A Low-Complexity Iterative Channel Estimation and Detection Technique for Doubly Selective Channels

Qinghua Guo and Li Ping
City University of Hong Kong, Hong Kong SAR, China
Email: eeqinghua@gmail.com, eeliping@cityu.edu.hk

Abstract— **In this paper, we propose a low-complexity iterative joint channel estimation, detection and decoding technique for doubly selective channels. The key is a segment-by-segment frequency domain equalization (FDE) strategy under the assumption that channel is approximately static within a short segment. Guard gaps (for cyclic prefixing or zero padding) are not required between adjacent segments, which avoids the power and spectral overheads due to the use of cyclic prefix (CP) in the conventional FDE technique. A low-complexity bi-directional channel estimation algorithm is also developed to exploit correlation information of time-varying channels. Simulation results are provided to demonstrate the efficiency of the proposed algorithms.**

I. INTRODUCTION

Single-carrier block transmission with frequency domain equalization (FDE) [1] is an efficient technique to alleviate inter-symbol interference (ISI) in multipath channels. The use of cyclic prefix (CP) avoids inter-block interference and converts linear convolution to cyclic convolution, which allows efficient implementation of receivers based on fast Fourier transform (FFT). However, the use of CP incurs overheads (in terms of both power and spectral efficiency loss) that can be measured by the following ratio:

CP length / block length.

In the conventional FDE technique, this ratio is limited by the following two requirements:

- The channel should be static within a block (and so the block length is limited by channel coherent time);
- CP should be longer than channel memory length.

Due to the above two requirements, the overhead ratio can be high in doubly selective channels (i.e., time-varying ISI channels) when channel coherent time is small and channel memory is long.

Using shorter CP is a way to reduce overhead. However, CP length less than channel memory length may cause interference among consecutive blocks, and the assumption of cyclic convolution for FDE is also invalid in this case. Remedies for these problems have been studied in [2-4].

In this paper, we propose a novel detection technique for doubly selective channels, in which each block of the transmitted signal is partitioned into a number of short segments $\{\boldsymbol{x}_k\}$ as shown in Fig. 1(a). There is no guard interval between two consecutive segments, which avoids the related overhead. The signal in a block is transmitted continuously in the same way as a conventional scheme, i.e., the segmentation does not affect the structure of the transmitted signal. We assume that the channel remains static within a segment but not necessary within a block. The received signal is shown in Fig. 1(b) where the observation vector $\boldsymbol{r}_k$ covers all the contribution of segment $\boldsymbol{x}_k$. Due to delay spread, $\boldsymbol{r}_k$ is longer than $\boldsymbol{x}_k$, so it suffers from the interference form its adjacent segments $\boldsymbol{x}_{k-1}$ and $\boldsymbol{x}_{k+1}$. In this paper, an iterative technique is developed to handle such interference. Since the block length is not limited by the channel coherent time, it can be large enough to ensure a negligible overhead caused by the guard interval.

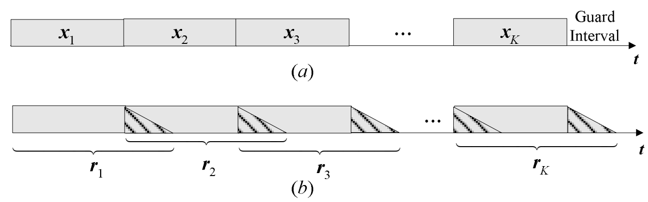

Fig. 1 (a) The transmitted signal $\boldsymbol{x}$ is partitioned into a number of short segments $\{\boldsymbol{x}_k\}$. Each segment is assumed to undergo a static ISI channel. (b) ISI causes interference among adjacent segments, as illustrated by the shadowing parts.

We will also develop a low-complexity bi-directional channel estimation algorithm that can be incorporated in the iterative process for joint channel estimation, equalization and decoding. This provides a solution for channel estimation that is another challenging problem in doubly selective channels.

Simulation results are provided to demonstrate the efficiency of the proposed technique.

The notations used in this paper are as follows. Lower case letters denote scalars, bold lower case letters denote column vectors, and bold upper case letters denote matrices. We use superscript "^{T}" to denote transpose, "*" conjugate and "^{H}" conjugate transpose. $\boldsymbol{I}$ denotes an identity matrix with proper size. Expectation and (co)variance are denoted by E(•) and V(•), respectively. For a complex variable, e.g., x, we use x^{Re} to denote its real part and x^{Im} its imaginary part.

II. PRELIMINARY

In this section, we provide a brief outline of the underlying MMSE estimation principle, and list the key results to be used in the following sections.

A. MMSE Estimation for Gaussian Variables

Consider a standard estimation problem based on the following linear model

$$\boldsymbol{r} = \boldsymbol{A}\boldsymbol{h} + \boldsymbol{n}\,, \tag{1}$$

where $\boldsymbol{r}$ is an observation vector, $\boldsymbol{A}$ a system transfer matrix, $\boldsymbol{h}$ a vector to be estimated, and $\boldsymbol{n}$ a sample vector of Gaussian noise.

Assume that $\boldsymbol{h}$ is Gaussian distributed and its *a priori* mean vector and covariance matrix are denoted by $\mathrm{E}(\boldsymbol{h})$ and $\mathrm{V}(\boldsymbol{h})$, respectively. Then the *a posteriori* mean and variance of $\boldsymbol{h}$ can be computed as [10]

$$\mathrm{E}^p(\boldsymbol{h}) = \mathrm{E}(\boldsymbol{h}) + \mathrm{V}(\boldsymbol{h})\boldsymbol{A}^H\left(\boldsymbol{A}\mathrm{V}(\boldsymbol{h})\boldsymbol{A}^H + \mathrm{V}(\boldsymbol{n})\right)^{-1}\left(\boldsymbol{r} - \boldsymbol{A}\mathrm{E}(\boldsymbol{h}) - \mathrm{E}(\boldsymbol{n})\right), \tag{2a}$$

$$\mathrm{V}^p(\boldsymbol{h}) = \left(\mathrm{V}(\boldsymbol{h})^{-1} + \boldsymbol{A}^H\mathrm{V}(\boldsymbol{n})^{-1}\boldsymbol{A}\right)^{-1}. \tag{2b}$$

B. Joint Gaussian Estimation for Binary Variables

The estimation becomes much more complicated for the following linear model

$$\boldsymbol{r} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}\,, \tag{3}$$

where the entries in $\boldsymbol{x}$ are binary.

We first assume that all the variables involved in (3) are real. In this case, the estimation is usually given in an entry-by-entry extrinsic logarithm of likelihood ratio (LLR) form as [5], [6], [14]

$$e(v_j) = \ln\frac{p(\boldsymbol{r}\mid x_j = +1)}{p(\boldsymbol{r}\mid x_j = -1)}\,, \tag{4}$$

where x_j is the jth entry of $\boldsymbol{x}$. The optimal approach to computing (4) is based on the maximum a *posterior*i probability (MAP) criterion, but its complexity is usually prohibitive. A low-complexity sub-optimal alternative is the so-called joint Gaussian (JG) approach [14], in which (3) is rewritten in the following form,

$$\boldsymbol{r} = \boldsymbol{a}_j x_j + \boldsymbol{\xi}_j\,, \tag{5}$$

where $\boldsymbol{a}_j$ is the jth column of $\boldsymbol{A}$ and

$$\boldsymbol{\xi}_j = \sum\nolimits_{j'\neq j}\boldsymbol{a}_{j'}x_{j'} + \boldsymbol{n}\,. \tag{6}$$

Note that $\boldsymbol{\xi}_j$ is the sum of the contributions of all entries in $\boldsymbol{x}$ except x_j and noise. By the central limit theorem, we can approximate the entries of $\boldsymbol{\xi}_j$ as joint Gaussian variables with

$$\mathrm{E}(\boldsymbol{\xi}_j) = \boldsymbol{A}\mathrm{E}(\boldsymbol{x}) - \boldsymbol{a}_j\mathrm{E}(x_j) + \mathrm{E}(\boldsymbol{n})\,, \tag{7a}$$

$$\mathrm{V}(\boldsymbol{\xi}_j) = \mathrm{V}(\boldsymbol{r}) - \mathrm{V}(x_j)\boldsymbol{a}_j\boldsymbol{a}_j^T\,, \tag{7b}$$

$$\mathrm{V}(\boldsymbol{r}) = \boldsymbol{A}\mathrm{V}(\boldsymbol{x})\boldsymbol{A}^T + \mathrm{V}(\boldsymbol{n})\,. \tag{7c}$$

Based on the above assumption, (4) can be computed as

$$\begin{aligned} e(x_j) &= \ln\frac{\exp\left[-\frac{1}{2}(\boldsymbol{r} - \boldsymbol{a}_j - \mathrm{E}(\boldsymbol{\xi}_j))^T\mathrm{V}(\boldsymbol{\xi}_j)^{-1}(\boldsymbol{r} - \boldsymbol{a}_j - \mathrm{E}(\boldsymbol{\xi}_j))\right]}{\exp\left[-\frac{1}{2}(\boldsymbol{r} + \boldsymbol{a}_j - \mathrm{E}(\boldsymbol{\xi}_j))^T\mathrm{V}(\boldsymbol{\xi}_j)^{-1}(\boldsymbol{r} + \boldsymbol{a}_j - \mathrm{E}(\boldsymbol{\xi}_j))\right]} \\ &= 2\boldsymbol{a}_j^T\mathrm{V}(\boldsymbol{\xi}_j)^{-1}\left(\boldsymbol{r} - \boldsymbol{A}\mathrm{E}(\boldsymbol{x}) - \mathrm{E}(\boldsymbol{n}) + \boldsymbol{a}_j\mathrm{E}(x_j)\right) \\ &= 2\frac{\boldsymbol{a}_j^T\mathrm{V}(\boldsymbol{r})^{-1}\left(\boldsymbol{r} - \boldsymbol{A}\mathrm{E}(\boldsymbol{x}) - \mathrm{E}(\boldsymbol{n})\right) + \boldsymbol{a}_j^T\mathrm{V}(\boldsymbol{r})^{-1}\boldsymbol{a}_j\mathrm{E}(x_j)}{1 - \mathrm{V}(x_j)\boldsymbol{a}_j^T\mathrm{V}(\boldsymbol{r})^{-1}\boldsymbol{a}_j}\,. \end{aligned} \tag{8}$$

The result (8) is the same as that in the so-called LMMSE approach derived in [5] and [6]. Compared with the derivation in [5] and [6], the derivation described above is more concise and straightforward.

The above result can be extended to complex systems. In this case, denote $x_j = x_j^{\mathrm{Re}} + i x_j^{\mathrm{Im}}$ where $i = \sqrt{-1}$ and $\{x_j^{\mathrm{Re}},\ x_j^{\mathrm{Im}}\}$ are binary. Define

$$e(x_j) = e(x_j^{\mathrm{Re}}) + i e(x_j^{\mathrm{Im}})\,, \tag{9}$$

where

$$e(x_j^{\mathrm{Re}}) = \ln\frac{p(\boldsymbol{r}\mid x_j^{\mathrm{Re}} = +1)}{p(\boldsymbol{r}\mid x_j^{\mathrm{Re}} = -1)}, \text{ and } e(x_j^{\mathrm{Im}}) = \ln\frac{p(\boldsymbol{r}\mid x_j^{\mathrm{Im}} = +1)}{p(\boldsymbol{r}\mid x_j^{\mathrm{Im}} = -1)}\,. \tag{10}$$

Let $\mathrm{V}(x_j^{\mathrm{Re}}) = \mathrm{V}(x_j^{\mathrm{Im}}) = 0.5\mathrm{V}(x_j)$ and assume $\mathrm{V}(x_j^{\mathrm{Re}}, x_j^{\mathrm{Im}}) = 0$ (i.e., the real and imaginary parts of x_j are uncorrelated). Then $e(x_j)$ can be calculated as

$$e(x_j) = 4\frac{\boldsymbol{a}_j^H\mathrm{V}(\boldsymbol{r})^{-1}\left(\boldsymbol{r} - \boldsymbol{A}\mathrm{E}(\boldsymbol{x}) - \mathrm{E}(\boldsymbol{n})\right) + \boldsymbol{a}_j^H\mathrm{V}(\boldsymbol{r})^{-1}\boldsymbol{a}_j\mathrm{E}(x_j)}{1 - \mathrm{V}(x_j)\boldsymbol{a}_j^H\mathrm{V}(\boldsymbol{r})^{-1}\boldsymbol{a}_j}\,, \tag{11}$$

or in a vector form as (letting $e(\boldsymbol{x}) = [e(x_0), e(x_1), \ldots]^T$)

$$e(\boldsymbol{x}) = 4(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{U})^{-1}\left(\boldsymbol{A}^H\mathrm{V}(\boldsymbol{r})^{-1}\left(\boldsymbol{r} - \boldsymbol{A}\mathrm{E}(\boldsymbol{x}) - \mathrm{E}(\boldsymbol{n})\right) + \boldsymbol{U}\mathrm{E}(\boldsymbol{x})\right), \tag{12}$$

where

$$\mathrm{V}(\boldsymbol{r}) = \boldsymbol{A}\mathrm{V}(\boldsymbol{x})\boldsymbol{A}^H + \mathrm{V}(\boldsymbol{n}), \tag{13a}$$

$$\boldsymbol{V} = \mathrm{diag}\{\mathrm{V}(x_0), \mathrm{V}(x_1), \ldots\}, \tag{13b}$$

$$\boldsymbol{U} = (\boldsymbol{A}^H\mathrm{V}(\boldsymbol{r})^{-1}\boldsymbol{A})_{diag}. \tag{13c}$$

In (13c), the operator $(\cdot)_{\mathrm{diag}}$ returns a diagonal matrix consisting of the diagonal elements of the matrix in the parentheses. For space limitation, we omit the derivation of (11).

III. THE OVERALL JOINT PROCESS

In this section, we introduce the signal model and the framework of joint channel estimation, detection and decoding. The detailed estimation and detection algorithms will be discussed in the following sections.

Consider a time-varying (complex) ISI channel model

$$r_j = \sum_{l=0}^{L} h_j^l \cdot x_{j-l} + \eta_j\,, \tag{14}$$

where $\{x_k\}$ are the transmitted signal formed by the outputs of a forward error correction (FEC) encoder with quadrature phase shift keying (QPSK) and Gray mapping, $\{r_j\}$ the observations, $\{\eta_j\}$ the samples of additive whiten Gaussian noise with zero mean and variance $2\sigma^2$, and $\{h_j^l,\ l = 0,1,\ldots, L\}$ the channel state information (CSI) at time j.

Return to Fig. 1 where the transmitted signal $\boldsymbol{x}$ is partitioned into K short segments $\{\boldsymbol{x}_k\}$, each with length M. We assume that each segment $\boldsymbol{x}_k$ undergoes a static ISI channel $\boldsymbol{h}_k = [h_k^0, h_k^1, \ldots, h_k^L]^T$. As shown in Fig. 1, the observation vector related to $\boldsymbol{x}_k$ can be represented in a convolution form as

$$\boldsymbol{r}_k = \boldsymbol{h}_k * \boldsymbol{x}_k + \boldsymbol{y}_{k-1}^{Inter} + \boldsymbol{y}_{k+1}^{Inter} + \boldsymbol{\eta}_k\,, \tag{15}$$

where "*" denotes linear convolution operation. Note that the length of $\boldsymbol{r}_k$ is M+L (see Fig. 1(b)), and all the information about $\boldsymbol{x}_k$ is included in $\boldsymbol{r}_k$.[1] In (15), $\boldsymbol{y}_{k-1}^{Inter}$ and $\boldsymbol{y}_{k+1}^{Inter}$ represent the interference from the adjacent segments $\boldsymbol{x}_{k-1}$ and $\boldsymbol{x}_{k+1}$ respectively, and they can be expressed as $\boldsymbol{y}_{k-1}^{Inter} = [tail(\boldsymbol{h}_{k-1} * \boldsymbol{x}_{k-1}), 0, \ldots, 0]^T$, and $\boldsymbol{y}_{k+1}^{Inter} = [0, \ldots, 0, head(\boldsymbol{h}_{k+1} * \boldsymbol{x}_{k+1})]^T$

[1] Here the detection of $\boldsymbol{x}_k$ is based on $\boldsymbol{r}_k$. In contrast, the detection of $\boldsymbol{x}_k$ discussed in [2-4] is based on the first M elements of $\boldsymbol{r}_k$, and hence some useful information is lost.

where $tail(\cdot)$ and $head(\cdot)$ represent two truncation functions that return the tail part and head part of the sequence in the parentheses, respectively. The length of tail/head part is L. We rewrite (15) in a more compact form as

$$\boldsymbol{r}_k = \boldsymbol{h}_k * \boldsymbol{x}_k + \boldsymbol{n}_k \,, \tag{16a}$$

with

$$\boldsymbol{n}_k = \boldsymbol{y}_{k-1}^{Inter} + \boldsymbol{y}_{k+1}^{Inter} + \boldsymbol{\eta}_k \,. \tag{16b}$$

We assume that a guard interval in the form of zero padding is appended to each block (see Fig. 1(a)). Thus the last segment $\boldsymbol{x}_K$ only sees the interference from $\boldsymbol{x}_{K-1}$ and there is no inter-block interference. This allows independently processing the blocks.

The iterative receiver is shown in the lower part of Fig. 2. It consists of three modules: a channel estimator, a channel equalizer and a decoder. We follow the framework of iterative channel estimation and detection detailed in [7] and [8]. The following is a brief outline of the function of each module in the iterative process.

- The channel estimator provides the estimates of $\{h_k^l\}$ (in the form of $\{\mathrm{E}(h_k^l)\}$ and $\{\mathrm{V}(h_k^l)\}$) based on the output of the channel decoder and the statistical property of the time-varying channel.
- Based on the *a priori* information about $\{x_j\}$ from the channel decoder and the channel estimates from the channel estimator, the channel equalizer computes the extrinsic LLRs for $\{x_j\}$ as given in (11).
- Based on the output of the channel equalizer and the FEC coding constraint, the channel decoder refines the data estimates. We assume that standard *a posteriori* probability (APP) decoding [15] is used.

The above three modules work in an iterative manner. The decoder makes hard decisions on the information bits during the final iteration. Refer to [5-9] for the detailed discussions for the above iteration process.

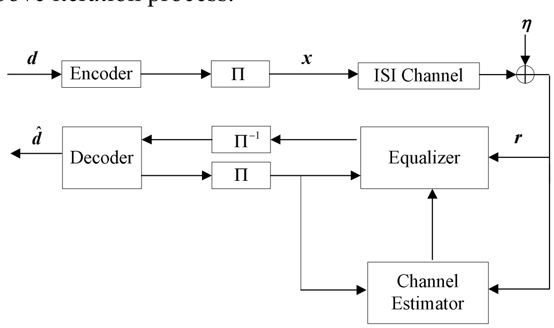

Fig. 2. The transmitter and turbo receiver. Π and Π^{-1} denote interleaver and de-interleaver, respectively.

In the following, we will discuss the details for the channel estimator and equalizer, respectively.

IV. CHANNEL ESTIMATION

Rewrite (16) in a matrix form as

$$\boldsymbol{r}_k = \boldsymbol{A}_k \boldsymbol{h}_k + \boldsymbol{n}_k \,, \tag{17}$$

where $\boldsymbol{A}_k$ is formed based on $\boldsymbol{x}_k$ and $\boldsymbol{h}_k = [h_k^0, h_k^1, \ldots, h_k^L]^T$.

First, we assume that $\{\boldsymbol{x}_k\}$ are perfect known. Based on (17), we can estimate $\boldsymbol{h}_k$ using the standard MMSE estimator (2). Alternatively, to avoid matrix inversion, we can employ the following low-complexity tap-by-tap estimation technique.

We focus on h_k^l and rewrite (17) as

$$\boldsymbol{r}_k = \boldsymbol{a}_{k,l} h_k^l + \boldsymbol{\zeta}_{k,l} \,, \tag{18}$$

where $\boldsymbol{a}_{k,l}$ is the lth column of $\boldsymbol{A}_k$, and

$$\boldsymbol{\zeta}_{k,l} = \sum_{l' \neq l} \boldsymbol{a}_{k,l'} h_k^{l'} + \boldsymbol{n}_k \,, \tag{19}$$

represents the noise plus interference from other taps. Its mean vector and covariance matrix are given by

$$\mathrm{E}(\boldsymbol{\zeta}_{k,l}) = \sum_{l' \neq l} \boldsymbol{a}_{k,l'} \mathrm{E}(h_k^{l'}) + \mathrm{E}(\boldsymbol{n}_k) \,, \tag{20a}$$

$$\mathrm{V}(\boldsymbol{\zeta}_{k,l}) = \boldsymbol{A}_k \mathrm{V}(\boldsymbol{h}_k) \boldsymbol{A}_k^H + \mathrm{V}(\boldsymbol{n}_k) \,. \tag{20b}$$

In general $\mathrm{V}(\boldsymbol{\zeta}_{k,l})$ is a full matrix. To reduce complexity, we approximate $\mathrm{V}(\boldsymbol{\zeta}_{k,l})$ using its diagonal part

$$\boldsymbol{D}_{k,l} = \left(\mathrm{V}(\boldsymbol{\zeta}_{k,l})\right)_{diag} \,. \tag{21}$$

Based on the above approximation and (2), we have

$$\mathrm{E}^p(h_k^l) \approx \mathrm{E}(h_k^l) + \frac{\mathrm{V}(h_k^l) \boldsymbol{a}_{k,l}^H \boldsymbol{D}_{k,l}^{-1} (\boldsymbol{r}_k - \mathrm{E}(\boldsymbol{\zeta}_{k,l}) - \boldsymbol{a}_{k,l} \mathrm{E}(h_k^l))}{\mathrm{V}(h_k^l) \boldsymbol{a}_{k,l}^H \boldsymbol{D}_{k,l}^{-1} \boldsymbol{a}_{k,l} + 1} \,, \tag{22a}$$

$$\mathrm{V}^p(h_k^l) \approx \frac{\mathrm{V}(h_k^l)}{\mathrm{V}(h_k^l) \boldsymbol{a}_{k,l}^H \boldsymbol{D}_{k,l}^{-1} \boldsymbol{a}_{k,l} + 1} \,. \tag{22b}$$

In the above equations, $\mathrm{E}(h_k^l)$ and $\mathrm{V}(h_k^l)$ denote the *a priori* mean and variance of the concerned variable h_k^l, and $\mathrm{E}(\boldsymbol{\zeta}_{k,l})$ and $\boldsymbol{D}_{k,l}$ are related to noise plus interference.

In the iterative process, the channel estimation results in the last iteration can be used as *a priori* information for the current iteration. Specifically, $\left\{\mathrm{E}^p(h_k^{l'}), \mathrm{V}^p(h_k^{l'}), l' \neq l\right\}$ calculated in the last iteration can be used to update $\mathrm{E}(\boldsymbol{\zeta}_{k,l})$ and $\boldsymbol{D}_{k,l}$ in (22) based on (20) and (21).[2] In contrast, $\mathrm{E}^p(h_k^l)$ and $\mathrm{V}^p(h_k^l)$ calculated in the last iteration should not be used as the *a priori* information $\mathrm{E}(h_k^l)$ and $\mathrm{V}(h_k^l)$ in (22) to estimate h_k^l in the current iteration, since the *a priori* information should be "extrinsic" according to the turbo principle. However, the extrinsic *a priori* information of h_k^l can come from its adjacent segments due to the correlation of time-varying channels. In other words, the estimation results for segments k-1 and k+1 can be used as extrinsic *a priori* information for the estimation of the concerned segment k, as discussed below.

We assume that the channel taps are independent of each other, and use the following first-order autoregressive model to approximately characterize the time-varying channel:

$$h_k^l = \beta_l h_{k-1}^l + w_l \,, \; l = 0, 1, \ldots, L, \tag{23}$$

[2] $\{\mathrm{E}(\boldsymbol{\zeta}_{k,l})$, $\boldsymbol{D}_{k,l}$, $l = 0, \ldots, L\}$ can be updated in parallel based on the results in the last iteration. Alternatively, we can update them in serial to accelerate the algorithm convergence. In this case, the updating of $\{\mathrm{E}(\boldsymbol{\zeta}_{k,l})$, $\boldsymbol{D}_{k,l}$, $l = 0, \ldots, L\}$ should be based on the most updated $\left\{\mathrm{E}^p(h_k^{l'}), \mathrm{V}^p(h_k^{l'})\right\}$ (i.e., some of them are computed in the current iteration). This means that the estimation of $\{h_k^l, l = 0, \ldots, L\}$ based on (22) is also performed in serial.

where β_l is a constant, and w_l denotes white Gaussian process with power p_l. The parameters β_l and p_l can be determined based on the correlation function of the time-varying channel. We can use the following bi-directional algorithm (similar to Kalman smoothing) to exploit correlation information of time-varying channels.

1. *Forward Recursion*

   Assume that $\left\{\mathrm{E}^p_{Fwd}(h^l_{k-1}), \mathrm{V}^p_{Fwd}(h^l_{k-1})\right\}$ are available, where the subscript "*Fwd*" denotes "forward". Set $\mathrm{E}(h^l_k)=\beta_l \mathrm{E}^p_{Fwd}(h^l_{k-1})$ and $\mathrm{V}(h^l_k)=\beta_l^2 \mathrm{V}^p_{Fwd}(h^l_{k-1})+p_l$, and then compute $\left\{\mathrm{E}^p_{Fwd}(h^l_k), \mathrm{V}^p_{Fwd}(h^l_k)\right\}$ using (22).

2. *Backward Recursion*

   Assume that $\left\{\mathrm{E}^p_{Bwd}(h^l_{k+1}), \mathrm{V}^p_{Bwd}(h^l_{k+1})\right\}$ are available, where the subscript "*Bwd*" denotes "backward". Set $\mathrm{E}(h^l_k)=\beta_l^{-1} \mathrm{E}^p_{Bwd}(h^l_{k+1})$ and $\mathrm{V}(h^l_k)=\beta_l^{-2} (\mathrm{V}^p_{Bwd}(h^l_{k+1})+p_l)$, and then compute $\left\{\mathrm{E}^p_{Fwd}(h^l_k), \mathrm{V}^p_{Fwd}(h^l_k)\right\}$ using (22).

3. *Combining the Forward and Backward Information*

   After the forward and backward recursions, the final estimates $\left\{\mathrm{E}^p(h^l_k), \mathrm{V}^p(h^l_k)\right\}$ are calculated as

$$\mathrm{V}^p(h^l_k)=\left(\frac{1}{\mathrm{V}^p_{Fwd}(h^l_k)}+\frac{1}{\beta_l^{-2}(\mathrm{V}^p_{Bwd}(h^l_{k+1})+p_l)}\right)^{-1}, \tag{24a}$$

$$\mathrm{E}^p(h^l_k)=\left(\frac{\mathrm{E}^p_{Fwd}(h^l_k)}{\mathrm{V}^p_{Fwd}(h^l_k)}+\frac{\beta_l^{-1}\mathrm{E}^p_{Bwd}(h^l_{k+1})}{\beta_l^{-2}(\mathrm{V}^p_{Bwd}(h^l_{k+1})+p_l)}\right)\mathrm{V}^p(h^l_k). \tag{24b}$$

   Calculating $\mathrm{E}(h^l_k)$ and $\mathrm{V}(h^l_k)$ in the forward and backward recursions and combining information in (24) are related to the Gaussian message passing technique. Refer to [11] for details.

The complexity of the above described channel estimation algorithm is $O(M)$ per tap.

In the above discussions, we assume that $\{\boldsymbol{x}_k\}$ are exactly known to construct matrices $\{\boldsymbol{A}_k\}$. In practice, only the means $\{\mathrm{E}(\boldsymbol{x}_k)\}$ are available. In this case, we simply use $\mathrm{E}(\boldsymbol{x}_k)$ to form $\boldsymbol{A}_k$.

V. Segment-by-Segment Equalization

We now turn our attention to the equalizer in Fig. 2. We first assume perfect knowledge of $\{\boldsymbol{h}_k\}$ at the receiver.

The equalizer in (12) can be realized in a segment-by-segment manner (see Fig.1). Based on the assumption that the channel is static within a segment, we can efficiently implement the equalizer using FFT. This is in principle equivalent to FDE [1] [13], but the matrix form derivation below is more concise and insightful.

We define the following two vectors with length $N = M+L$:

$$\tilde{\boldsymbol{h}}_k=[\boldsymbol{h}_k^T, \underbrace{0,\ldots,0}_{M-1\ replicas}]^T, \text{ and } \tilde{\boldsymbol{x}}_k=[\boldsymbol{x}_k^T, \underbrace{0,\ldots,0}_{L\ replicas}]^T. \tag{25}$$

Then (16a) can be rewritten as

$$\boldsymbol{r}_k=\tilde{\boldsymbol{h}}_k \otimes \tilde{\boldsymbol{x}}_k+\boldsymbol{n}_k, \tag{26}$$

where "⊗" denotes the cyclic convolution operation. Here, appending zeros to $\boldsymbol{h}_k$ and $\boldsymbol{x}_k$ transforms the linear convolution in (16a) into the cyclic convolution in (26).

Define $\boldsymbol{F}$ as the normalized discrete Fourier transform (DFT) matrix with size $N\times N$, i.e., the (m, n)th element of $\boldsymbol{F}$ is given by $\boldsymbol{F}(m,n)=N^{-1/2}e^{-i2\pi mn/N}$. Hence $\boldsymbol{F}\boldsymbol{F}^H=\boldsymbol{I}$. According to the property of cyclic convolution, we can rewrite (26) as

$$\sqrt{N}\boldsymbol{F}\boldsymbol{r}_k=\sqrt{N}\boldsymbol{F}\tilde{\boldsymbol{h}}_k \bullet \sqrt{N}\boldsymbol{F}\tilde{\boldsymbol{x}}_k+\sqrt{N}\boldsymbol{F}\boldsymbol{n}_k, \tag{27}$$

where "•" denotes element-wise product. Denote the DFT of $\tilde{\boldsymbol{h}}_k$ as:

$$\sqrt{N}\boldsymbol{F}\tilde{\boldsymbol{h}}_k=[g_{k,0}, g_{k,1}, \ldots g_{k,N-1}]^T, \tag{28}$$

and define a diagonal matrix

$$\boldsymbol{G}_k=diag\{g_{k,0}, g_{k,1}, \ldots g_{k,N-1}\}. \tag{29}$$

Then (27) can be rewritten in a matrix form as

$$\boldsymbol{r}_k=\underbrace{\boldsymbol{F}^H\boldsymbol{G}_k\boldsymbol{F}}_{\boldsymbol{A}_k}\tilde{\boldsymbol{x}}_k+\boldsymbol{n}_k. \tag{30}$$

From (16b), we can see that

$$\mathrm{E}(\boldsymbol{n}_k)=\mathrm{E}(\boldsymbol{y}^{Inter}_{k-1})+\mathrm{E}(\boldsymbol{y}^{Inter}_{k+1}), \tag{31a}$$

$$\mathrm{V}(\boldsymbol{n}_k)=\mathrm{V}(\boldsymbol{y}^{Inter}_{k-1})+\mathrm{V}(\boldsymbol{y}^{Inter}_{k+1})+2\sigma^2\boldsymbol{I}. \tag{31b}$$

Define v as the average variance of $\{x_j\}$, and α_k as the average of diagonal elements of matrix $\mathrm{V}(\boldsymbol{n}_k)$. The following two approximations may incur marginal performance loss but lead to considerable cost reduction:

$$\mathrm{V}(\tilde{\boldsymbol{x}}_k)\approx v\boldsymbol{I}, \tag{32a}$$

$$\mathrm{V}(\boldsymbol{n}_k)\approx \alpha_k\boldsymbol{I}, \tag{32b}$$

where α_k can be calculated as

$$\alpha_k=2\sigma^2+v(M+L)^{-1}\sum_{l=0}^{L}(l\,|h^l_{k-1}|^2+(L-l)\,|h^l_{k+1}|^2). \tag{33}$$

Based on (30) and (32), we have

$$\mathrm{V}(\boldsymbol{r}_k)=v\boldsymbol{A}\boldsymbol{A}^H+\mathrm{V}(\boldsymbol{n})=\boldsymbol{F}^H(v\boldsymbol{G}_k\boldsymbol{G}_k^H+\alpha_k\boldsymbol{I})\boldsymbol{F}. \tag{34}$$

Hence

$$\boldsymbol{U}_k=\left(\boldsymbol{A}_k^H\mathrm{V}(\boldsymbol{r})^{-1}\boldsymbol{A}_k\right)_{diag}=u_k\boldsymbol{I}, \tag{35}$$

where

$$u_k=\sum\nolimits_{n=0}^{N-1}N^{-1}|g_{k,n}|^2\left(v\,|g_{k,n}|^2+\alpha_k\right)^{-1}. \tag{36}$$

Based on (34), (35) and (12), we have

$$\begin{aligned} e(\boldsymbol{x}_k)=4(1-vu_k)^{-1}\boldsymbol{S}\Big[&\boldsymbol{F}^H\boldsymbol{G}_k^H(v\boldsymbol{G}_k\boldsymbol{G}_k^H+\alpha_k\boldsymbol{I})^{-1} \\ &(\boldsymbol{z}_k-\boldsymbol{G}_k\boldsymbol{F}\mathrm{E}(\tilde{\boldsymbol{x}}_k)-\boldsymbol{F}\mathrm{E}(\boldsymbol{n}_k))+u_k\mathrm{E}(\tilde{\boldsymbol{x}}_k)\Big] \end{aligned}, \tag{37}$$

where $\boldsymbol{S}=[\boldsymbol{I}_{M\times M}, \boldsymbol{0}]_{M\times N}$, and $\boldsymbol{z}_k=\boldsymbol{F}\boldsymbol{r}_k$.

Remarks:

1. The matrix inversion involved in (37) is trivial because the related matrix is diagonal.
2. The operation related to $\boldsymbol{F}$ and $\boldsymbol{F}^{\boldsymbol{H}}$ can be efficiently realized using FFT. The complexity involved in (37) is only $O(\log_2 N)$ per entry.
3. The term "$-\boldsymbol{F}\mathrm{E}(\boldsymbol{n}_k)$" involved in (37) provides soft cancellation of the interference from the adjacent segments k+1 and k-1. (See (16b) for the definition of $\boldsymbol{n}_k$.) Here, $\mathrm{E}(\boldsymbol{n}_k)=\mathrm{E}(\boldsymbol{y}^{Inter}_{k-1})+\mathrm{E}(\boldsymbol{y}^{Inter}_{k+1})$ can be efficiently computed based on FFT as follows:

$$\begin{aligned}\mathrm{E}(\boldsymbol{y}^{Inter}_{k-1})&=\mathrm{E}([tail(\boldsymbol{h}_{k-1}*\boldsymbol{x}_{k-1}),0,\ldots,0]^T)\\ &=\mathrm{E}([tail(\tilde{\boldsymbol{h}}_{k-1}\otimes\tilde{\boldsymbol{x}}_{k-1}),0,\ldots,0]^T)\\ &=[tail(\boldsymbol{F}^H\boldsymbol{G}_{k-1}\boldsymbol{F}\mathrm{E}(\tilde{\boldsymbol{x}}_{k-1})),0,\ldots,0]^T\end{aligned}.$$

Note that the term $\boldsymbol{G}_k \boldsymbol{F}\mathrm{E}(\tilde{\boldsymbol{x}}_k)$ is involved in (37), which indicates that $\boldsymbol{G}_{k-1}\boldsymbol{F}\mathrm{E}(\tilde{\boldsymbol{x}}_{k-1})$ can be shared by $e(\boldsymbol{x}_{k-1})$ and $e(\boldsymbol{x}_k)$. A similar treatment can be applied to $\mathrm{E}(\boldsymbol{y}_{k+1}^{Inter})$.

The above provides a fast technique to implement the equalizer in (12).

In practice, only the means $\{\mathrm{E}(\boldsymbol{h}_k)\}$ are available. In this case, we simply replace $\boldsymbol{h}_k$ by $\mathrm{E}(\boldsymbol{h}_k)$ in the above discussed algorithm.

VI. Start up the Iterative Process Using Pilot Signal

In the iterative joint process discussed above, channel estimation should be first performed at the receiver. According to the discussions in Section IV, we need set up matrices $\{\boldsymbol{A}_k\}$ based on the mean vectors $\{\mathrm{E}\{\boldsymbol{x}_k\}\}$ that are updated based on the feedbacks from the decoder in the iterative process. However, in the first iteration, there are no decoder feedbacks available, and in general we don't have *a priori* information about them, which means $\{\mathrm{E}(\boldsymbol{x}_k) = \boldsymbol{0}\}$. This causes difficulty in evaluating (22). Pilot signal can be used to solve this problem.

Fig. 3 shows the placement of the pilot signal. The pilot signal is superposed with the data signal, which only incurs power loss (without rate loss).

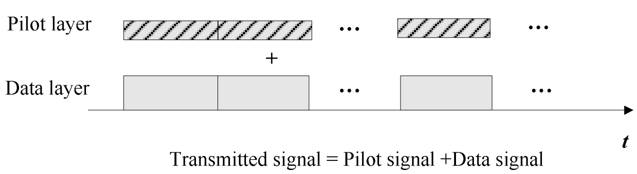

Fig. 3. Placement of pilot signal. The pilot signal only involves power loss (without spectral loss).

The transmitted signal can be represented as

$$\boldsymbol{x} = \boldsymbol{p} + \boldsymbol{c}, \tag{38}$$

where $\boldsymbol{p}$ denotes the pilot signal and $\boldsymbol{c}$ denotes the data signal. In the first iteration,

$$\mathrm{E}(\boldsymbol{x}) = \mathrm{E}(\boldsymbol{p}) + \mathrm{E}(\boldsymbol{c}) = \boldsymbol{p}. \tag{39}$$

The iterative process can be started here. Note that, in equalization, some extra operations should be included to handle the contribution of the pilot signal.

Now we conclude the overall iterative process as follows:

Step 1. Based on the feedbacks from the decoder, the channel estimator performs channel estimation via forward and backward recursions based on (22) and (24).

Step 2. Based on the channel estimate information from the channel estimator and the feedbacks from the decoder, the channel equalizer resolves ISI and inter-segment interference to provide data estimates based on (37).

Step 3. The decoder refines the data estimates, and the results will be used by the channel estimator and equalizer in the next iteration. Then go to Step 1.

Hard decisions on the information bits are made by the decoder during the final iteration.

VII. Simulation Results

We first examine the proposed segment-by-segment equalization algorithm in quasi-static ISI channels under the assumption of perfect CSI at receiver side. This example is to compare the performance of the proposed method with the known performance limit.

Example 1: Quasi-static channels with perfect CSI

In this example, the encoding scheme is a rate-1/2 convolutional code with generator $(23, 35)_8$, and the information length is 4096. QPSK modulation is used. Hence the length of block (i.e., length of $\boldsymbol{x}$) is also 4096. The segment length (i.e., the length of $\boldsymbol{x}_k$) is set to be 64. The number of iterations is 10. The length of ISI channels is 17. The 17 coefficients remain constant for all the segments in a block, and they are independently drawn from a complex Gaussian distribution with mean 0 and variance 1 for different blocks. In each channel realization, the channel energy is normalized to 1.

We know that the performance of the system over such ISI channels is bounded by the performance of the code over an AWGN channel. The performance is shown in Fig. 4, from which we can clearly see that the proposed equalization algorithm can almost achieve the ISI-free performance at relatively high E_b/N_o, which also implies that the inter-segment interference is almost eliminated.

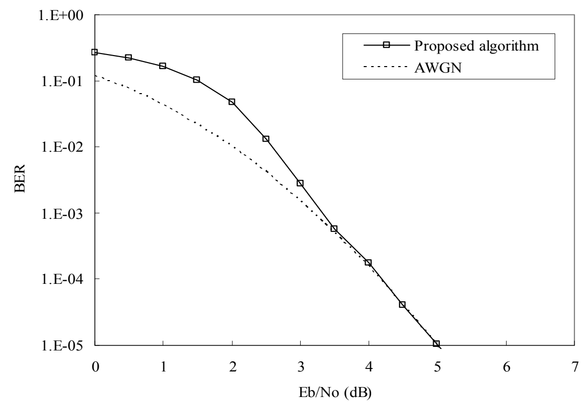

Fig. 4. Performance of the proposed algorithm in 17-tap quasi-static ISI channels with energy 1. Segment length is 64 and the number of iterations is 10.

Next we examine the proposed joint channel estimation, equalization and decoding scheme over a doubly selective channel.

Example 2: Doubly selective channels with estimated CSI

We adopt the basis expansion model (BEM) detailed in [12]. In this model, the channel coefficients are generated by

$$h_j^l = \sum_{q=0}^{Q} \lambda_{l,q} e^{i\omega_q j}, \; l = 0, 1, \ldots L, \tag{40}$$

where $\omega_q = 2\pi(q-Q/2)/N$, $Q = \lceil f_{max}T \rceil$, f_{max} represents Doppler spread, and T is the time duration of a block. Define T_s and N_s as the symbol duration and the number of symbols in one block, then $T = N_s T_s$. The BEM coefficients $\lambda_{l,q}$ is a zero-mean, complex Gaussian random variable with variance $\sigma^2_{l,q}$. We set carrier frequency f_0 = 2 GHz, sampling period T_s = 10 μs, and mobile speed v = 140 km/hr. The corresponding Doppler spread f_{max} = 259Hz (i.e., the normalized Doppler spread $f_{max}T_s$ = 0.00259). The Doppler power spectrum is chosen as

$$S_c(f) = \begin{cases} \left(\pi\sqrt{f_{max}^2 - f^2}\right)^{-1}, & f \le f_{max} \\ 0, & f > f_{max} \end{cases} \tag{41}$$

The number of multi-paths is 9, and the multipath intensity profile is selected as $p(\tau) = \exp(-0.1\tau / T_s)$. The variance

$\lambda_{l,q} = \alpha^{-1} p(lT_s) S_c(q/(NT_s))$ with $\alpha = \sum_{l',q'} p(l'T_s) S_c(q'/(NT_s))$. The coding scheme is a rate-1/2 convolutional code with generator $(23, 35)_8$. QPSK modulation is used. The information length is 4096, and so is the block length.

We first examine the performance of the proposed scheme with CSI available at the receiver. In this case, our focus is the effect of different segment lengths on system performance. We assume that CSI corresponding to the middle point of each block is exactly known at the receiver. Fig. 5 shows the system performance when the segment length M is 16, 32, 64, and 128, respectively. We can clearly observe that the performance degrades with the increase of segment length. (Performance is relatively poor at $M = 64$ compared with $M = 32$ and 16. A high error floor occurs at $M = 128$.) This is because the algorithm assumes a static ISI channel within a segment. This assumption is invalid when $M = 64$ and 128. We can also see that the performance with $M = 16$ and 32 is almost the same, which indicates that the static channel assumption is valid when $M \leq 32$.

If the conventional FDE is used with $M = 32$, the use of CP will incur extra power loss of 1 dB and spectral loss of 20%.

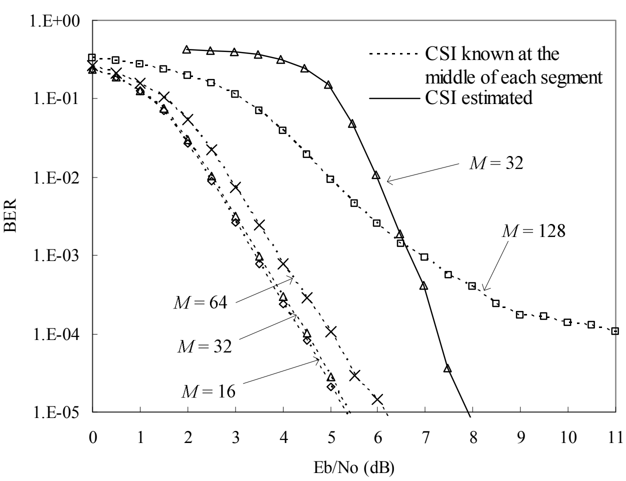

Fig. 5. Performance of the proposed approach over doubly selective channels. The power loss due to the use of pilot is included in Eb/No. The normalized Doppler spread $f_{max}T_s = 0.00259$. The number of iterations is 10.

We next examine the performance of the proposed scheme with estimated CSI. We set $M = 32$ at which the channel can be regarded as approximately static within a segment. The pilot segments are superposed with the data segments. The power ratio of the pilot signal and data signal is 1/4. Hence the power loss due to pilot signal is $10\log_{10}(5/4) \approx 1$dB. From Fig. 5, we can see that the performance gap between the scheme with CSI and estimated CSI is less than 3dB at the BER of 1.0×10^{-5}. Here the power loss of about 1dB due to pilot signal is included in Eb/No.

VIII. CONCLUSIONS

We have proposed a low-complexity iterative joint channel estimation, equalization and decoding scheme for doubly selective channels. The key is a segment-by-segment processing technique. A bi-directional channel estimation algorithm has been developed to exploit correlation of time-varying channels. The proposed channel equalizer inherits the low-complexity advantage of FDE technique, but does not resort to cyclic prefixing and so avoids the related power and spectral overheads. Simulation results demonstrate the efficiency of the proposed algorithms.

ACKNOWLEDGEMENT

This work was performed in the framework of the ICT project ICT-217033 WHERE partly funded by the European Union.

REFERENCES

[1] M. Tüchler, and J. Hagenauer, "Turbo equalization using frequency domain equalizers," in *Proc. Of Allerton Conference*, Monticello, IL, USA, Oct. 2000.

[2] D. Kim and G. Stüber, "Residual ISI cancellation for OFDM with applications to HDTV broadcasting," *IEEE. J. Sel. Areas Commun*., vol. 16, no. 8, pp.1590-1599, Oct. 1998.

[3] Y. Li, S. McLaughlin and D.G.M. Cruickshank, "Bandwidth efficient single carrier systems with frequency domain equalization," *Electron. Lett*, vol. 41, no. 15, pp. 857-858, Jul. 2005.

[4] A. Gusmão, P. Torres, and R. Dinis, "A turbo FDE technique for reduced-CP SC-based block transmission systems," *IEEE Trans. Commun.*, vol.55, pp. 16-20, Jan. 2007.

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

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

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

[8] H. Schoeneich and P. A. Hoeher, "Iterative pilot-layer aided channel estimation with emphasis on interleave-division multiple access systems," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp.1-15, 2006.

[9] K. Ishihara, K. Takeda, and F. Adachi, "Iterative channel estimation for frequency-domain equalization of DSSS signals," *IEICE Trans. Commun*., vol.E90-B, pp.1171-1180, May 2007.

[10] Steven M. Kay, *Fundamentals of Statistical Signal Processing*, Prentice-Hall PTR, 1993.

[11] H. -A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, pp. 28-41, Jan. 2004.

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

[13] Q. Guo, X. Yuan and Li Ping, "Multi-user detection techniques for potential 3GPP long term evolution (LTE) schemes," *Multi-Carrier Spread Spectrum 2007, Lecture Notes Electrical Engineering*, vol. 1, pp. 77-86, Jun. 2007.

[14] L. Liu, W. K. Leung, Li Ping, "Simple chip-by-chip multiuser detection for CDMA systems," *IEEE VTC*'03, pp.2157-2161.

[15] L.Bahl, J.Cocke, F.Jelinek, and J.Raviv, "Optimal Decoding of Linear Codes for minimizing symbol error rate", *IEEE Trans. Inform. Theory*, vol. 20, pp.284-287, March 1974.