Assuming the estimate $\hat{\mathbf{h}}$ asymptotically Gaussian distributed, we are trying to evaluate mean-square-error (MSE) of channel estimation when the true channel follows CE-BEM.

The MSE of the channel estimate using CE-BEM is defined as

$$\text{MSE}\left(\hat{\mathbf{h}}\right) := \underbrace{E\left\{E\left\{\left\|\hat{\mathbf{h}} - \mathbf{h}_1^{(tr)}\right\|^2 \mid \mathbf{h}^{(tr)}\right\}\right\}}_{=:\text{MSE}_1\left(\hat{\mathbf{h}}\right)} + \underbrace{\frac{1}{T}\sum_{n=1}^{T}\sum_{l=0}^{L} E\left\{\left\|\mathbf{e}_{\text{BEM}}(n;l)\right\|^2\right\}}_{=:\text{MSE}_2\left(\hat{\mathbf{h}}\right)}, \tag{6.47}$$

where $\text{MSE}_1\left(\hat{\mathbf{h}}\right)$ comes from the channel estimation and $\text{MSE}_2\left(\hat{\mathbf{h}}\right)$ is an mean square modeling error that is intrinsic to the BEM representation. Note that $\text{MSE}_2\left(\hat{\mathbf{h}}\right)$ has nothing to do with channel estimation.

Following [51, Section 4.3], consider the BEM representation (6.2) as an approximation where only Q (out of total T) basis functions are used to describe the channel. Then $\text{MSE}_2\left(\hat{\mathbf{h}}\right)$ is given by

$$\begin{aligned}\text{MSE}_2\left(\hat{\mathbf{h}}\right) &= \frac{1}{T}\sum_{q=Q+1}^{T}\sum_{l=0}^{L} E\left\{\left\|\mathbf{h}_q(l)\right\|^2\right\} \\ &= \frac{1}{T}\sum_{q=1}^{T}\sum_{l=0}^{L}\sum_{n_1=0}^{T-1}\sum_{n_2=0}^{T-1} R_h(n_1,n_2;l)e^{-j\frac{2\pi[q-(T+1)/2]}{T}(n_1-n_2)}\end{aligned} \tag{6.48}$$

where $R_h(n_1,n_2;l) := E\left\{\mathbf{h}^H(n_1;l)\mathbf{h}(n_2;l)\right\}$. For the modified Jakes' model,

$$R_h(n_1,n_2;l) = N\sigma_h^2 J_0\left(2\pi f_d T_s(n_1-n_2)\right)$$

where $J_0(\cdot)$ denotes the zero-th order Bessel function of the first kind. After some manipulations, we obtain

$$\begin{aligned}\mathrm{MSE}_2\left(\hat{\mathbf{h}}\right) &= \frac{L+1}{T^2}\sum_{n_1=0}^{T-1}\sum_{n_2=0}^{T-1} N\sigma_h^2 J_0\left(2\pi f_d T_s(n_1-n_2)\right)\left[\sum_{q=\frac{Q+1}{2}}^{T-\frac{Q+1}{2}} e^{j\frac{2\pi q(n_1-n_2)}{T}}\right] \\ &= \frac{L+1}{T}N\sigma_h^2\left[(T-Q)-2\sum_{\tau=1}^{T-1}\left\{1-\frac{\tau}{T}J_0\left(2\pi f_d T_s\tau\right)\frac{\sin\frac{\pi\tau Q}{T}}{\sin\frac{\pi\tau}{T}}\right\}\right]. \end{aligned} \tag{6.49}$$

For the mean square error of channel estimate MSE_1, consider nonzero bias and the channel estimation error can be written as

$$\hat{\mathbf{h}}-\mathbf{h}_1^{(tr)} = \hat{\mathbf{h}}-\bar{\mathbf{h}}+\underbrace{\bar{\mathbf{h}}-\mathbf{h}_1^{(tr)}}_{=\text{bias}},$$

where $\bar{\mathbf{h}} := E\left\{\hat{\mathbf{h}}\right\} \neq \mathbf{h}^{(tr)}$. Then MSE_1 is given by

$$\mathrm{MSE}_1\left(\hat{\mathbf{h}}\right) = \underbrace{E\left\{\left\|\hat{\mathbf{h}}-\bar{\mathbf{h}}\right\|^2\right\}}_{=tr\left\{\mathrm{cov}\left\{\hat{\mathbf{h}}\right\}\right\}} + E\left\{E\left\{\left\|\bar{\mathbf{h}}-\mathbf{h}_1^{(tr)}\right\|^2 \mid \mathbf{h}^{(tr)}\right\}\right\} \tag{6.50}$$

The difference between the estimate $\hat{\mathbf{h}}$ and its expected $\bar{\mathbf{h}}$ can be determined by [61, Chapter 8, p.261]

$$\hat{\mathbf{h}}-\bar{\mathbf{h}} = \mathbf{A}^{-1}\tilde{\mathbf{B}} \tag{6.51}$$

where

$$\tilde{\mathbf{B}} := \frac{1}{N_s}\sum_{i=0}^{\infty}\lambda^i\mathbf{C}^H(p-i)\mathbf{v}_{m_s}(p-i). \tag{6.52}$$

Then the covariance of channel estimate, $\mathrm{cov}\left\{\hat{\mathbf{h}}\right\}$ in (6.50) is given by

$$\mathrm{cov}\left\{\hat{\mathbf{h}}\right\} = E\left\{\left(\hat{\mathbf{h}}-\bar{\mathbf{h}}\right)\left(\hat{\mathbf{h}}-\bar{\mathbf{h}}\right)^H\right\} = E\left\{\mathbf{A}^{-1}\tilde{\mathbf{B}}\tilde{\mathbf{B}}^H(\mathbf{A}^{-1})^H\right\}. \tag{6.53}$$

We can rewrite (6.43) as

$$\mathbf{A} = \sum_{m=1}^{\infty}\lambda^{(m-1)N_s}\mathbf{A}_m, \tag{6.54}$$

where

$$\mathbf{A}_m = \frac{1}{N_s}\sum_{i=0}^{N_s-1}\lambda^i\mathbf{C}^H(p-i-(m-1)N_s)\mathbf{C}(p-i-(m-1)N_s).$$

Now we establish that

$$\lim_{N_s\to\infty} E\left\{\left\|\mathbf{A}_m - E\left\{\mathbf{A}_m\right\}\right\|^2\right\} = 0. \tag{6.55}$$

We have

$$E\left\{\left\|\mathbf{A}_m - E\left\{\mathbf{A}_m\right\}\right\|^2\right\} \le \frac{1}{N_s^2}\sum_{i=0}^{N_s-1}\sum_{l=0}^{N_s-1}\lambda^{i+l}\left\|E\left\{\mathbf{Y}_i^H\mathbf{Y}_l\right\}\right\|, \tag{6.56}$$

where

$$\begin{aligned}\mathbf{Y}_i &:= \mathbf{X}_i - E\left\{\mathbf{X}_i\right\},\\ \mathbf{X}_i &:= \mathbf{C}^H(p-i-(m-1)N_s)\mathbf{C}(p-i-(m-1)N_s).\end{aligned}$$

Since the information sequence $\{s(n)\}$ is zero-mean white, it follows that $E\left\{\mathbf{Y}_i^H\mathbf{Y}_l\right\} = 0$ for $|i-l| > l_0 := 1 + \left\lceil\frac{L-1}{m_s}\right\rceil$. Setting $i-l=j$ and using the facts that $\left\|E\left\{\mathbf{Y}_i^H\mathbf{Y}_l\right\}\right\| \le b_0$ for $|i-l|\le l_0$ and some $0<b_0<\infty$, and $\lambda<1$, we can simplify (6.56) as

$$\begin{aligned}E\left\{\left\|\mathbf{A}_m - E\left\{\mathbf{A}_m\right\}\right\|^2\right\} &\le \frac{1}{N_s^2}\sum_{j=-N_s+1}^{N_s-1}\sum_{l=0}^{N_s-1}\lambda^{j+2l}\left\|E\left\{\mathbf{Y}_{j+l}^H\mathbf{Y}_l\right\}\right\|\\ &\le \frac{2N_s-1}{N_s^2}(2l_0+1)b_0 \to 0 \text{ as } N_s\to\infty.\end{aligned} \tag{6.57}$$

Since $\lambda<1$, in a similar fashion we can show that

$$\lim_{N_s\to\infty} E\left\{\left\|\mathbf{A} - E\left\{\mathbf{A}\right\}\right\|^2\right\} = 0. \tag{6.58}$$

Defining $\mathbf{z}(p)$ for the signal part and $\mathbf{\Psi}(p)$ for the BEM basis function part in $\mathbf{C}(p)$ as

$$\mathbf{z}(p) := \begin{bmatrix}\mathbf{s}^T(pm_s) & & & \\ & \mathbf{s}^T(pm_s+1) & & \\ & & \ddots & \\ & & & \mathbf{s}^T((p+1)m_s-1)\end{bmatrix}, \tag{6.59}$$

$$\mathbf{\Psi}(p) := \mathbf{I}_{L+1}\otimes\begin{bmatrix}\boldsymbol{\mathcal{E}}(pm_s) & \boldsymbol{\mathcal{E}}(pm_s+1) & \cdots & \boldsymbol{\mathcal{E}}((p+1)m_s-1)\end{bmatrix}^H, \tag{6.60}$$

we have $\mathbf{C}(p) = \mathbf{z}(p)\mathbf{\Psi}(p)$. Note that we assumed the information sequence $\{s(n)\}$ is mutually independent and identically distributed (i.i.d.) with zero mean and variance $E\{s(n)s^*(n)\} = \sigma_s^2$. An important observation is that since $E\left\{\mathbf{C}^H(p)\mathbf{C}(p)\right\}$ is periodic

with period $N_s = T/ms$, $E\{\mathbf{A}_m\}$ is not a function of m. Therefore, we have

$$\begin{aligned}
E\{\mathbf{A}\} &= \sum_{m=0}^{\infty} \lambda^{(m-1)N_s} E\{\mathbf{A}_m\} = \frac{1}{1-\lambda_s^N} E\{\mathbf{A}_m\} \\
&= \frac{1}{1-\lambda_s^N} E\left\{ \frac{1}{N_s} \sum_{i=0}^{N_s-1} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{z}^H(p-i)\mathbf{z}(p-i)\mathbf{\Psi}(p-i) \right\} \\
&= \frac{1}{1-\lambda_s^N} \frac{1}{N_s} \sum_{i=0}^{N_s-1} \lambda^i \mathbf{\Psi}^H(p-i) E\left\{\mathbf{z}^H(p-i)\mathbf{z}(p-i)\right\} \mathbf{\Psi}(p-i) \\
&= \frac{\sigma_s^2}{1-\lambda_s^N} \underbrace{\frac{1}{N_s} \sum_{i=0}^{N_s-1} \lambda^i \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)}_{=:\mathbf{A}_0}.
\end{aligned} \tag{6.61}$$

and (i.p. stands for in probability)

$$\lim_{N_s\to\infty} \mathbf{A} \overset{i.p.}{=} E\{\mathbf{A}\} = \frac{\sigma_s^2}{1-\lambda_s^N}\mathbf{A}_0. \tag{6.62}$$

Based on (6.62), we can rewrite (6.53) as

$$\operatorname{cov}\left\{\hat{\mathbf{h}}\right\} = E\{\mathbf{A}\}^{-1} \operatorname{cov}\left\{\tilde{\mathbf{B}}\right\} E\left\{\mathbf{A}^H\right\}^{-1}. \tag{6.63}$$

Assuming $v(n)$ is zero-mean, white noise, uncorrelated with $s(n)$, with $E\{v(n+\tau)v^*(n)\} = \sigma_v^2\delta(\tau)$, the covariance of $\tilde{\mathbf{B}}$ is given as (using (6.52) and $\mathbf{C}(p-i) = \mathbf{z}(p-i)\,\mathbf{\Psi}(p-i)$)

$$\begin{aligned}
&\operatorname{cov}\left\{\tilde{\mathbf{B}}\right\} = E\left\{\tilde{\mathbf{B}}\tilde{\mathbf{B}}^H\right\} \\
&= \frac{1}{N_s^2} E\left\{ \left[\sum_{i_1=0}^{\infty} \lambda^{i_1} \mathbf{C}^H(p-i_1)\mathbf{v}_{m_s}(p-i_1)\right] \left[\sum_{i_2=0}^{\infty} \lambda^{i_2} \mathbf{C}^H(p-i_2)\mathbf{v}_{m_s}(p-i_2)\right]^H \right\} \\
&= \frac{1}{N_s^2} \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty} \lambda^{i_1+i_2} E\left\{ \mathbf{C}^H(p-i_1)\left[\mathbf{v}_{m_s}(p-i_1)\mathbf{v}_{m_s}^H(p-i_2)\right]\mathbf{C}(p-i_2)\right\} \\
&= \frac{\sigma_v^2}{N_s^2} \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty} \lambda^{i_1+i_2} \mathbf{\Psi}^H(p-i_1) E\left\{\mathbf{z}^H(p-i_1)\mathbf{z}(p-i_2)\right\} \mathbf{\Psi}(p-i_2)\delta(i_1-i_2) \\
&= \frac{\sigma_v^2}{N_s^2} \sum_{i=0}^{\infty} \lambda^{2i} \mathbf{\Psi}^H(p-i) E\left\{\mathbf{z}^H(p-i)\mathbf{z}(p-i)\right\} \mathbf{\Psi}(p-i) \\
&= \frac{\sigma_v^2\sigma_s^2}{N_s^2} \sum_{i=0}^{\infty} \lambda^{2i} \mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)
\end{aligned} \tag{6.64}$$

Using the periodicity, $\mathbf{\Psi}(p-i-(m-1)N_s) = \mathbf{\Psi}(p-i)$ for integer m's, we can rewrite (6.64) as

$$\begin{aligned}\operatorname{cov}\left\{\tilde{\mathbf{B}}\right\} &= \frac{\sigma_v^2\sigma_s^2}{N_s^2}\sum_{m=1}^{\infty}\lambda^{(m-1)2N_s}\left[\sum_{i=0}^{N_s-1}\lambda^{2i}\mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)\right] \\ &= \frac{\sigma_v^2\sigma_s^2}{1-\lambda^{2N_s}}\underbrace{\frac{1}{N_s^2}\sum_{i=0}^{N_s-1}\lambda^{2i}\mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)}_{=:\mathbf{B}_0}\end{aligned} \tag{6.65}$$

Hence, using (6.62), (6.63) and (6.65), we have

$$\operatorname{cov}\left\{\hat{\mathbf{h}}\right\} = \frac{\sigma_v^2}{\sigma_s^2}\left(\frac{1-\lambda^{N_s}}{1+\lambda^{N_s}}\right)\mathbf{A}_0^{-1}\mathbf{B}_0\mathbf{A}_0^{-1} = \frac{1}{\mathrm{SNR}}\left(\frac{1-\lambda^{N_s}}{1+\lambda^{N_s}}\right)\mathbf{A}_0^{-1}\mathbf{B}_0\mathbf{A}_0^{-1}. \tag{6.66}$$

Considering the symbol decision errors in the decision-directed channel estimation, define symbol error as

$$|s_e(n)| := |s(n)-\hat{s}(n)| = \begin{cases} 0 & \text{with probability} \quad 1-P_{se} \\ d & \text{with probability} \quad P_{se} \end{cases} \tag{6.67}$$

where $d=1$ is the distance between two nearby symbols in the QPSK constellation diagram and P_{se} is the symbol error rate. Then the variance of the symbol error is given by

$$E\left\{s_e(n)s_e^*(n)\right\} = E\left\{|s_e(n)|^2\right\} = d^2\sigma_s^2P_{se} = \sigma_s^2P_{se}. \tag{6.68}$$

Using the symbol decision, one may rewrite the received signal as

$$\begin{aligned}\mathbf{y}(n) &= \sum_{l=0}^{L}\mathbf{h}(n;l)\,s(n-l) + \mathbf{v}(n) \\ &= \sum_{l=0}^{L}\mathbf{h}(n;l)\left[\hat{s}(n-l) + \left\{s(n-l)-\hat{s}(n-l)\right\}\right] + \mathbf{v}(n) \\ &= \sum_{l=0}^{L}\mathbf{h}(n;l)\,\hat{s}(n-l) + \underbrace{\sum_{l=0}^{L}\mathbf{h}(n;l)\,s_e(n-l) + \mathbf{v}(n)}_{=:\mathbf{v}_e(n)}\end{aligned} \tag{6.69}$$

where $\mathbf{v}_e(n)$ is the "effective" noise including the symbol decision errors. In order to compensate the symbol decision errors, we take the effective noise variance σ_{ve}^2 as

$$E\left\{\mathbf{v}_e(n)\mathbf{v}_e^H(n)\right\} = \sigma_v^2\mathbf{I}_N + \sum_{l_1=0}^{L}\sum_{l_2=0}^{L} E\left\{\mathbf{h}(n;l_1)\mathbf{h}^H(n;l_2)\right\} E\left\{s_e(n-l_1)s_e^*(n-l_2)\right\}$$
$$= \sigma_v^2\mathbf{I}_N + \sigma_s^2 P_{se}\sum_{l=0}^{L} E\left\{\mathbf{h}(n;l)\mathbf{h}^H(n;l)\right\} \tag{6.70}$$

For zero-mean, complex Gaussian and spatially white channel with variance $\sigma_h^2 = 1/(L+1)$, it can be simplified as

$$\sigma_{ve}^2 = \sigma_v^2 + \sigma_s^2 P_{se}. \tag{6.71}$$

The probability of symbol error may be approximated as $P_{se} \approx 2P_{be}$ for high SNR's (as is necessary for practical QPSK systems), where P_{be} is a specific bit error rate. Using this approximation, the effective noise variance is given by

$$\sigma_{ve}^2 \approx \sigma_v^2 + \sigma_s^2 \cdot 2P_{be} \tag{6.72}$$

and we can replace σ_v^2 in (6.66) with σ_{ve}^2 in (6.72). For example, the effective variance of noise is $3\sigma_v^2$ for $P_{be} = 0.01,\ \text{SNR} = 20\text{dB}$.

Meanwhile, the other of MSE_1 in (6.50) is a squared bias of channel estimate. Given the true system model in (6.45), we can take the expected of $\hat{\mathbf{h}}$ in (6.42) as

$$E\left\{\mathbf{h} \mid \mathbf{h}^{(tr)}\right\} = \left[E\left\{\mathbf{A}\right\}\right]^{-1} E\left\{\mathbf{B} \mid \mathbf{h}^{(tr)}\right\}. \tag{6.73}$$

Similarly using (6.44), (6.45) and a matrix $\mathbf{A}_0$ defined in (6.61), we have

$$E\left\{\mathbf{B} \mid \mathbf{h}^{(tr)}\right\} = E\left\{\frac{1}{N_s}\sum_{i=0}^{\infty}\lambda^i\mathbf{C}^H(p-i)\mathbf{C}(p-i)\mathbf{h}^{(tr)}(p-i)\right\} + \underbrace{E\left\{\frac{1}{N_s}\sum_{i=0}^{\infty}\lambda^i\mathbf{C}^H(p-i)\mathbf{v}_{m_s}(p-i)\right\}}_{=0}$$
$$= \frac{\sigma_s^2}{N_s}\sum_{i=0}^{\infty}\lambda^i\mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)\mathbf{h}^{(tr)}(p-i)$$
$$= \sigma_s^2\sum_{m=1}^{\infty}\lambda^{(m-1)N_s}\mathbf{h}_m^{(tr)}\left[\frac{1}{N_s}\sum_{i=0}^{N_s-1}\lambda^i\mathbf{\Psi}^H(p-i)\mathbf{\Psi}(p-i)\right]$$
$$= \sigma_s^2\sum_{m=1}^{\infty}\lambda^{(m-1)N_s}\mathbf{h}_m^{(tr)}\mathbf{A}_0. \tag{6.74}$$

Using (6.61) and (6.74), we can rewrite (6.73) as

$$E\left\{\mathbf{h} \mid \mathbf{h}^{(tr)}\right\} = (1-\lambda^{N_s})\left[\mathbf{h}_1^{(tr)} + \lambda^{N_s}\mathbf{h}_2^{(tr)} + \cdots\right], \tag{6.75}$$

and it follows that for large N_s (recall we are estimating the most recent true BEM coefficient vector $\mathbf{h}_1^{(tr)}$ via $\hat{\mathbf{h}}$),

$$E\left\{\bar{\mathbf{h}} - \mathbf{h}_1^{(tr)} \mid \mathbf{h}^{(tr)}\right\} \cong \lambda^{N_s}\left[\mathbf{h}_2^{(tr)} - \mathbf{h}_1^{(tr)}\right]. \tag{6.76}$$

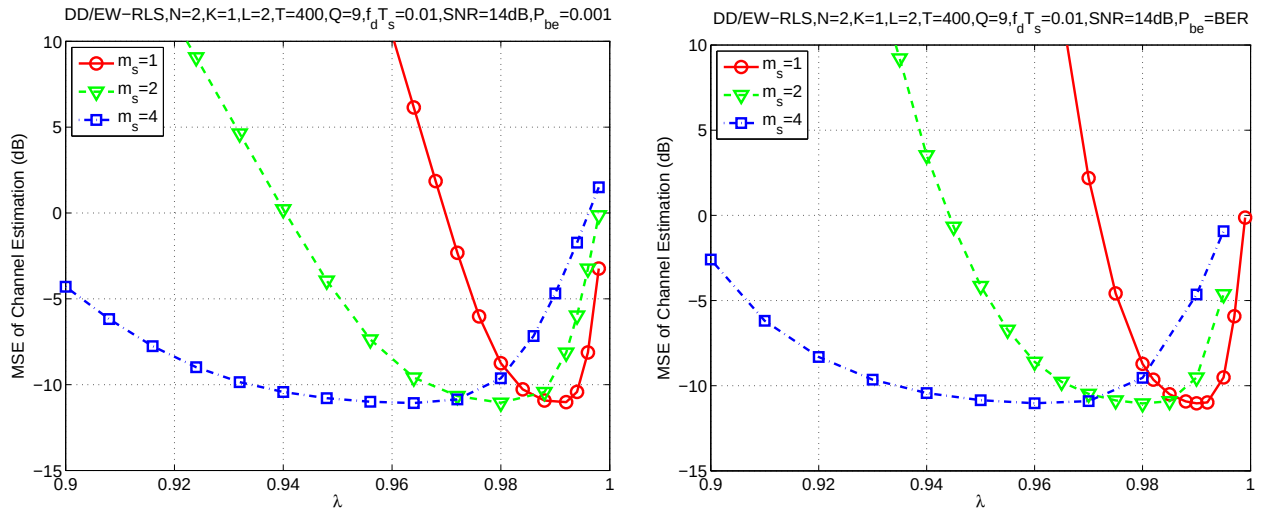
For WSSUS channel, where $\mathbf{h}_1^{(tr)}$ and $\mathbf{h}_2^{(tr)}$ have the same statistics and assuming that $E\left\{\left\|\mathbf{h}_1^{(tr)}\right\|^2\right\} = E\left\{\left\|\mathbf{h}_2^{(tr)}\right\|^2\right\} = 1$, the squared bias of channel estimate in (6.50) can be simplified as

$$\begin{aligned} E\left\{E\left\{\left\|\bar{\mathbf{h}} - \mathbf{h}_1^{(tr)}\right\|^2 \mid \mathbf{h}^{(tr)}\right\}\right\} &= \lambda^{2N_s} E\left\{\left[\mathbf{h}_2^{(tr)} - \mathbf{h}_1^{(tr)}\right]^H \left[\mathbf{h}_2^{(tr)} - \mathbf{h}_1^{(tr)}\right]\right\} \\ &= \lambda^{2N_s}\left(E\left\{\left\|\mathbf{h}_1^{(tr)}\right\|^2\right\} + E\left\{\left\|\mathbf{h}_2^{(tr)}\right\|^2\right\}\right) \\ &= 2\lambda^{2N_s} \end{aligned} \tag{6.77}$$

So far, we obtain the MSE_1 of the channel estimate using (6.50), (6.66) and (6.77),

$$\mathrm{MSE}_1\left(\hat{\mathbf{h}}\right) = tr\left\{\mathrm{cov}\left\{\hat{\mathbf{h}}\right\}\right\} + 2\lambda^{2N_s}, \tag{6.78}$$

and the MSE of channel estimate based on CE-BEM by (6.47), (6.49) and (6.78).



(a) $P_{be} = 0.02$ (b) $P_{be} =$ BER in Fig. 6.9

Figure 6.13: Decision-directed EW-RLS tracking: MSE with different step size m_s's for forgetting factor λ's, under $N = 2, f_dT_s = 0.01, \mathrm{SNR} = 14dB$.

In Fig. 6.13a, a theoretical MSE of EW-RLS decision-directed channel tracking is shown for $N = 2, K = 1, \text{SNR} = 14\text{dB}$ over different λ values in the fast-fading channel ($f_d = 400Hz$). We set $T = 400, Q = 9$ for CE-BEM and $m_s = 1, 2, \text{and } 4$ for the step sizes. We prespecified $P_{be} = 0.002$ in (6.72) for the effective noise variance including the symbol decision errors. One can pick a suitable λ to get the minimum-mean-square-error (MMSE) for one's step size. The MMSE's of different m_s's are similar although the respective suitable range of λ's are different. It is clear the smaller step size (i.e., $m_s = 1$) is more sensitive to λ than the larger ones (i.e., $m_s = 4$) although the decision-directed channel tracking with the smaller one has more complexity.

In Fig. 6.13b, the actual BER values based on 500 Monte Carlo runs in Fig. 6.9 are use for P_{be} in (6.72) to obtain the more realistic analysis. The MSE curves shrink inside since larger P_{be}'s increase the covariance of channel estimates.

6.5 Conclusions

A decision-directed channel tracking approach is investigated for doubly-selective channel in this paper. We exploited a CE-BEM to capture the overall time-variations of the channel, and an AR model to update the BEM coefficients. Since the time-varying nature of the channel is typically well captured in the CE-BEM by the known exponential basis functions, the time-variations of the unknown BEM coefficients are likely much slower than those of channels. Our decision-directed scheme tracks the slow-varying BEM coefficients via Kalman filtering and extended exponentially-weighted RLS, using the detected information symbols of MMSE-DFE as pseudo-trainings. A time gap, arising from the decision-directed tracking, occurred between the symbol decisions and the required channel estimates of the DFE, which could be successfully bridged by channel prediction using the CE-BEM and the estimated BEM coefficients. Simulation examples illustrated that our decision-directed approach has superior performance over the AR channel model-based tracking scheme of [6]; also, it achieves comparable performances to the subblock tracking scheme proposed in Chapter 3 and 4, with much less training overhead.

Chapter 7
Summary and Future Work

7.1 Summary of Original Work

An adaptive channel estimation approach, exploiting TM training and subblock-wise tracking, was presented for frequency-selective time-varying channels. One block of T_B symbols comprises several subblocks of m_b symbols. We employ the CE-BEM to describe the channel. Rather than track each channel tap, we estimate the BEM coefficients subblock-by-subblock and then regenerate the channel via CE-BEM with the estimated BEM coefficients. In this way, the modeling mismatch introduced by the conventionally used symbol-wise AR channel model can be greatly reduced, and hence better performance can be achieved in fast-fading environments. The performance of symbol-wise channel tracking in [68, 76] (without turbo equalization procedure) deteriorates as Doppler spread increases. Since the time-varying nature of the channel is well captured in the CE-BEM by the known exponential basis functions, the time variations of the unknown BEM coefficients are likely much slower than those of the real channel and thus more convenient to track. The existing BEM-based block-wise channel tracking in [70] works well only with the oversampled CE-BEM, which makes it hard to satisfy the parameter identifiability requirement. The BEM-based subblock-wise tracking is insensitive to the parameter identifiability and maintains a satisfactory performance with a flexible subblock size (i.e., a flexible training overhead). Hence the subblock-wise channel tracking using CE-BEM outperforms symbol-wise or block-wise channel estimation schemes.

In Chapter 3, the Kalman filter-based subblock-wise tracking was proposed, where we assume the BEM coefficients follow a first-order AR model. We analyzed the choice of the AR(1) coefficient for the subblock-wise Kalman channel estimation. However, this first-order AR assumption is an arbitrary *a priori* model that is not necessarily true for a real-world channel and possibly incurs modeling error in channel estimation. With no *a priori* models for the BEM coefficients, we considered adaptive filtering algorithms with finite memory: EW-RLS and SW-RLS algorithms. It was shown that they are superior to Kalman filtering for the subblock-wise channel tracking. In particular, subblock-wise EW-RLS tracking is a good alternative since it has the same computational complexity as Kalman tracking. We also analyze the theoretical choice of the forgetting factor λ in EW-RLS for the subblock-wise tracking using CE-BEM.

In Chapter 5, we investigated the BEM-based approach to coded modulation communication systems using turbo equalization receiver. Based on the subblock-wise BEM-based approach, we extended the turbo equalization approach of [68] using AR modeling of channels to channels using CE-BEM where an adaptive turbo equalizer with non-linear Kalman filtering is coupled with a soft-in soft-out (SfiSfo) decoder to iteratively perform equalization and decoding using soft information feedback. Assuming the BEM coefficients follow an AR(1) model, the adaptive equalizer jointly optimizes the estimates of the BEM coefficients and data symbols, thereby automatically accounting for the correlation between the channel estimates and data symbols. The proposed CE-BEM-based turbo equalizer has better performances than the AR-model-based approach in [68], even with the same computational complexity. We need only a few iterations to have a performance close to the optimal maximum *a posteriori* (MAP) approach or the turbo equalizer with perfect knowledge of the true channels. Since the time-variations of the BEM coefficients are likely much slower than those of the real channels, the BEM-based approach is not so sensitive to the Doppler spread as the AR-model-based one. We also provided EXIT chart analyses that can be used to infer performance comparisons without running the actual simulations.

The decision-directed tracking of doubly-selective channels using CE-BEM was also investigated. In [6], the decision-directed scheme was proposed that relies on the AR model only. Acting as virtual training symbols, the information symbol decisions, provided by a DFE (with delay $d \geq 0$) utilizing the estimated channel, were used to enhance the estimation of the BEM coefficients, so that much of the spectrum resource allocated to training can be saved. Although a time gap still exists between the available symbol decisions and the channel estimates required by the DFE, it can be successfully bridged by the CE-BEM-based channel prediction, without incurring much estimation variance. Similar to the other BEM-based approaches, we exploited CE-BEM to capture the overall time-variations of the channel and the first-order AR model to update the BEM coefficients. Our decision-directed scheme updates the BEM coefficients for the step size m_s, which is much smaller than the BEM period (i.e., $m_s \ll T$), via the Kalman filtering or the extended EW-RLS algorithm. With a larger subblock size (i.e., a small training overhead), the CE-BEM-based decision-directed tracking outperforms the subblock-wise tracking of Chapter 3 and Chapter 4. The AR-model-based decision-directed scheme in [6] with the comparable complexity (letting $P = Q$) performs better only for the small range of slow fading. Since the BEM-based decision-directed tracking updates the BEM coefficients for the step size $m_s \geq 1$ symbol, we can reduce the computational complexity by increasing the step size (e.g., $m_s = 2$ or 4 symbols) with little loss in performance. It was shown that the approach of [6] based on an AR model

only has serious performance loss with increasing Doppler spread. However, the CE-BEM-based decision-directed approach maintains satisfactory performance and thereby is more reliable in practice, where we do not know the actual Doppler spread. We also presented a performance analysis to provide guidelines for the theoretical choice of the forgetting factor λ for the decision-directed tracking using CE-BEM.

All the channel estimation and equalization schemes were extended to MIMO systems, and simulation examples were provided for performance comparisons based on Monte Carlo simulations.

7.2 Suggested Future Work

So far we have discussed doubly-selective channel estimation and equalization using exponential basis models, including subblock-wise channel tracking, turbo equalization and decision-directed tracking. Future work may include the following areas.

First, one should investigate other training schemes for the subblock-wise tracking in Chapter 3 and 4 that is more efficient than the time-multiplexed (TM) training. The TM training may waste resources for long channel length or multiple users. Since the subblock-wise channel estimation depends only on the training session, it is important to find a training scheme that can save system resources and overcome the user interference in a MIMO system.

For the turbo equalization using the extended Kalman filter in Chapter 5, one may reduce the complexity by increasing the number of symbols for one recursion of the channel estimation and detection algorithm, namely, use step size $m_s > 1$. Since the neighboring symbols have very similar BEM coefficients in CE-BEM with large block size ($T_B \gg m_s$), we do not need to update the BEM coefficient every symbol. Although the algorithm can be more "complicated", the overall computational complexity decreases with little loss of performances; we already have similar examples in decision-directed tracking with $m_s = 1, 2$ and 4 in Chapter 6.

In Section 6.4, the optimal value for the forgetting factor in EW-RLS decision-directed tracking varies with SNR; as SNR increases, the optimal λ becomes smaller (farther from one) due to the lower covariance of the channel estimate in (6.66). One can expect improvement in performance by using a variable forgetting factor dependent on the measured SNR. For instance, [7, 25, 58] investigated the variable forgetting factor RLS algorithm. The Kalman filter with a variable forgetting factor has also been proposed in some papers [24, 34].

Finally, one may use other basis expansion models instead of the Fourier basis function based CE-BEM, i.e., discrete prolate spheroidal basis expansion model (DPS-BEM) in [62,63] or the orthogonal polynomial basis expansion model (OP-BEM) in [8, 26].

Bibliography

[1] A. Anastassopoulos and K. Chugg, "Iterative equalization/decoding for TCM for frequency-selective channels," *Proc. Asilomar*, vol. 1, pp. 177-181, Nov. 1997.

[2] A.H. Sayed, *Fundamentals of Adaptive Filtering*, New York, NJ: Wiley, 2003.

[3] A. Lozano and C. Papadias, "Layered space-time receivers for frequency-selective wireless channels," *IEEE Trans. Commun.*, vol. 50, pp. 65-73, Jan. 2002.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. on Commun.*, Geneva, Switzerland, May 1993.

[5] 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, Sept.-Oct. 1995.

[6] 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.

[7] C. Paleologu, J. Benesty, and S. Ciochina, "A Robust Variable Forgetting Factor Recursive Least-Squares Algorithm for System Identification," *IEEE Signal Processing Letters*, vol. 15, pp. 597-600, 2008.

[8] D.K. Borah and B.D. Hart, "Frequency-selective fading channel estimation with a polynomial time-varying channel model," *IEEE Tran. Comm.*, vol. 47, no. 6, pp. 862-873, Jun. 1999.

[9] E. Karami and M. Shiva, "Blind multi-input–multi-output channel tracking using decision-directed maximum-likelihood estimation," *IEEE Trans. Veh. Technol.*, vol. 56, no. 3, pp. 1447-1454, May 2007.

[10] 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.

[11] 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, pp. 1969-1986, Nov. 1998.

[12] G. Bauch and N. Al-Dhahir, "Reduced complexity space-time turbo equalization for frequency-selective MIMO channels," *IEEE Trans. Wireless Commun.*, vol. 1, pp. 819-828, Oct. 2002.

[13] G. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.

[14] G. Leus, "On the estimation of rapidly time-varying channels," in *Proc. European Signal Proc. Conf.*, pp. 2227-2230, Vienna, Austria, Sept. 6-10, 2004.

[15] G.L. Stüber, *Principles of Mobile Communication*, 2nd ed., Boston, MA: Kluwer, 2002.

[16] 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, no. 2, pp. 353-357, May 1996.

[17] I. Barhumi, G. Leus, and M. Moonen, "Time-varying FIR decision feedback equalization of doubly-selective channels," in *Proc. IEEE Global Commun. Conf*, pp. 2263-2268, San Francisco, CA, Dec. 2003.

[18] I. Barhumi, G. Leus, and M. Moonen, "Estimation and direct equalization of doubly selective channels," in *EURASIP Journal on Applied Signal Process.*, article ID: 62831, pp. 1-15, vol. 2006.

[19] J.G. Proakis, *Digital Communications*, 4th ed., NY: McGraw-Hill, 2001.

[20] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing*, 3rd Ed., Upper Saddle River, NJ: Prentice-Hall, 1996.

[21] J. Gao and H. Liu, "Decision-directed estimation of MIMO time-varying Rayleigh fading channels," *IEEE Trans. Signal Process.*, vol. 4, no. 4, pp. 1412-1417, Jul. 2005.

[22] J.K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 686-693, Nov. 1991.

[23] J.K. Tugnait and S. He, "Recursive least-squares doubly-selective channel estimation using exponential basis models and subblock-wise tracking," in *Proc. IEEE ICASSP 2008*, Las Vegas, NV, pp. 2861-2864, Mar. 31 - Apr. 4, 2008.

[24] J. Lim, J. Song and K. Sung, "Forward-backward time varying forgetting factor Kalman Filter based DOA estimation algorithm for UAV (unmanned aerial vehicle) autolanding," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing 2002 (ICASSP '02)*, vol. 4, pp. IV-3964 – IV-3967, 2002.

[25] Junfeng Wang, "A variable forgetting factor RLS adaptive filtering algorithm," *3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Commun.*, pp. 1127-1130, Oct. 27-29, 2009.

[26] K.A.D. Teo, S. Ohno and T. Hinamoto, "Kalman channel estimation based on oversampled polynomial model for OFDM over doubly-selective channel," in *Proc. IEEE SPAWC*, New York, NY, June 2005, pp. 116-120.

[27] K.E. Baddour and N.C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Trans. Wireless Communications*, vol. 4, pp. 1650-1662, July 2005.

[28] 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, Sept. 2006.

[29] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, March 1974.

[30] L. Yang, X. Ma, and G.B. Giannakis, "Optimal training for MIMO fading channels with time- and frequency- selectivity," in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Process.(ICASSP) 2004*, vol. 3, pp. 17-21, May 2004.

[31] M. Dong, L. Tong and B.M. Sadler, "Optimal insertion of pilot symbols for transmissions over time-varying flat fading channels," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1403-1418, May 2004.

[32] M.D. Srinath, P.K. Rajasekaran and R. Viswanathan, *Introduction to Statistical Signal Processing with Applications*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[33] M.F. Pop and N.C. Beaulieu, "Limitations of sum-of-sinusoids fading channel simulators," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 699-708, Apr. 2001.

[34] M.G. Pappas and J.E. Doss, "Design of a Parallel Kalman Filter with Variable Forgetting Factors," *American Control Conference, 1988*, pp. 2368-2372, June 15-17, 1988.

[35] M. Martone, "Wavelet-based separating kernels for array processing of cellular DS/CDMA signals in fast fading," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 979-995, Jun. 2000.

[36] 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.

[37] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. Conference on Information Science and Systems*, CISS 2002, March 20-22, 2002.

[38] M. Tüchler, A. Singer and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 673-683, March 2002.

[39] M. Tüchler, R. Koetter and A. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754-767, May 2002.

[40] N. Al-Dhahir and J.M. Cioffi, "MMSE decision-feedbck equalizers: finite-length results," *IEEE Trans. Inf. Theory*, vol. 41, no. 4, pp. 961-975, Jul. 1995.

[41] P. Banelli, R.C. Cannizzaro and L. Rugini, "Data-aided Kalman tracking for channel estimation in Doppler-affected OFDM systems," in *Proc. IEEE ICASSP 2007*, vol. 3, Honolulu, HI, April 2007, pp. 133-136.

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

[43] R. Bosisio, M. Nicoli and U. Spagnolini, "Kalman filter of channel modes in time-varying wireless systems," in *Proc. IEEE ICASSP '05*, vol. 3, pp. 785-788, Philadelphia, PA, Mar. 18-23, 2005.

[44] R.C. Cannizzaro and P. Banelli and G. Leus, "Adaptive channel estimation for OFDM systems with Doppler spread," in *Proc. IEEE Workshop Signal Process. Advances in Wireless Commun.*, Cannes, France, Jul. 2-5, 2006.

[45] R.H. Clarke, "A statistical theory of mobile-radio reception," *Bell Syst. Tech. J.*, vol. 47, no. 6, pp. 957-1000, July-Aug. 1968.

[46] R. Koetter, A.C. Singer and M. Tüchler, "Turbo Equalization," *IEEE Signal Proc. Mag.*, vol. 21, pp. 67-80, 2004.

[47] 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. 4-7, 2002.

[48] 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.

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

[50] S. Haykin, *Adaptive Filter Theory*, 4rd Ed., Upper Saddle River, NJ: Prentice Hall, 2001.

[51] S. He, "Doubly Selective Channel Estimation and Equalization using Superimposed Training and Basis Expansion Models," Ph.D. Dissertation, Auburn University, Auburn, AL, 2007.

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

[53] S. Kalyani and K. Giridhar, "Mitigation of error propagation in decision directed OFDM channel tracking using generalized M estimators," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1659-1672, May 2007.

[54] S. Lin and J.J. Costello, *Error Control Coding.*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

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

[56] S. Roy and T.M. Duman, "Soft input soft output Kalman equalizer for MIMO frequency selective fading channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 506-514, Feb. 2007.

[57] 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.

[58] S. Song, J. Lim, S. Baek and K. Sung, "Variable forgetting factor linear least squares algorithm for frequency selective fading channel estimation," *IEEE Trans. on Veh. Technol.*, vol. 51, no. 3, pp. 613-616, May 2002.

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

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

[61] T. Söderström, P. Stoica, *System Identification*, Prentice-Hall Intern.: New York, NY, 1989.

[62] T. Zemen and C.F. Mecklenbräuker, "Time-variant channel equalization via discrete prolate spheroidal Sequences," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, invited paper, pp. 1288-1292, Nov. 2003.

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

[64] W. Chen and R. Zhang, "Estimation of time and frequency selective channels in OFDM systems: a Kalman filter structure," in *Proc. IEEE GLOBECOM '04*, vol. 2, pp. 800-803. Dallas, TX, Nov. 29 - Dec. 3, 2004.

[65] W.C. Jakes, *Microwave Mobile Communications*, New York, NY: Wiley, 1974.

[66] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time varying intersymbol interference," in *Proc. of the IEEE Global Telecomm. Conf.*, Dec. 1990, pp. 1679–1684.

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

[68] 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.

[69] X. Ma and G.B. Giannakis, "Maximum-diveristy transmissions over doubly selective wireless channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1832-1840, July 2003.

[70] 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.

[71] X. Wang and H. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046-1061, July 1999.

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

[73] X. Zhu and R. Murch, "Layered space-time equalization for wireless MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 10, pp. 57-60, no. 3, Mar. 2003.

[74] Y. Li, B. Vucetic, and Y. Sato, "Optimum soft-ouput detection for channel with inter-symbol interference," *IEEE Trans. Inform. Theory*, vol. 41, pp. 704–713, May 1995.

[75] 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.

[76] 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.

[77] Z. Tang, R.C. Cannizzaro, G. Leus and P. Banelli, "Pilot-assisted time-varying channel estimation for OFDM systems," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2226-2238, May 2007.