Iterative Frequency-Domain Equalization of Single-Carrier Transmissions over Doubly Dispersive Channels

Philip Schniter* and Hong Liu

Abstract

Frequency-domain equalization (FDE) offers an attractive alternative to time-domain equalization in systems that communicate over large-delay-spread channels. Traditionally, FDE leverages the fact that time-domain convolution is equivalent to frequency-domain multiplication and the fact that time/frequency conversion is efficiently handled by the fast Fourier transform (FFT). In doubly dispersive channels, i.e., quickly varying large-delay-spread channels, the traditional FDE methods fail when the channel response varies significantly over the FFT analysis window. Here we present a new FDE that is based on Doppler channel shortening, soft iterative interference cancellation, and block decision feedback. Numerical simulations show that the proposed technique has advantages over the well-known FIR-MMSE-DFE in both performance and complexity.

I. Introduction

In systems that communicate over large-delay-spread channels, the use of time-domain equalization (TDE) leads to expensive receivers. For example, North American terrestrial digital television is plagued by delay spreads on the order of hundreds of symbol intervals, requiring time-domain equalizers with hundreds of coefficients [1]. Frequency-domain equalization (FDE) offers an attractive alternative. FDE leverages the facts that circular convolution in the time domain can be accomplished by pointwise multiplication in the frequency domain, and

The authors are with the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, OH.

Please direct all correspondence to Prof. Philip Schniter, Dept. ECE, 2015 Neil Ave., Columbus OH 43210, e-mail: schniter.1@osu.edu, phone 614.247.6488, fax 614.292.7596. Hong Liu can be reached at the same address/phone/fax and the e-mail liuh@ece.osu.edu.

Portions of this work were presented at the 2003 and 2004 Asilomar Conferences on Signals, Systems, and Computers.

This work was supported by NSF CAREER CCR-0237037.

that transformation to/from the frequency domain can be efficiently accomplished using the fast Fourier transform (FFT). Roughly speaking, the per-sample processing complexity required for TDE is linear in the channel delay spread while for FDE it is logarithmic in the delay spread. Thus, FDE can lead to significant savings over TDE for long channels.

FDE is the principle idea behind orthogonal frequency division multiplexing (OFDM) [2], [3] and single-carrier cyclic-prefix (SCCP) [4], [5] modulation schemes. Both OFDM and SCCP systems transmit blocks of data separated by guard intervals. The guard prevents inter-block interference, thereby simplifying receiver processing. The use of a cyclic-prefix (CP) guard makes the channel's dispersion act as a cyclic (rather than linear) convolution, implying that deconvolution can be accomplished through a simple FFT-domain multiplication. When guard intervals are *not* included, FDE can still be accomplished using overlap-add/save FFT algorithms (see, e.g., [6], [7]) or residual ISI cancellation (RISIC) [8].

The previously mentioned FDE techniques assume a delay-spread channel whose impulse response varies slowly. Some applications, however, have channels with more significant time variation, i.e., significant Doppler *and* delay spreads. For such doubly dispersive channels, the standard approach to FDE (i.e., pointwise multiplication in the frequency-domain) fails when the channel varies significantly over the FFT block duration. Essentially, the channel variation induces inter-carrier interference (ICI) in the frequency domain. In response, several equalization schemes for doubly dispersed CP-OFDM have been proposed (see, e.g., [9]–[13] and the references within). While most of these schemes are computationally intensive, [12] maintains per-symbol processing complexity logarithmic in the block length, in keeping with the spirit of FDE. In addition, [12] exploits the finite-alphabet property of frequency-domain symbols, allowing its performance to surpass that of minimum mean-squared error (MMSE) linear equalization. The CP-OFDM FDE scheme [12] was extended to SCCP in [14]. Though the SCCP FDE scheme [14] is complicated by the fact that the finite-alphabet property resides in the time domain, it nevertheless maintains the desired logarithmic per-symbol processing complexity.

Though capable of FDE on doubly dispersive channels, the algorithms [9]–[12] and [14] require block-based transmissions with an adequate inter-block guard interval. While [13] does not require a guard, its complexity scaling properties restrict its application to channels with mild spreading. Thus, one might wonder: Is it possible to build a FDE algorithm for single-carrier *continuous-stream* modulation over doubly dispersive channels that exhibits logarithmic complexity scaling? If so, such an algorithm would present an efficient frequency-domain alternative to the time-domain equalization approaches that are commonly used in doubly selective single-carrier receivers (e.g.,

for North American terrestrial digital television [15] and underwater acoustic communication [16]).

In this paper we present an iterative frequency-domain equalizer (IFDE) for a continuous finite-alphabet stream corrupted by a noisy and doubly dispersive channel. In brief, the algorithm first parses the received time-domain signals into blocks which are first windowed and then transformed into the frequency domain by an FFT. The window is designed so that both channel variations and the lack-of-CP manifest as a sparse frequency-domain ICI response. A low-complexity serial technique is then applied to equalize the channel response in the presence of ICI, and the output is transformed back to the time domain, yielding soft symbol estimates. Using the finite-alphabet symbol property, reliability information on these soft estimates is computed for use in another round of (frequency-domain) equalization and ICI cancellation. The time- and frequency-domain steps are alternated until the soft symbols estimates converge.

Through simulation, our IFDE algorithm's performance is compared to that of the FIR-MMSE-DFE [17], a well-known benchmark, as well as to that of the matched filter bound (MFB) [18]—the "holy grail" of uncoded equalization. We find that our IFDE performs 1 dB worse than the MFB, and several dB better than the FIR-MMSE-DFE, over the SNR range of interest. In addition, we analyze the number of multiplications required by our IFDE and compare it to that of the FIR-MMSE-DFE updated using a fast algorithm. We find that our IFDE has complexity advantages over the FIR-MMSE-DFE for channels of a reasonable length.

In our equalizer design, we treat the channel as perfectly known. In practical terms, this means that a well-designed channel estimation algorithm is assumed to be operating in tandem with the equalizer. While channel estimation is an interesting and important topic, it is outside the scope of this manuscript. In fact, the decoupling of equalization from channel identification is typical of work that studies non-trivial equalizer structures, e.g., [9]–[13], [17], [19], [20].

The paper is organized as follows. Section II gives the system model (in time- and frequency- domains), Section III and Sec. IV describe our IFDE scheme, and Sec. V presents our fast IFDE implementation. Section VI reports the results of numerical studies, and Sec. VII concludes. We use the following notation throughout. Transpose is denoted by $(\cdot)^t$, conjugate by $(\cdot)^*$, and conjugate transpose by $(\cdot)^H$. The zero matrix is denoted by $\mathbf{0}$, the identity matrix by $\boldsymbol{I}$, and the k^{th} column of the the identity matrix by $\boldsymbol{i}_k$. The element in the m^{th} row and n^{th} column of matrix $\boldsymbol{B}$ is denoted by $[\boldsymbol{B}]_{m,n}$, where row/column indices begin with zero. The diagonal matrix created from vector $\boldsymbol{b}$ is denoted by $\mathcal{D}(\boldsymbol{b})$, and the circulant matrix with first column $\boldsymbol{b}$ by $\mathcal{C}(\boldsymbol{b})$. The $N \times 1$ vector created from the i^{th} sub-diagonal of $N \times N$ matrix $\boldsymbol{B}$ is denoted by $\mathrm{diag}_i(\boldsymbol{B})$, i.e., $[\mathrm{diag}_i(\boldsymbol{B})]_k = [\boldsymbol{B}]_{\langle k+i \rangle_N, k}$ for $k \in \{0, 1, \ldots, N-1\}$.

Expectation is denoted by $\mathrm{E}\{\cdot\}$ and covariance by $\mathrm{Cov}\{\boldsymbol{b},\boldsymbol{c}\} := \mathrm{E}\{\boldsymbol{b}\boldsymbol{c}^H\} - \mathrm{E}\{\boldsymbol{b}\}\,\mathrm{E}\{\boldsymbol{c}^H\}$. Finally, the Kronecker delta is denoted by δ_m, the modulo-N operation by $\langle\cdot\rangle_N$, element-wise matrix multiplication by $\odot$, and the set of integers by $\mathbb{Z}$.

II. System Model

Consider a single-carrier modulation system where a stream of finite-alphabet symbols $\{s_n\}$ is transmitted over a noisy linear time-varying (LTV) multipath channel. The channel is described by its time-variant discrete impulse response $h_{n,l}$, defined as the time-n response to an impulse applied at time $n-l$. We assume a causal impulse response of length N_h. The signal observed by the receiver is

$$r_n \;=\; \nu_n + \sum_{l=0}^{N_h-1} h_{n,l}\, s_{n-l} \tag{1}$$

where ν_n denotes samples of zero-mean circular white Gaussian noise (CWGN) with variance σ^2. We assume wide-sense stationary uncorrelated scattering (WSSUS) [18] so that $\mathrm{E}\{h_{n,l}h^*_{n-q,l-m}\} = \gamma_q\sigma_l^2\delta_m$. Here, γ_q denotes the normalized autocorrelation (i.e., $\gamma_0 = 1$) and σ_l^2 the variance of the channel at delay l^{th}.

The remainder of this section establishes the block-based frequency-domain equivalent of (1). At each frame index $i \in \mathbb{Z}$, the receiver windows an iN-shifted version of the time-domain observation $\{r_n\}$ and applies a discrete Fourier transform (DFT) with frequency spacing $\frac{2\pi}{PN}$, yielding the i^{th}-frame frequency domain observation $\{x_d(i)\}_{d=0}^{PN-1}$:

$$x_d(i) \;=\; \frac{1}{\sqrt{PN}} \sum_n r_{iN+n} b_n e^{-j\frac{2\pi}{PN}dn}. \tag{2}$$

Note that the window length is arbitrary. For convenience, we define

$$s_n(i) \;:=\; s_{iN+n}, \quad n \in \{0,\ldots,PN-1\} \tag{3}$$

$$a_n \;:=\; \begin{cases} 1 & n \in \{0,\ldots,PN-1\}, \\ 0 & \text{else}, \end{cases}$$

noting that $\{a_n\}$ specifies a PN-length rectangular window and that

$$s_{iN+n} \;=\; \sum_{\ell=-\infty}^{\infty} s_{\langle n\rangle_{PN}}(i-P\ell)\, a_{\ell PN+n}. \tag{4}$$

Equation (4) says that, for a particular i, the transmitted sequence $\{s_{iN+n}\}$ can be constructed using PN-sample shifts of the disjoint subsequences $\{s_n(i-P\ell)\}_{n=0}^{PN-1}$ for $\ell \in \mathbb{Z}$. Combining (1)-(4), we find

$$x_d(i) \;=\; w_d(i) + \frac{1}{\sqrt{PN}} \sum_n b_n \sum_{l=0}^{N_h-1} h_{iN+n,l} \sum_{\ell=-\infty}^{\infty} s_{\langle n-l \rangle_{PN}}(i-P\ell) a_{\ell PN+n-l} e^{-j\frac{2\pi}{PN}nd} \tag{5}$$

$$w_d(i) \;:=\; \frac{1}{\sqrt{PN}} \sum_n b_n \nu_{iN+n} e^{-j\frac{2\pi}{PN}nd}. \tag{6}$$

Frequency-domain equalization involves the i^{th}-frame *virtual subcarriers* $\{t_k(i)\}_{k=0}^{PN-1}$, where

$$t_k(i) \;:=\; \frac{1}{\sqrt{PN}} \sum_{n=0}^{PN-1} s_n(i) e^{-j\frac{2\pi}{PN}nk}. \tag{7}$$

Equation (7) implies that $s_n(i) = \frac{1}{\sqrt{PN}} \sum_{k=0}^{PN-1} t_k(i) e^{j\frac{2\pi}{PN}nk}$ for $n \in \{0,\ldots,PN-1\}$. Using this in (5) gives

$$x_d(i) \;=\; w_d(i) + \sum_{\ell=-\infty}^{\infty} \sum_{k=0}^{PN-1} t_k(i-P\ell) H_{d-k,k}(i,\ell) \tag{8}$$

$$H_{d,k}(i,\ell) \;:=\; \frac{1}{PN} \sum_n \sum_{l=0}^{N_h-1} h_{iN+n,l} b_n a_{\ell PN+n-l}\, e^{-j\frac{2\pi}{PN}(kl+nd)}. \tag{9}$$

Equation (8) indicates that $H_{d,k}(i,\ell)$ can be interpreted as the response, at DFT output $k+d$ in frame i, to a frequency-domain impulse applied at virtual subcarrier k in frame $i-\ell$.

In practice we implement a causal length-N_b window $\{b_n\}$ implying that, for any i, only a finite number of terms in the set $\{H_{d,k}(i,\ell), \ell \in \mathbb{Z}\}$ will be non-zero. Specifically, (9) implies that non-zero terms result from indices ℓ which satisfy $0 \le \ell NP + n - l \le PN-1$ for some $n \in \{0,\ldots,N_b-1\}$ and some $l \in \{0,\ldots,N_h-1\}$. It is straightforward to show that $H_{d,k}(i,\ell)$ is non-zero for $\ell \in \{-L_{\mathsf{pre}},\ldots,L_{\mathsf{pst}}\}$ where $L_{\mathsf{pre}} = -\lfloor \frac{N_b-1}{PN} \rfloor$ and $L_{\mathsf{pst}} = \lfloor \frac{PN+N_h-2}{PN} \rfloor$.

With the definitions $\boldsymbol{x}(i) := [x_0(i),\ldots,x_{PN-1}(i)]^t$, $\boldsymbol{w}(i) := [w_0(i),\ldots,w_{PN-1}(i)]^t$, $\boldsymbol{t}(i) := [t_0(i),\ldots,t_{PN-1}(i)]^t$, $\boldsymbol{s}(i) := [s_0(i),\ldots,s_{PN-1}(i)]^t$, and $[\boldsymbol{\mathcal{H}}(i,\ell)]_{d,k} := H_{d-k,k}(i,\ell)$, (8) implies the LTV multiple-input multiple-output (MIMO) system

$$\boldsymbol{x}(i) \;=\; \boldsymbol{w}(i) + \sum_{\ell=-L_{\mathsf{pre}}}^{L_{\mathsf{pst}}} \boldsymbol{\mathcal{H}}(i,\ell) \boldsymbol{t}(i-\ell P). \tag{10}$$

For any i, nonzero $\{\boldsymbol{\mathcal{H}}(i,\ell)\}_{\ell\neq 0}$ cause inter-frame interference (IFI) and nonzero off-diagonal elements of $\{\boldsymbol{\mathcal{H}}(i,0)\}$ cause inter-carrier interference (ICI) among the virtual subcarriers. In the sequel, we refer to $\{\boldsymbol{\mathcal{H}}(i,\ell)\}_{\ell<0}$ as pre-cursor IFI and $\{\boldsymbol{\mathcal{H}}(i,\ell)\}_{\ell>0}$ as post-cursor IFI.

It will sometimes be convenient to write the windowed frequency-domain noise vector $\boldsymbol{w}(i)$ as

$$\boldsymbol{w}(i) \;=\; \underbrace{\boldsymbol{F}\boldsymbol{J}\,\mathcal{D}(\boldsymbol{b})}_{:=\boldsymbol{C}}\boldsymbol{\nu}(i) \tag{11}$$

$$\boldsymbol{J} \;:=\; \left[\begin{array}{ccc|c} \boldsymbol{I}_{PN} & \cdots & \boldsymbol{I}_{PN} & \boldsymbol{I}_{\bar{N}_o} \\ & & & \boldsymbol{0}_{PN-\bar{N}_o\times\bar{N}_o} \end{array}\right] \tag{12}$$

where $\boldsymbol{F}$ denotes the PN-point unitary DFT matrix, $\bar{N}_o = \langle N_b\rangle_{PN}$, and the number of $\boldsymbol{I}_{PN}$ matrices in $\boldsymbol{J}$ is $\lfloor\frac{N_b}{PN}\rfloor$.

III. Max-SINR Window Design

The choice of window $\{b_n\}$ affects the IFI/ICI patterns of the MIMO system (10). Motivated by the low-pass nature of typical Doppler spectra, we aim to find $\{b_n\}$ such that the "cursor" coefficient $\boldsymbol{\mathcal{H}}(i,0)$ has the banded structure illustrated in Fig. 1 and the IFI coefficients $\{\boldsymbol{\mathcal{H}}(i,\ell)\}_{\ell\neq 0}$ vanish. This approach can be viewed as the frequency-domain dual of inter-symbol interference (ISI) response shortening used to reduce the complexity of maximum likelihood sequence detection (MLSD) [21]. For our purposes, the goal of time-domain windowing is to give the channel a sparse structure that leads to low-complexity estimation of $\boldsymbol{t}(i)$, and hence, low-complexity detection of $\boldsymbol{s}(i)$. We choose time-domain windowing, rather than a general matrix operation on the received signal, due to its low complexity. Since complete cancellation of out-of-target ICI/IFI is, in general, not possible with time-domain windowing, we choose to maximize signal to interference-plus-noise ratio (SINR) as a means of suppressing residual IFI/ICI.

We define SINR by $\mathcal{E}_{\mathsf{s}}/\mathcal{E}_{\mathsf{ni}}$, where $\mathcal{E}_{\mathsf{s}} := \sum_d \mathcal{E}_{\mathsf{s},d}$ and $\mathcal{E}_{\mathsf{ni}} := \sum_d \mathcal{E}_{\mathsf{ni},d}$. For each $x_d(i)$, $\mathcal{E}_{\mathsf{s},d}$ is defined as the signal energy contributed by neighboring carriers $\{t_k(i)\}_{k=d-D}^{d+D}$, and $\mathcal{E}_{\mathsf{ni},d}$ is defined as the interference-plus-noise energy contributed by non-neighboring carriers $\{t_k(i)\}_{k=0}^{d-D-1}\cup\{t_k(i)\}_{k=d+D+1}^{PN-1}$, non-cursor carriers $\{t_k(j)\}_{j\neq i}$, and additive noise $\boldsymbol{w}(i)$. Note that indices here are taken modulo-PN. The ICI radius D is typically chosen as $D = \lceil f_{\mathsf{d}}T_{\mathsf{s}}PN\rceil$, where $f_{\mathsf{d}}T_{\mathsf{s}}$ is the maximum Doppler frequency normalized to the symbol rate. Using the approach outlined in [22], we find that the SINR-maximizing window $\boldsymbol{b}_\star$ is given by

$$\begin{aligned}\boldsymbol{b}_\star \;&=\; \underset{\boldsymbol{b}:\|\boldsymbol{b}\|^2=PN}{\arg\max}\,\frac{\boldsymbol{b}^H\big(\boldsymbol{R}_b\odot\boldsymbol{D}_b\odot\boldsymbol{A}_{\mathsf{s}}\big)\boldsymbol{b}}{\boldsymbol{b}^H\big(\sigma^2\boldsymbol{I}+\boldsymbol{R}_b\odot\boldsymbol{C}_b\odot\boldsymbol{A}_{\mathsf{t}}-\boldsymbol{R}_b\odot\boldsymbol{D}_b\odot\boldsymbol{A}_{\mathsf{s}}\big)\boldsymbol{b}} \\ &=\; \boldsymbol{v}_\star\big(\boldsymbol{R}_b\odot\boldsymbol{D}_b\odot\boldsymbol{A}_{\mathsf{s}},\;\sigma^2\boldsymbol{I}+\boldsymbol{R}_b\odot\boldsymbol{C}_b\odot\boldsymbol{A}_{\mathsf{t}}-\boldsymbol{R}_b\odot\boldsymbol{D}_b\odot\boldsymbol{A}_{\mathsf{s}}\big)\cdot\sqrt{PN}\end{aligned} \tag{13}$$

where $\boldsymbol{R}_b$, $\boldsymbol{A}_{\mathsf{s}}$, $\boldsymbol{C}_b$, $\boldsymbol{D}_b$ and $\boldsymbol{A}_{\mathsf{t}}$ are $N_b\times N_b$ matrices defined element-wise as $[\boldsymbol{R}_b]_{m,n} := \gamma_{n-m}$, $[\boldsymbol{A}_{\mathsf{s}}]_{m,n} := \sum_{l=0}^{N_h-1}\sigma_l^2 a_{n-l}a_{m-l}^*$, $[\boldsymbol{C}_b]_{m,n} := \delta_{\langle n-m\rangle_{PN}}$, $[\boldsymbol{D}_b]_{m,n} := \frac{1}{PN}\sin(\frac{\pi}{PN}(2D+1)(n-m))/\sin(\frac{\pi}{PN}(n-m))$ and

$[\boldsymbol{A}_{\mathsf{t}}]_{m,n} := \sum_{\ell=-L_{\mathsf{pre}}}^{L_{\mathsf{pst}}} \sum_{l=0}^{N_h-1} \sigma_l^2 a_{\ell PN+n-l} a_{\ell PN+m-l}^*$. In (13), $\boldsymbol{v}_\star(\boldsymbol{B},\boldsymbol{C})$ denotes the principle generalized eigenvector [23] of the matrix pair $(\boldsymbol{B},\boldsymbol{C})$. Through max-SINR windowing and proper selection of other design parameters, the IFI and non-neighboring ICI can be made small enough to base the symbol detection procedure on the following approximate system model.

$$\boldsymbol{x}(i) \;\approx\; \boldsymbol{\mathcal{H}}(i,0)\boldsymbol{t}(i) + \boldsymbol{C}\boldsymbol{\nu}(i). \tag{14}$$

As an alternative, the design parameters (e.g., frame length PN) could be chosen to yield non-negligible post-cursor IFI, which could then be canceled using block decision-feedback equalization (BDFE). In this case, the window should be designed to suppress *only* ICI and pre-cursor IFI, implying $[\boldsymbol{A}_{\mathsf{t}}]_{m,n} := \sum_{\ell=-L_{\mathsf{pre}}}^{0} \sum_{l=0}^{N_h-1} \sigma_l^2 a_{\ell PN+n-l} a_{\ell PN+m-l}^*$. Figure 2 plots windows for both BDFE and non-BDFE cases at $f_{\mathsf{d}}T_{\mathsf{s}} \in \{0.001, 0.0075\}$. In generating Fig. 2, we assumed $N_h = 64$, $PN = 256$, $N_b = PN + N_h - 1$, SNR$=$ 10dB, and $\sigma_l^2 = N_h^{-1}$, which are typical choices for the numerical results in Sec. VI. Assuming reliable post-cursor IFI cancellation, (14) would still describe the model used for detection of symbols in the current frame.

While windowing gives a sparse channel response that enables a reduced complexity symbol detection procedure, it can lead to a non-uniform collection of energy from symbols in the current frame. Specifically, it can be shown that the energy in $\boldsymbol{x}(i)$ contributed by $s_n(i)$ is

$$\begin{aligned}\mathcal{E}_{\mathsf{ss},n} \;&:=\; \sum_{d=0}^{PN-1} \mathrm{E}\left\{ \left| \sum_{k=0}^{PN-1} H_{d-k,k}(i,0) \cdot \frac{1}{\sqrt{PN}} e^{-j\frac{2\pi}{PN}nk} s_n(i) \right|^2 \right\} \\ &=\; \sum_{l=0}^{N_h-1} \sigma_l^2 |b_{l+n}|^2 \end{aligned} \tag{15}$$

which is clearly dependent on n, the symbol position within the frame. This implies that, for typical max-SINR window shapes, we will collect less energy from symbols near the frame edges. This phenomenon motivates the frame-overlapping procedure proposed in Sec. IV.

IV. Symbol Detection

In Sec. IV-A, we propose an iterative method for the detection of the finite-alphabet symbol vector $\boldsymbol{s}(i) = \boldsymbol{F}^H \boldsymbol{t}(i)$ assuming the observation model (14). We are careful to leverage the banded structure of $\boldsymbol{\mathcal{H}}(i,0)$ and the existence of fast algorithms for the transformation $\boldsymbol{F}$. It was previously observed that the max-SINR windowing described in Sec. III collects less energy from symbols near the frame edges, which, if unaccounted for, could lead to high frame-averaged error rates. Hence, Sec. IV-B proposes a scheme whereby frame overlap (i.e., $P > 1$) is exploited, in conjunction with the algorithm of Sec. IV-A, to circumvent this problem.

A. Intraframe Processing

Here we propose an iterative method for the detection of the finite-alphabet symbol vector $\boldsymbol{s}(i)$ from the windowed frequency-domain observation $\boldsymbol{x}(i)$ in (14). Note that the focus of this section is *intraframe* processing, whereas the focus of Sec. IV-B is *interframe* processing. Since here we focus exclusively on the i^{th} symbol and on the cursor IFI coefficient, we can omit the symbol and lag indices, abbreviating, e.g., $\boldsymbol{s}(i)$ by $\boldsymbol{s}$ and $\boldsymbol{\mathcal{H}}(i,0)$ by $\boldsymbol{\mathcal{H}}$. We now give a brief summary of the intraframe detection algorithm illustrated in Fig. 3; a more detailed description will be given in Sections IV-A.1–IV-A.3.

Given current guesses of the log-likelihood ratios (LLRs) of the symbols $\{s_k\}$ (which, on the first iteration, are set to zero), the means and variances of the elements in $\boldsymbol{s}$ are calculated as $\bar{\boldsymbol{s}}$ and $\boldsymbol{v}$, respectively. These are then transformed into the mean and covariance of $\boldsymbol{t}$. Using linear MMSE estimation and incorporating these mean/variance priors, the elements $\{t_k\}$ are estimated one-at-a-time, leveraging the banded structure of $\boldsymbol{\mathcal{H}}$ for complexity reduction. The resulting estimates $\hat{\boldsymbol{t}}$ are then transformed back into the $\boldsymbol{s}$-domain, from which the LLRs are updated. To accomplish this last step we assume a conditionally-Gaussian model for the estimates $\{\hat{s}_k\}$. The procedure then repeats, starting with the most recent LLRs. In the detailed description below, we use the superscript $^{(n)}$ to denote the n^{th} iteration.

1) Linear Estimation with Priors: The banded structure of $\boldsymbol{\mathcal{H}}$ suggests that linear estimation of a particular element t_k might be accomplished with reasonable accuracy from the truncated observation $\boldsymbol{x}_k := [x_{k-D}, \ldots, x_{k+D}]^t$, with indices taken modulo-PN, as opposed to the full observation $\boldsymbol{x}$. (See Fig. 5.) We hope to realize substantial complexity reduction in doing so. The truncated model becomes

$$\boldsymbol{x}_k \;=\; \boldsymbol{\mathcal{H}}_k \boldsymbol{t} + \boldsymbol{C}_k \boldsymbol{\nu}, \tag{16}$$

where $\boldsymbol{\mathcal{H}}_k$ contains rows $\{k-D, \ldots, k+D\}$ of $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{C}_k$ contains rows $\{k-D, \ldots, k+D\}$ of $\boldsymbol{C}$. The MMSE linear estimate of t_k given $\boldsymbol{x}_k$ is [24]

$$\hat{t}_k \;=\; \mathrm{E}\{t_k\} + \mathrm{Cov}(t_k, \boldsymbol{x}_k)\,\mathrm{Cov}(\boldsymbol{x}_k, \boldsymbol{x}_k)^{-1}(\boldsymbol{x}_k - \mathrm{E}\{\boldsymbol{x}_k\}). \tag{17}$$

We assume $\mathrm{E}\{\boldsymbol{\nu}\} = \boldsymbol{0}$, $\mathrm{Cov}(\boldsymbol{\nu}, \boldsymbol{\nu}) = \sigma^2 \boldsymbol{I}$, and $\mathrm{Cov}(\boldsymbol{s}, \boldsymbol{\nu}) = \boldsymbol{0}$, and we model the elements in $\boldsymbol{s}$ as uncorrelated with means $\bar{\boldsymbol{s}}^{(n)}$ and variances $\boldsymbol{v}^{(n)}$ during the n^{th} iteration. Then, defining $\bar{\boldsymbol{t}}^{(n)} := \boldsymbol{F}\bar{\boldsymbol{s}}^{(n)}$, (17) becomes

$$\hat{t}_k^{(n)} \;=\; \bar{t}_k^{(n)} + \boldsymbol{g}_k^{(n)H}(\boldsymbol{x}_k - \boldsymbol{\mathcal{H}}_k \bar{\boldsymbol{t}}^{(n)}) \tag{18}$$

$$\boldsymbol{g}_k^{(n)} \;:=\; \left(\boldsymbol{\mathcal{H}}_k \boldsymbol{F}\, \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{F}^H \boldsymbol{\mathcal{H}}_k^H + \sigma^2 \boldsymbol{C}_k \boldsymbol{C}_k^H\right)^{-1} \boldsymbol{\mathcal{H}}_k \boldsymbol{F}\, \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{F}^H \boldsymbol{i}_k \tag{19}$$

from which estimates of $\boldsymbol{s}$ can be obtained as

$$\hat{\boldsymbol{s}}^{(n)} \;=\; \boldsymbol{F}^H \hat{\boldsymbol{t}}^{(n)} \quad \Leftrightarrow \quad \hat{s}_l^{(n)} \;=\; \boldsymbol{i}_l^H \boldsymbol{F}^H \sum_k \boldsymbol{i}_k \hat{t}_k^{(n)}. \tag{20}$$

2) A Conditionally Gaussian Model: Leveraging the finite-alphabet structure of the elements $\{s_k\}$ and assuming reasonably large PN (to invoke the Central Limit Theorem), we assume that the estimation error is Gaussian, or, equivalently, that the estimates are conditionally Gaussian:

$$p(\hat{s}_l^{(n)}|s_l = b) \;=\; \frac{1}{\sigma_l^{(n)}(b)} \phi\Big(\frac{\hat{s}_l^{(n)} - \mu_l^{(n)}(b)}{\sigma_l^{(n)}(b)} \Big), \tag{21}$$

where $\phi(w) := \frac{1}{\sqrt{\pi}} e^{-w^2}$, $\mu_l^{(n)}(b) := \mathrm{E}\{\hat{s}_l^{(n)}|s_l = b\}$, and $[\sigma_l^{(n)}(b)]^2 := \mathrm{Cov}(\hat{s}_l^{(n)}, \hat{s}_l^{(n)}|s_l = b)$.

In Appendix I we show that

$$\mu_l^{(n)}(b) \;=\; \bar{s}_l^{(n)} + Q_{l,l}^{(n)*}(b - \bar{s}_l^{(n)}) \tag{22}$$

$$[\sigma_l^{(n)}(b)]^2 \;=\; \boldsymbol{q}_l^{(n)H} \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{q}_l^{(n)} - |Q_{l,l}^{(n)}|^2 v_l^{(n)} + \sigma^2 \|\boldsymbol{p}_l^{(n)}\|^2, \tag{23}$$

where $\boldsymbol{q}_l^{(n)}$ denotes the l^{th} column of $\boldsymbol{Q}^{(n)}$ and where $\boldsymbol{p}_l^{(n)}$ denotes the l^{th} column of $\boldsymbol{P}^{(n)}$:

$$\boldsymbol{Q}^{(n)} \;=\; \boldsymbol{F}^H \Big(\sum_k \boldsymbol{\mathcal{H}}_k^H \boldsymbol{g}_k^{(n)} \boldsymbol{i}_k^H \Big) \boldsymbol{F} \tag{24}$$

$$\boldsymbol{P}^{(n)} \;=\; \Big(\sum_k \boldsymbol{C}_k^H \boldsymbol{g}_k^{(n)H} \boldsymbol{i}_k^H \Big) \boldsymbol{F}. \tag{25}$$

3) Log-Likelihood Ratio and Update of Priors: From now on, we restrict ourselves to the BPSK alphabet so that $b \in \{-1, +1\}$; QAM extensions are straightforward but tedious (see, e.g., [19], [20]). The n^{th}-iteration *a priori* and *a posteriori* LLRs are then defined as $L_l^{(n)} := \log \frac{P(s_l=+1)}{P(s_l=-1)}$ and $L_l(\hat{s}_l^{(n)}) := \log \frac{P(s_l=+1|\hat{s}_l^{(n)})}{P(s_l=-1|\hat{s}_l^{(n)})}$, respectively. Note that, after the first iteration, we expect to have partial information on s_l such that $L_l^{(n)} \neq 0$. The LLR update $\Delta(\hat{s}_l^{(n)}) := L_l(\hat{s}_l^{(n)}) - L_l^{(n)}$ can be written

$$\begin{aligned} \Delta(\hat{s}_l^{(n)}) \;&=\; \log \frac{p(\hat{s}_l^{(n)}|s_l = +1)}{p(\hat{s}_l^{(n)}|s_l = -1)} \\ &=\; \frac{|\hat{s}_l^{(n)} - \mu_l^{(n)}(-1)|^2 - |\hat{s}_l^{(n)} - \mu_l^{(n)}(+1)|^2}{[\sigma_l^{(n)}(\pm 1)]^2} \\ &=\; \frac{4\Big(\mathrm{Re}\big(Q_{l,l}^{(n)}(\hat{s}_l^{(n)} - \bar{s}_l^{(n)}) \big) + |Q_{l,l}^{(n)}|^2 \bar{s}_l^{(n)} \Big)}{\boldsymbol{q}_l^{(n)H} \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{q}_l^{(n)} - |Q_{l,l}^{(n)}|^2 v_l^{(n)} + \sigma^2 \|\boldsymbol{p}_l^{(n)}\|^2}, \end{aligned} \tag{26}$$

where we used the facts that $\sigma_l^{(n)}(+1) = \sigma_l^{(n)}(-1)$ and

$$\begin{aligned}
&\big|\hat{s}_l^{(n)} - \mu_l^{(n)}(-1)\big|^2 - \big|\hat{s}_l^{(n)} - \mu_l^{(n)}(+1)\big|^2 \\
&= \big|\hat{s}_l^{(n)} - (1 - Q_{l,l}^{(n)*})\bar{s}_l^{(n)} + Q_{l,l}^{(n)*}\big|^2 \\
&\quad - \big|\hat{s}_l^{(n)} - (1 - Q_{l,l}^{(n)*})\bar{s}_l^{(n)} - Q_{l,l}^{(n)*}\big|^2 \\
&= 4\,\mathrm{Re}\big\{\big(\hat{s}_l^{(n)} - (1 - Q_{l,l}^{(n)*})\bar{s}_l^{(n)}\big)Q_{l,l}^{(n)}\big\} \\
&= 4\,\mathrm{Re}\big\{Q_{l,l}^{(n)}(\hat{s}_l^{(n)} - \bar{s}_l^{(n)})\big\} + |Q_{l,l}^{(n)}|^2\bar{s}_l^{(n)}, \qquad (27)
\end{aligned}$$

since the use of BPSK implies $\bar{s}_l^{(n)} \in \mathbb{R}$. Updates of the symbol mean and variance can be accomplished via

$$\begin{aligned}
\bar{s}_l^{(n+1)} &= \sum_{b\in\mathcal{B}} b \cdot P(s_l = b|\hat{s}_l^{(n)}) \\
&= \tanh\left(\frac{L_l(\hat{s}_l^{(n)})}{2}\right) \qquad (28) \\
v_l^{(n+1)} &= \sum_{b\in\mathcal{B}} \big(b - \bar{s}_l^{(n+1)}\big)^2 P(s_l = b|\hat{s}_l^{(n)}) \\
&= 1 - (\bar{s}_l^{(n+1)})^2. \qquad (29)
\end{aligned}$$

To update the *a priori* LLR, we set $L_l^{(n+1)} := L_l(\hat{s}_l^{(n)})$, giving

$$L_l^{(n+1)} = L_l^{(n)} + \Delta(\hat{s}_l^{(n)}). \qquad (30)$$

Hard symbol estimates can be generated as $\hat{\hat{s}}_l^{(n)} := \mathrm{sign}\big(\mathrm{Re}(\hat{s}_l^{(n)})\big) = \mathrm{sign}\big(\bar{s}_l^{(n)}\big) = \mathrm{sign}\big(L(s_l|\hat{s}_l^{(n)})\big)$. An algorithm summary appears in Table I. Note that a soft decoding algorithm could be easily embedded within the bottom path of Fig. 3, as proposed in [19] and investigated in [20].

B. Interframe Processing

As previously discussed, the use of max-SINR windowing causes less energy to be collected from symbols near the edges of frame $\boldsymbol{s}(i)$ than from those near the center of the frame. As a result, the iterative detection algorithm described in Sec. IV-A is more likely to incorrectly detect symbols near the frame edges. However, by overlapping the frames (i.e., choosing $P > 1$), we can exploit the fact that every symbol will be near the center of some frame. Specifically, (3) implies that s_m maps to the frame-quantities $\Big\{s_{\langle m\rangle_N}(\lfloor\frac{m}{N}\rfloor), s_{\langle m\rangle_N+N}(\lfloor\frac{m}{N}\rfloor - 1), \ldots, s_{\langle m\rangle_N+(P-1)N}(\lfloor\frac{m}{N}\rfloor - P + 1)\Big\}$, i.e., s_m appears in P distinct frames. The frame index i_m for which s_m appears closest to frame center is easily

found to be

$$i_m = \left\lfloor \frac{m}{N} \right\rfloor - j_m \tag{31}$$

$$j_m := \arg \min_{j=0,\ldots,P-1} \left| \langle m \rangle_N + jN - \frac{PN}{2} \right|. \tag{32}$$

Thus, in exploiting frame overlap, we stipulate that

1) the hard estimate of s_m is generated at frame index i_m, i.e., $\hat{\hat{s}}_m = \hat{\hat{s}}_{\langle m \rangle_N + j_m N}(i_m)$, and
2) the final LLR calculated for symbol s_m during frame i_m is used to initialize the LLR of that symbol in subsequent frames within which it appears, i.e., in frames with index $i \in \{i_m + 1, i_m + 2, \ldots, \lfloor \frac{m}{N} \rfloor\}$.

In the case that BDFE is employed, these hard estimates are then also used for post-cursor IFI cancellation. Figure 4 illustrates this process for $P = 2$.

Since every symbol s_m is estimated P times, the overall equalizer complexity increases linearly with P. Numerical simulations suggest that the performance with $P > 2$ is not significantly better than $P = 2$, while the performance with $P = 1$ is relatively poor. Hence, we focus on $P = 2$ for the remainder of the paper.

V. Fast Algorithm and Complexity Analysis

In Table II we present a fast version of the detection algorithm summarized in Table I. In the fast version, we avoid explicit computation of $\boldsymbol{Q}^{(n)}$ and $\boldsymbol{P}^{(n)}$, instead computing $y_k^{(n)} := \boldsymbol{q}_k^{(n)H} \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{q}_k^{(n)}$, $z_k^{(n)} := \|\boldsymbol{p}_k^{(n)}\|^2$, and $Q_{k,k}^{(n)}$ for $k \in \{0, 1, \ldots, PN-1\}$. The approximate number of complex multiplications[1] per step is given in the right column of Table II, and per-symbol averages are summarized in Table III (assuming M iterations) for both BDFE and non-BDFE cases. We include the cost of transforming the known time-domain channel coefficients $\{h_{n,l}\}$ to frequency-domain channel coefficients $\{\boldsymbol{\mathcal{H}}(i,\ell)\}$, as well as that of post-cursor IFI cancellation in the BDFE case. Table III also includes the per-symbol cost of a fast version of the LTV-channel FIR-MMSE-DFE.[2]

The details of each step are enumerated below in correspondence with the left column of Table II. For brevity, we use $\widetilde{D} := 2D + 1$ in the sequel. We make frequent use of the property $\boldsymbol{F}\,\mathcal{D}(\boldsymbol{a})\boldsymbol{F}^H = \mathcal{C}(\boldsymbol{F}\boldsymbol{a}/\sqrt{PN})$. Finally, we assume that PN-length FFTs require $\frac{1}{2}PN\log_2(PN)$ and $PN\log_2(PN)$ complex multiplies for real- and complex-valued inputs, respectively (as per the radix-2 Cooley-Tukey algorithm [25]).

[1]While the number of additions and divisions could also be counted, we feel that such an endeavor would complicate the presentation without providing significant additional insight.

[2]See the "Fast FIR-MMSE-DFE Details" document which will be made available as a technical report.

Detail 1: At each frame index i, we must compute the frequency domain coefficients $\boldsymbol{\mathcal{H}}(i,0)$, or $\{\boldsymbol{\mathcal{H}}(i,\ell)\}_{\ell=0}^{L_{\text{pst}}}$ when BDFE is used, from the time-domain channel coefficients $\{h_{n,l}\}$. If we define the time-domain matrix $\overline{\boldsymbol{H}}(i,\ell) \in \mathbb{C}^{N_b \times PN}$ such that $[\overline{\boldsymbol{H}}(i,\ell)]_{n,l} := h_{iN+n,l} a_{\ell PN+n-l}$, then (9) can be rewritten

$$H_{d,k}(i,\ell) \;=\; \sum_{n=0}^{N_b-1} \sum_{l=0}^{N_h-1} [\boldsymbol{F}\boldsymbol{J}]_{d,n} [\mathcal{D}(\boldsymbol{b})]_{n,n} [\overline{\boldsymbol{H}}(i,\ell)]_{n,l} [\boldsymbol{F}]_{l,k} \tag{33}$$

which implies

$$\boldsymbol{H}(i,\ell) \;=\; \boldsymbol{F}\boldsymbol{J}\,\mathcal{D}(\boldsymbol{b})\overline{\boldsymbol{H}}(i,\ell)\boldsymbol{F} \tag{34}$$

for the frequency-domain matrix $\boldsymbol{H}(i,\ell)$, a rearrangement of $\boldsymbol{\mathcal{H}}(i,\ell)$ defined such that $[\boldsymbol{H}(i,\ell)]_{d,k} = H_{d,k}(i,\ell)$. Note that $\overline{\boldsymbol{H}}(i,\ell)$ has at most PNN_h nonzero elements, and that we are only interested in computing only $\widetilde{D}$ rows of $\boldsymbol{H}(i,\ell)$. With this in mind, we see that computation of $\boldsymbol{J}\,\mathcal{D}(\boldsymbol{b})\overline{\boldsymbol{H}}(i,\ell)$ costs PNN_h multiplies. Multiplication by the left $\boldsymbol{F}$ in (34) can be accomplished by N_h FFTs for a cost of $N_h PN \log_2 PN$ multiplies, after which only $\widetilde{D}$ rows of $\boldsymbol{F}\boldsymbol{J}\,\mathcal{D}(\boldsymbol{b})\overline{\boldsymbol{H}}(i,\ell)$ are retained. Finally, multiplication by the right $\boldsymbol{F}$ in (34) can be accomplished using $\widetilde{D}$ FFTs, for a cost of $\widetilde{D}PN \log_2 PN$ multiplies. In total, $PNN_h + (N_h + \widetilde{D})PN \log_2 PN$ multiplies are needed for each (i,ℓ) pair.

Detail 2: In the BDFE case, the frequency domain observation is computed as

$$\boldsymbol{x}(i) \;=\; \boldsymbol{F}\boldsymbol{J}\,\mathcal{D}(\boldsymbol{b})\boldsymbol{r}(i) - \sum_{\ell=1}^{L_{\text{pst}}} \boldsymbol{\mathcal{H}}(i,\ell)\hat{\tilde{\boldsymbol{t}}}(i-\ell P), \tag{35}$$

where $\hat{\tilde{\boldsymbol{t}}}(i) := \boldsymbol{F}\hat{\boldsymbol{s}}(i)$. The non-BDFE case is similar, but without the IFI cancellation. The first term in $\boldsymbol{x}(i)$ requires $N_b + PN \log_2 PN$ multiplications per frame to compute, while the second requires $L_{\text{pst}}\widetilde{D}PN$ since $\boldsymbol{\mathcal{H}}(i,\ell)$ contains only $\widetilde{D}PN$ non-zero elements. Since $\hat{\tilde{\boldsymbol{t}}}(i)$ needs to be computed only when i is a multiple of P, it requires an average of $\frac{1}{P}PN \log_2 PN$ multiplications per frame. Using $P = 2$ and the approximation $N_b \approx PN$, we get a total of $(\widetilde{D}L_{\text{pst}} + 1)PN + 1.5PN \log_2 PN$ multiplications per frame.

Detail 3: From (11) and the property $\boldsymbol{J}\,\mathcal{D}(\boldsymbol{b}) = \mathcal{D}(\boldsymbol{J}\boldsymbol{b})$, it follows that $\boldsymbol{C}\boldsymbol{C}^H = \boldsymbol{F}\,\mathcal{D}(\sigma^2 \boldsymbol{J}\boldsymbol{b} \odot \boldsymbol{J}\boldsymbol{b}^*)\boldsymbol{F}^H = \mathcal{C}\left(\sqrt{PN}\sigma^2 \boldsymbol{F}\,(\boldsymbol{J}\boldsymbol{b} \odot \boldsymbol{J}\boldsymbol{b}^*)\right)$. Thus, the PN coefficients that specify $\boldsymbol{C}\boldsymbol{C}^H$ can be computed in roughly $2PN + \frac{1}{2}PN \log_2 PN$ multiplies. Notice that $\boldsymbol{C}_k\boldsymbol{C}_k^H$ is a sub-block of $\boldsymbol{C}\boldsymbol{C}^H$, and that the Toeplitz nature of $\boldsymbol{C}\boldsymbol{C}^H$ implies that $\boldsymbol{C}_k\boldsymbol{C}_k^H$ is identical for every k.

Detail 4: This step initializes the recursive computation of $\boldsymbol{R}_k^{(n)} := \sqrt{PN}\left(\boldsymbol{\mathcal{H}}_k \boldsymbol{F}\,\mathcal{D}(\boldsymbol{v}^{(n)})\boldsymbol{F}^H \boldsymbol{\mathcal{H}}_k^H + \boldsymbol{C}_k\boldsymbol{C}_k^H\right)$, where we note $\sqrt{PN}\boldsymbol{F}\,\mathcal{D}(\boldsymbol{v}^{(n)})\boldsymbol{F}^H = \mathcal{C}(\boldsymbol{u}^{(n)})$. For computation of $\boldsymbol{\mathcal{H}}_0\,\mathcal{C}(\boldsymbol{u}^{(n)})\boldsymbol{\mathcal{H}}_0^H$, we first compute $\boldsymbol{\mathcal{H}}_0\,\mathcal{C}(\boldsymbol{u}^{(n)})$, then post-multiply the result by $\boldsymbol{\mathcal{H}}_0^H$. But since $\boldsymbol{\mathcal{H}}_0^H$ contains only $4D+1 \approx 2\widetilde{D}$ non-zero rows, only $2\widetilde{D}$ non-zero

columns of $\boldsymbol{\mathcal{H}}_0\,\mathcal{C}(\boldsymbol{u}^{(n)})$ need be computed. This requires $2\widetilde{D}^3$ multiplications, since $\boldsymbol{\mathcal{H}}_0$ contains $\widetilde{D}$ rows, each with only $\widetilde{D}$ non-zero elements. Using a similar reasoning, the post-multiplication also requires $\widetilde{D}^3$ multiplications.

Detail 5: $\boldsymbol{R}_k^{(n)}$ can be inverted directly or recursively since $\boldsymbol{R}_{k+1}^{(n)} = \begin{bmatrix} \boldsymbol{\Theta}_k & \tilde{\boldsymbol{\theta}}_k \\ \tilde{\boldsymbol{\theta}}_k^H & \tilde{\theta}_k \end{bmatrix}$ when $\boldsymbol{R}_k^{(n)} = \begin{bmatrix} \theta_k & \boldsymbol{\theta}_k^H \\ \boldsymbol{\theta}_k & \boldsymbol{\Theta}_k \end{bmatrix}$. In the direct method, we first compute $[\tilde{\boldsymbol{\theta}}_k^t \;\; \tilde{\theta}_k]^t$ to obtain $\boldsymbol{R}_{k+1}^{(n)}$ from $\boldsymbol{R}_k^{(n)}$. Cost-wise, this is similar to computing one column (i.e., $1/\widetilde{D}$ of the total elements) of $\boldsymbol{R}_0^{(n)}$, requiring $2\widetilde{D}^2$ multiplies. The direct inversion of Hermitian $\boldsymbol{R}_{k+1}^{(n)}$ then requires an additional $\frac{1}{3}\widetilde{D}^3$ multiplies (using LDL* factorization [23]). The procedure for recursive computation of $(\boldsymbol{R}_{k+1}^{(n)})^{-1}$ follows directly from the well-known block-matrix inversion formula [26] $\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1}(\boldsymbol{I} + \boldsymbol{B}\boldsymbol{P}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1}) & -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{P}^{-1} \\ -\boldsymbol{P}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{P}^{-1} \end{bmatrix}$, where $\boldsymbol{P} := \boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}$, and is detailed in Table IV. In summary, the total cost of the direct and recursive inversions are approximately $2\widetilde{D}^2 + \frac{1}{3}\widetilde{D}^3$ and $7\widetilde{D}^2$ multiplications, respectively.

Detail 6: Since $\boldsymbol{\mathcal{H}}_k$ contains $\widetilde{D}$ rows, each with only $\widetilde{D}$ nonzero elements, the calculation of $\boldsymbol{\mathcal{H}}_k\tilde{\boldsymbol{V}}^{(n)}\boldsymbol{i}_k$ consumes only $\widetilde{D}^2$ multiplies. Multiplication by $(\boldsymbol{R}_k^{(n)})^{-1}$ consumes an additional $\widetilde{D}^2$.

Detail 7: LLR updating requires $\{y_k^{(n)}\}_{k=0}^{PN-1}$, where $y_k^{(n)} := \boldsymbol{q}_k^{(n)H}\,\mathcal{D}(\boldsymbol{v}^{(n)})\boldsymbol{q}_k^{(n)}$. Note that the explicit calculation of $\boldsymbol{Q}^{(n)}$, as defined in (24), would involve $2PN$ FFTs of length PN, and thus a total complexity of $\mathcal{O}(P^2N^2\log_2 PN)$. In Appendix II-A we show that

$$\boldsymbol{y}^{(n)} \;=\; \frac{1}{\sqrt{PN}} \sum_{d,l=-2D}^{2D} \boldsymbol{F}\,\mathcal{D}\left(\boldsymbol{T}_{l-d}\bar{\boldsymbol{u}}^{(n)}\right)\boldsymbol{F}^H(\boldsymbol{\alpha}_d^{(n)} \odot \boldsymbol{\alpha}_l^{(n)*}) \tag{36}$$

where $[\boldsymbol{y}^{(n)}]_k = y_k^{(n)}$, $\bar{\boldsymbol{u}}^{(n)} := \boldsymbol{F}^H\boldsymbol{v}^{(n)}$, $\boldsymbol{T}_k := \mathcal{C}(\boldsymbol{i}_{\langle k\rangle_{PN}})$ is the right circular k-shift matrix, and where

$$\boldsymbol{\alpha}_d^{(n)} \;=\; \boldsymbol{F}\,\mathrm{diag}_d(\boldsymbol{G}^{(n)}) \tag{37}$$

$$\boldsymbol{G}^{(n)} \;=\; \sum_{k=0}^{PN-1} \boldsymbol{\mathcal{H}}_k^H\boldsymbol{g}_k^{(n)}\boldsymbol{i}_k \tag{38}$$

Note that $\bar{\boldsymbol{u}}^{(n)}$ is simply a rearrangement of $\boldsymbol{u}^{(n)}$. The k^{th} column of $\boldsymbol{G}^{(n)}$ equals $\boldsymbol{\mathcal{H}}_k^H\boldsymbol{g}_k^{(n)}$ and requires $\widetilde{D}^2$ multiplies to compute, and so $\boldsymbol{G}^{(n)}$ requires $PN\widetilde{D}^2$ multiplies to compute. Computation of $\{\boldsymbol{\alpha}_d\}_{d=-2D}^{2D}$ involves $4D+1 \approx 2\widetilde{D}$ FFTs for a total cost of $2\widetilde{D}PN\log_2 PN$ multiplies. For each (d,l) pair, the computation of (36) requires an additional $2PN + 2PN\log_2 PN$ multiplies. However, due to conjugate symmetry, only about half of the $\approx 4\widetilde{D}^2$ pairs need be evaluated. Hence, using (36) rather than direct computation of $\boldsymbol{Q}^{(n)}$, the calculation of $\{y_k^{(n)}\}_{k=0}^{PN-1}$ requires only about $4\widetilde{D}^2(PN + PN\log_2 PN) + 2\widetilde{D}PN\log_2 PN + PN\widetilde{D}^2$, or $5\widetilde{D}^2PN + (4\widetilde{D}^2 + 2\widetilde{D})PN\log_2 PN$, multiplies.

Detail 8: LLR updating also requires $\{z_k^{(n)}\}_{k=0}^{PN-1}$, where $z_k^{(n)} := \|\boldsymbol{p}_k^{(n)}\|^2$. In Appendix II-B we show that

$$\boldsymbol{z}^{(n)} \;=\; \frac{1}{\sqrt{PN}} \sum_{d,l=-D}^{D} \boldsymbol{F}\,\mathcal{D}\left(\boldsymbol{T}_{l-d}\boldsymbol{F}^H\bar{\boldsymbol{b}}\right)\boldsymbol{F}^H(\bar{\boldsymbol{\alpha}}_d^{(n)} \odot \bar{\boldsymbol{\alpha}}_l^{(n)*}), \tag{39}$$

where $[\boldsymbol{z}^{(n)}]_k = z_k^{(n)}$, $[\bar{\boldsymbol{b}}]_m := \sum_{n=0}^{N_b-1} |b_{\langle n\rangle_{PN}}|^2 \delta_{\langle n\rangle_{PN}-m}$, and

$$\bar{\boldsymbol{\alpha}}_d^{(n)} \;=\; \boldsymbol{F}\,\mathrm{diag}_d(\bar{\boldsymbol{G}}^{(n)}) \tag{40}$$

$$\bar{\boldsymbol{G}}^{(n)} \;=\; \sum_{k=0}^{PN-1} \bar{\boldsymbol{g}}_k^{(n)} \boldsymbol{i}_k \tag{41}$$

Note that $\boldsymbol{F}^H\bar{\boldsymbol{b}}$ can be computed in advance, $\bar{\boldsymbol{G}}^{(n)}$ requires no computation, and $\{\bar{\boldsymbol{\alpha}}_d^{(n)}\}_{d=-D}^{D}$ involves $\widetilde{D}$ FFTs, for a total cost of $\widetilde{D}PN\log_2 PN$ multiplies. For each (d,l) pair, (39) requires an additional $2PN+2PN\log_2 PN$ multiplies, but only about half of the $\widetilde{D}^2$ pairs need be evaluated (due to conjugate symmetry). Hence, calculation of $\{z_k^{(n)}\}_{k=0}^{PN-1}$ requires about $\frac{1}{2}\widetilde{D}^2(2PN+2PN\log_2 PN)+\widetilde{D}PN\log_2 PN$, or $\widetilde{D}^2PN+(\widetilde{D}^2+\widetilde{D})PN\log_2 PN$, multiplies.

Detail 9: LLR updating also requires $\{Q_{k,k}^{(n)}\}_{k=0}^{PN-1}$. From (24), (37), (38), and Lemma 1, it follows that

$$Q_{k,k}^{(n)} \;=\; \frac{1}{\sqrt{PN}} \sum_{d=-2D}^{2D} [\boldsymbol{\alpha}_d^{(n)}]_0\, e^{j\frac{2\pi}{PN}kd}. \tag{42}$$

As reported in Table II, direct evaluation of (42) requires $4D+1 \approx 2\widetilde{D}$ multiplies for each k. Note that, if $2\widetilde{D} > \log_2 PN$, it would be more efficient to compute $\{Q_{k,k}^{(n)}\}_{k=0}^{PN-1}$ using a single PN-point FFT. However, since the cost of this step is relatively small, the difference is insignificant.

VI. Numerical Results

In this section, we compare the performance and complexity of the fast iterative frequency domain equalization (IFDE) algorithm summarized in Table II with the well known FIR-MMSE-DFE. While the FIR-MMSE-DFE was originally derived for LTI channels [17], it can be straightforwardly extended to the LTV channel case.[3] and then design a recursive algorithm to update the filter coefficients at the symbol rate assuming a fixed estimation delay Δ. In all simulations, BPSK symbols are transmitted over a noisy WSSUS Rayleigh-fading channel with uniform power profile (i.e., $\sigma_l^2 = N_h^{-1}$) that is generated using Jakes method [27]. Throughout, we assume IFDE uses an ICI radius of $D = \lceil f_{\mathsf{d}} T_{\mathsf{s}} PN \rceil$ and frame overlap factor of $P = 2$. Both IFDE and FIR-MMSE-DFE designs are based on known time-domain coefficients $\{h_{n,l}\}_{l=0}^{N_h-1}$.

[3]See the "Fast FIR-MMSE-DFE Details" document which will be made available as a technical report.

First, we establish IFDE-BDFE design rules for frame length PN and number-of-iterations M. While we will see that smaller values of PN (for fixed N_h) are advantageous from a complexity standpoint (see Fig. 7), experiments[4] suggest the choice $PN \geq 4N_h$ for good symbol error rate (SER) performance. With radix-2 FFTs in mind, we choose $PN = 2^{\lceil \log_2 4N_h \rceil}$ in the sequel. A related set of experiments[5] has shown that SER performance improves with M up to about $M = 10$, after which there is little additional improvement. Interestingly, we find that, after 2 iterations, IFDE-BDFE gives approximately the same performance as FIR-MMSE-DFE. Hence, we focus on IFDE-BDFE-2 and IFDE-BDFE-10 in the sequel, i.e., IFDE-BDFE using $M = 2$ and $M = 10$, respectively.

Next, we establish FIR-MMSE-DFE design rules for feedforward filter length N_f and estimation delay Δ, assuming that the feedback filter is just long enough to cancel all post-cursor ISI. To investigate the effect of Δ, we fixed $N_f = N_h$ and conducted experiments[6] measuring MSE for several values of N_f (assuming $f_{\mathsf{d}}T_{\mathsf{s}} = 0.0075$ and SNR=10dB). Since the choice $\Delta = N_f - 1$ maximized performance in every case, we adopt this rule. To investigate the effect of N_f, we fixed $\Delta = N_f - 1$ and conducted experiments measuring MSE[7] at several values of SNR (when $f_{\mathsf{d}}T_{\mathsf{s}} = 0.0075$ and $N_h = 64$). In every case, performance increased with N_f, though the gains diminished rapidly when $N_f > N_h$. With complexity in mind, we adopt the rule $N_f = N_h$.

Having established IFDE-BDFE and FIR-MMSE-DFE design rules, we are ready to compare the two approaches in performance and complexity. In Fig. 6, we compare SER performances when $N_h = 64$ and $f_{\mathsf{d}}T_{\mathsf{s}} \in \{0.001, 0.003, 0.0075\}$ over a wide range of SNR. Note that, at all $f_{\mathsf{d}}T_{\mathsf{s}}$, IFDE-BDFE-2 performs equivalently to FIR-MMSE-DFE whereas IFDE-BDFE-10 outperforms FIR-MMSE-DFE, significantly so when SNR> 5. We also plot the matched-filter bound (MFB) [18]—the ultimate in (uncoded) receiver performance—which is not far from IFDE-BDFE-10.

Figure 7 examines the multiplies-per-symbol ratio of FIR-MMSE-DFE to IFDE-BDFE-2 using the expressions in Table III. Note that values > 1 in Fig. 7 imply a complexity *advantage* for IFDE-BDFE, and that this complexity advantage increases with N_h and decreases with $f_{\mathsf{d}}T_{\mathsf{s}}$. Since FIR-MMSE-DFE and IFDE-BDFE-2 have similar performance, Fig. 7 constitutes a direct *complexity* comparison. A similar comparison[8] between FIR-MMSE-DFE and IFDE-BDFE-10 also shows complexity advantage over a wide range of $(N_h, f_{\mathsf{d}}T_{\mathsf{s}})$.

A final comment regarding the complexity comparison Fig. 7 is in order. One could argue that the FIR-MMSE-

[4]See Fig. 1 in the "Simulation Details" document.
[5]See Fig. 2 in the "Simulation Details" document.
[6]See Fig. 3 in the "Simulation Details" document.
[7]See Fig. 4 in the "Simulation Details" document.
[8]See Fig. 7 in the "Simulation Details" document.

DFE, which—for our LTV channels—calculates a filter update *every* symbol period, is "overkill" for slowly varying channels. For these channels, decent performance should result from approximating the LTV channel response as *fixed* over, say, N_f symbol intervals and designing a single *fixed* MMSE-DFE to operate over this N_f-symbol interval. In this case, there exist computationally efficient implementations that exploit the Toeplitz structure of the channel matrix [17], [28]. But for what range of $(f_{\mathsf{d}}T_{\mathsf{s}}, N_h)$ will the channel be "slow enough" for this block-LTI approximation to hold? Numerical experiments[9] at 10dB SNR have shown that this block-LTI approximation results in an equivalent SNR loss of $\approx$ 3dB when $f_{\mathsf{d}}T_{\mathsf{s}}N_h > 0.11$ and a loss of $\approx$ 1dB when $f_{\mathsf{d}}T_{\mathsf{s}}N_h > 0.06$. For reference, the curves $f_{\mathsf{d}}T_{\mathsf{s}}N_h = 0.11$ and $f_{\mathsf{d}}T_{\mathsf{s}}N_h = 0.06$ were superimposed on Figs. 7.

VII. Conclusion

In this paper, we presented an iterative frequency domain equalization (IFDE) scheme for single-carrier transmissions over noisy doubly dispersive channels. Time-domain windowing is used make the effective ICI/IFI response sparse, after which iterative symbol estimation is performed in the frequency domain. The estimation algorithm leverages the finite-alphabet property of symbols, the sparse ICI/IFI structure, and the low computational cost of the FFT. Simulations show that the IFDE performs significantly better than the FIR-MMSE-DFE and within about 1 dB of the MFB over the SNR range of interest. A fast version of the IFDE algorithm was also derived and its complexity compared to that of a fast FIR-MMSE-DFE for LTV channels. The IFDE algorithm was found to yield significant cost savings relative to the fast FIR-MMSE-DFE for reasonable channel lengths.

References

[1] I. M. Garrison, R. K. Martin, W. Sethares, B. Hart, W. Chung, J. Balakrishnan, R. Casas, T. Endres, M. Larimore, P. Schniter, and C. Johnson, Jr., "DTV channel characterization," in *Proc. Conf. Inform. Science and Systems*, pp. 584–589, Mar. 2001.

[2] S. B. Weinstein and P. M. Ebert, "Data transmission by frequency division multiplexing using the discrete Fourier transform," *IEEE Trans. Commun.*, vol. 19, pp. 628–634, Oct. 1971.

[3] L. J. Cimini, Jr., "Analysis and simulation of a digital mobile radio channel using orthogonal frequency division multiplexing," *IEEE Trans. Commun.*, vol. 33, pp. 665–765, July 1985.

[4] H. Sari, G. Karam, and I. Jeanclaude, "Transmission techniques for digital terrestrial TV broadcasting," *IEEE Commun. Mag.*, pp. 100–109, Feb. 1995.

[5] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Commun. Mag.*, vol. 40, pp. 58–66, Apr. 2002.

[9] See Figs. 5-6 in the "Simulation Details" document.

[6] T. Walzman and M. Schwartz, "Automatic equalization using the discrete frequency domain," *IEEE Trans. Inform. Theory*, vol. 19, pp. 59–68, Jan. 1973.

[7] E. R. Ferrara, Jr., "Frequency-domain adaptive filtering," in *Adaptive Filters* (C. F. N. Cowan and P. M. Grant, eds.), Prentice-Hall, 1985.

[8] D. Kim and G. St uber, "Residual ISI cancellation for OFDM with application to HDTV broadcasting," *IEEE J. Select. Areas In Commun.*, vol. 16, no. 8, pp. 1590–1599, 1998.

[9] A. Stamoulis, S. N. Diggavi, and N. Al-Dhahir, "Intercarrier interference in MIMO OFDM," *IEEE Trans. Signal Processing*, vol. 50, pp. 2451–2464, Oct. 2002.

[10] X. Cai and G. B. Giannakis, "Bounding performance and suppressing inter-carrier interference in wireless mobile OFDM," *IEEE Trans. Commun.*, vol. 51, pp. 2047–2056, Dec. 2003.

[11] A. Gorokhov and J.-P. Linnartz, "Robust OFDM receivers for dispersive time-varying channels: Equalization and channel acquisition," *IEEE Trans. Commun.*, vol. 52, pp. 572–583, Apr. 2004.

[12] P. Schniter, "Low-complexity equalization of OFDM in doubly-selective channels," *IEEE Trans. Signal Processing*, vol. 52, pp. 1002–1011, Apr. 2004.

[13] G. Leus, I. Barhumi, and M. Moonen, "Equalization techniques for fading channels," in *Handbook on Signal Processing for Communications* (M. Ibnkahla, ed.), CRC Press, 2004.

[14] P. Schniter and H. Liu, "Iterative equalization for single carrier cyclic-prefix in doubly-dispersive channels," in *Proc. Asilomar Conf. Signals, Systems and Computers*, Oct. 2003.

[15] U. S. A. T. S. Committee, "Guide to the use of the atsc digital television standard." ATSC Doc. A/54, Oct. 1995.

[16] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE J. Oceanic Eng.*, vol. 25, Jan. 2000.

[17] N. Al-Dhahir and J. M. Cioffi, "MMSE decision feedback equalizers: Finite-length results," *IEEE Trans. Inform. Theory*, vol. 41, pp. 961–976, July 1995.

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

[19] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean square error equalization using *a priori* information," *IEEE Trans. Signal Processing*, vol. 50, pp. 673–683, Mar. 2002.

[20] P. Schniter and S. Das, "Multi-carrier modulation over doubly dispersive channels: Pulse shaping and equalization," *IEEE Trans. Signal Processing*, submitted 2004.

[21] D. D. Falconer and F. R. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell System Tech. J.*, vol. 52, pp. 1541–1562, Nov. 1973.

[22] P. Schniter, "A new approach to multicarrier pulse design for doubly-dispersive channels," in *Proc. Allerton Conf. Commun., Control, and Computing*, Oct. 2003.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: John Hopkins University Press, 3rd ed., 1996.

[24] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer, 2nd ed., 1994.

[25] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. New York: Springer, 1979.

[26] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY: Cambridge University Press, 1985.

[27] W. C. Jakes, *Microwave Mobile Communications*. Wiley, 1974.

[28] N. Al-Dhahir and J. M. Cioffi, "Efficient computation of the delay-optimized finite-length MMSE-DFE," *IEEE Trans. Signal Processing*, vol. 44, pp. 1288–1292, May 1996.

APPENDIX I
CONDITIONAL MEAN AND VARIANCE

From (18), (20), and the definition of $\mu_l^{(n)}(b)$,

$$\begin{aligned}\mu_l^{(n)}(b) &= \boldsymbol{i}_l^H \boldsymbol{F}^H \sum_k \boldsymbol{i}_k \,\mathrm{E}\{\hat{t}_k^{(n)}|s_l = b\} \\ &= \boldsymbol{i}_l^H \boldsymbol{F}^H \sum_k \boldsymbol{i}_k \Big(\bar{t}_k^{(n)} + \boldsymbol{g}_k^{(n)H}\big(\mathrm{E}\{\boldsymbol{x}_k|s_l = b\} - \boldsymbol{\mathcal{H}}_k \bar{\boldsymbol{t}}^{(n)}\big)\Big) \\ &= \bar{s}_l^{(n)} + \boldsymbol{i}_l^H \boldsymbol{Q}^{(n)H} \boldsymbol{i}_l (b - \bar{s}_l^{(n)})\end{aligned}$$

which leads to (22). In the last step above, we used the fact that $\mathrm{E}\{\boldsymbol{x}_k|s_l = b\} = \boldsymbol{\mathcal{H}}_k \boldsymbol{F}\big(\bar{\boldsymbol{s}}^{(n)} + \boldsymbol{i}_l(b - \bar{s}_l^{(n)})\big) = \boldsymbol{\mathcal{H}}_k \bar{\boldsymbol{t}}^{(n)} + \boldsymbol{\mathcal{H}}_k \boldsymbol{F} \boldsymbol{i}_l (b - \bar{s}_l^{(n)})$. Next we find an expression for $[\sigma_l^{(n)}(b)]^2$. Before doing so, however, it will be convenient to note from (18) and (20) that

$$\begin{aligned}\hat{s}_l^{(n)} &= \boldsymbol{i}_l^H \boldsymbol{F}^H \sum_k \boldsymbol{i}_k \Big(\bar{t}_k^{(n)} + \boldsymbol{g}_k^{(n)H}\big(\boldsymbol{x}_k - \boldsymbol{\mathcal{H}}_k \bar{\boldsymbol{t}}^{(n)}\big)\Big) \\ &= \boldsymbol{i}_l^H \boldsymbol{F}^H \sum_k \boldsymbol{i}_k \Big(\bar{t}_k^{(n)} + \boldsymbol{g}_k^{(n)H}\big(\boldsymbol{\mathcal{H}}_k \boldsymbol{F} \boldsymbol{s} + \boldsymbol{C}_k \boldsymbol{\nu} - \boldsymbol{\mathcal{H}}_k \boldsymbol{F} \bar{\boldsymbol{s}}^{(n)}\big)\Big) \\ &= \bar{s}_l^{(n)} + \boldsymbol{i}_l^H \boldsymbol{Q}^{(n)H}(\boldsymbol{s} - \bar{\boldsymbol{s}}^{(n)}) + \boldsymbol{i}_l^H \boldsymbol{P}^{(n)H} \boldsymbol{\nu} \\ &= \mu_l^{(n)}(b) + \boldsymbol{i}_l^H \boldsymbol{Q}^{(n)H}\big(\boldsymbol{s} - \bar{\boldsymbol{s}}^{(n)} + \boldsymbol{i}_l(\bar{s}_l^{(n)} - b)\big) + \boldsymbol{i}_l^H \boldsymbol{P}^{(n)H} \boldsymbol{\nu} \qquad (43)\end{aligned}$$

and that, since $\mathrm{E}\{\boldsymbol{s}|s_l = b\} = \bar{\boldsymbol{s}}^{(n)} - \boldsymbol{i}_l(\bar{s}_l^{(n)} - b)$,

$$\begin{aligned}&\mathrm{E}\Big\{\big(\boldsymbol{s} - \bar{\boldsymbol{s}}^{(n)} + \boldsymbol{i}_l(\bar{s}_l^{(n)} - b)\big)\big(\boldsymbol{s} - \bar{\boldsymbol{s}}^{(n)} + \boldsymbol{i}_l(\bar{s}_l^{(n)} - b)\big)^H |s_l = b\Big\} \\ &= \mathrm{Cov}(\boldsymbol{s}, \boldsymbol{s}|s_l = b) \\ &= \mathcal{D}(\boldsymbol{v}^{(n)}) - \boldsymbol{i}_l \boldsymbol{i}_l^H v_l^{(n)}. \qquad (44)\end{aligned}$$

Using (43), (44), and the definition of $\sigma_l^{(n)}(b)$,

$$\begin{aligned}[\sigma_l^{(n)}(b)]^2 &= \mathrm{E}\Big\{\big(\hat{s}_l^{(n)} - \mu_l^{(n)}(b)\big)\big(\hat{s}_l^{(n)} - \mu_l^{(n)}(b)\big)^H |s_l = b\Big\} \\ &= \boldsymbol{i}_l^H \boldsymbol{Q}^{(n)H}\Big(\mathcal{D}(\boldsymbol{v}^{(n)}) - \boldsymbol{i}_l \boldsymbol{i}_l^H v_l^{(i)}\Big)\boldsymbol{Q}^{(n)} \boldsymbol{i}_l + \sigma^2 \boldsymbol{i}_l^H \boldsymbol{P}^{(n)H} \boldsymbol{P}^{(n)} \boldsymbol{i}_l\end{aligned}$$

which leads to (23).

APPENDIX II
FAST-IFDE DETAILS

A. Derivation of (36)

Here we derive an expression for $\boldsymbol{y}^{(n)}$ enabling fast computation. First, however, we present a useful lemma. Without loss of generality, we omit superscripts in this appendix.

Lemma 1. *If $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{PN\times PN}$ has the banded structure of Fig. 1 with $2\rho D+1$ non-zero diagonals, and if $\boldsymbol{B} = \boldsymbol{F}^H\boldsymbol{\mathcal{H}}\boldsymbol{F}$, then*

$$[\boldsymbol{B}]_{n,m} \;=\; \frac{1}{\sqrt{PN}} \sum_{d=-\rho D}^{\rho D} e^{j\frac{2\pi}{PN}nd}[\boldsymbol{F}\,\mathrm{diag}_d(\boldsymbol{\mathcal{H}})]_{\langle m-n\rangle_{PN}}.$$

Proof. Denote $\boldsymbol{a}_d = \mathrm{diag}_d(\boldsymbol{\mathcal{H}})$, so that $[\boldsymbol{\mathcal{H}}]_{n,m} = a_{n-m,m}$ where $a_{k,l} := [\boldsymbol{a}_k]_l$. Then, since $\boldsymbol{a}_d = \boldsymbol{0}$ for $d \notin \{-\rho D,\dots,\rho D\}$,

$$\begin{aligned}
[\boldsymbol{B}]_{n,m} \;&=\; \frac{1}{PN}\sum_{k=0}^{PN-1}\sum_{l=0}^{PN-1} e^{j\frac{2\pi}{PN}nk}a_{k-l,l}e^{-j\frac{2\pi}{PN}lm}\\
&=\; \frac{1}{PN}\sum_{d=-\rho D}^{\rho D} e^{j\frac{2\pi}{PN}nd}\sum_{l=0}^{PN-1} a_{d,l}e^{-j\frac{2\pi}{PN}l(m-n)}\\
&=\; \frac{1}{\sqrt{PN}}\sum_{d=-\rho D}^{\rho D} e^{j\frac{2\pi}{PN}nd}[\boldsymbol{F}\boldsymbol{a}_d]_{\langle m-n\rangle_{PN}},
\end{aligned}$$

where we used the substitution $d = k-l$. □

From (24), (37), (38), and Lemma 1

$$Q_{n,m} \;=\; \frac{1}{\sqrt{PN}}\sum_{d=-2D}^{2D} e^{j\frac{2\pi}{PN}nd}[\boldsymbol{\alpha}_d]_{\langle m-n\rangle_{PN}} \tag{45}$$

where $\boldsymbol{\alpha}_d := \boldsymbol{F}\,\mathrm{diag}_d(\boldsymbol{G})$. With $\alpha_{d,m} := [\boldsymbol{\alpha}_d]_m$, we find

$$\begin{aligned}
y_k \;&=\; \sum_{n=0}^{PN-1}|Q_{n,k}|^2 v_n\\
&=\; \frac{1}{PN}\sum_{n=0}^{PN-1} v_n \sum_{d,l=-2D}^{2D} e^{-j\frac{2\pi}{PN}n(l-d)}\alpha_{d,\langle k-n\rangle_{PN}}\alpha^*_{l,\langle k-n\rangle_{PN}}\\
&=\; \frac{1}{PN}\sum_{m=0}^{PN-1} v_{\langle k-m\rangle_{PN}} \sum_{d,l=-2D}^{2D} e^{-j\frac{2\pi}{PN}(l-d)(k-m)}\alpha_{d,m}\alpha^*_{l,m} \qquad (46)
\end{aligned}$$

where, for (46), $m = \langle k-n\rangle_{PN}$ so that $n = \langle k-m\rangle_{PN}$. Defining the matrix $\boldsymbol{D}_k := \mathcal{D}(\boldsymbol{F}\boldsymbol{i}_k)$ and the vector $\boldsymbol{\beta}(d,l)$ such that $[\boldsymbol{\beta}(d,l)]_m = \alpha_{d,m}\alpha^*_{l,m}$,

$$\boldsymbol{y} \;=\; \sum_{d,l=-2D}^{2D}\boldsymbol{D}_{l-d}\mathcal{C}(\boldsymbol{v})\boldsymbol{D}^H_{l-d}\boldsymbol{\beta}(d,l) \tag{47}$$

Using the property $\boldsymbol{D}_k\boldsymbol{F} = \frac{1}{\sqrt{PN}}\boldsymbol{F}\boldsymbol{T}_k$,

$$\begin{aligned}\boldsymbol{D}_{l-d}\,\mathcal{C}(\boldsymbol{v})\boldsymbol{D}_{l-d}^H &= \sqrt{PN}\boldsymbol{D}_{l-d}\boldsymbol{F}\,\mathcal{D}(\boldsymbol{F}^H\boldsymbol{v})\boldsymbol{F}^H\boldsymbol{D}_{l-d}^H \\ &= \frac{1}{\sqrt{PN}}\boldsymbol{F}\boldsymbol{T}_{l-d}\,\mathcal{D}(\boldsymbol{F}^H\boldsymbol{v})\boldsymbol{T}_{l-d}^H\boldsymbol{F}^H \\ &= \frac{1}{\sqrt{PN}}\boldsymbol{F}\,\mathcal{D}\left(\boldsymbol{T}_{l-d}\boldsymbol{F}^H\boldsymbol{v}\right)\boldsymbol{F}^H. \end{aligned} \tag{48}$$

Substituting (48) into (47) yields (36).

B. Derivation of (39)

Here we derive an expression for $\boldsymbol{z}^{(n)}$ enabling fast computation. Without loss of generality, we omit superscripts in this appendix. From the definition of $\bar{\boldsymbol{g}}_k$ we notice $\boldsymbol{C}_k^H\boldsymbol{g}_k = \boldsymbol{C}^H\bar{\boldsymbol{g}}_k$, and thus, with (11), (25), and (41), we have $\boldsymbol{P} = \boldsymbol{C}^H\sum_k \bar{\boldsymbol{g}}_k\boldsymbol{i}_k^H\boldsymbol{F} = \boldsymbol{C}^H\bar{\boldsymbol{G}}\boldsymbol{F} = \mathcal{D}(\boldsymbol{b}^*)\boldsymbol{J}^H\boldsymbol{F}^H\bar{\boldsymbol{G}}\boldsymbol{F}$. Then we can write $z_k = \|\boldsymbol{p}_k\|^2 = \sum_{n=0}^{N_b-1}|b_n^*[\boldsymbol{F}^H\bar{\boldsymbol{G}}\boldsymbol{F}]_{\langle n\rangle_{PN},k}|^2$. Since $\bar{\boldsymbol{G}}$ is banded with $2D{+}1$ non-zero diagonals, Lemma 1 implies $[\boldsymbol{F}^H\bar{\boldsymbol{G}}\boldsymbol{F}]_{\langle n\rangle_{PN},k} = \frac{1}{\sqrt{PN}}\sum_{d=-D}^{D} e^{j\frac{2\pi}{PN}nd}\bar{\alpha}_{d,\langle k-n\rangle_{PN}}$ for $\bar{\alpha}_{d,m} := [\bar{\boldsymbol{\alpha}}_d]_m$. Thus

$$\begin{aligned} z_k &= \frac{1}{PN}\sum_{n=0}^{N_b-1}\left|b_n^*\sum_{d=-D}^{D} e^{j\frac{2\pi}{PN}nd}\bar{\alpha}_{d,\langle k-n\rangle_{PN}}\right|^2 \\ &= \frac{1}{PN}\sum_{n=0}^{N_b-1}|b_n|^2\sum_{d,l=-D}^{D} e^{-j\frac{2\pi}{PN}n(l-d)}\bar{\alpha}_{d,\langle k-n\rangle_{PN}}\bar{\alpha}^*_{l,\langle k-n\rangle_{PN}} \\ &= \frac{1}{PN}\sum_{m=0}^{PN-1}\bar{b}_{\langle k-m\rangle_{PN}}\sum_{d,l=-D}^{D} e^{-j\frac{2\pi}{PN}(l-d)(k-m)}\bar{\alpha}_{d,m}\bar{\alpha}^*_{l,m} \end{aligned}$$

where $\bar{b}_m := \sum_{n=0}^{N_b-1}|b_{\langle n\rangle_{PN}}|^2\delta_{\langle n\rangle_{PN}-m}$. Using $\boldsymbol{D}_k$ from Appendix II-A, and defining $\bar{\boldsymbol{\beta}}(d,l)$ such that $[\bar{\boldsymbol{\beta}}(d,l)]_m = \bar{\alpha}_{d,m}\bar{\alpha}^*_{l,m}$, we find that

$$\boldsymbol{z} = \sum_{d,l=-D}^{D}\boldsymbol{D}_{l-d}\,\mathcal{C}(\bar{\boldsymbol{b}})\boldsymbol{D}_{l-d}^H\bar{\boldsymbol{\beta}}(d,l) \tag{49}$$

where $[\bar{\boldsymbol{b}}]_m = \bar{b}_m$. Similar to Appendix II-A, we substitute $\boldsymbol{D}_{l-d}\,\mathcal{C}(\bar{\boldsymbol{b}})\boldsymbol{D}_{l-d}^H = \frac{1}{\sqrt{PN}}\boldsymbol{F}\,\mathcal{D}(\boldsymbol{T}_{l-d}\boldsymbol{F}^H\bar{\boldsymbol{b}})\boldsymbol{F}^H$ into (49) to get (39).

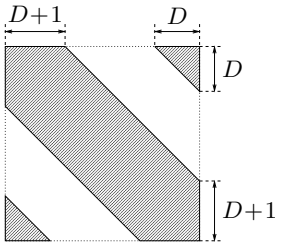

Fig. 1. Desired "banded" structure of matrix $\boldsymbol{\mathcal{H}}(i,0)$.

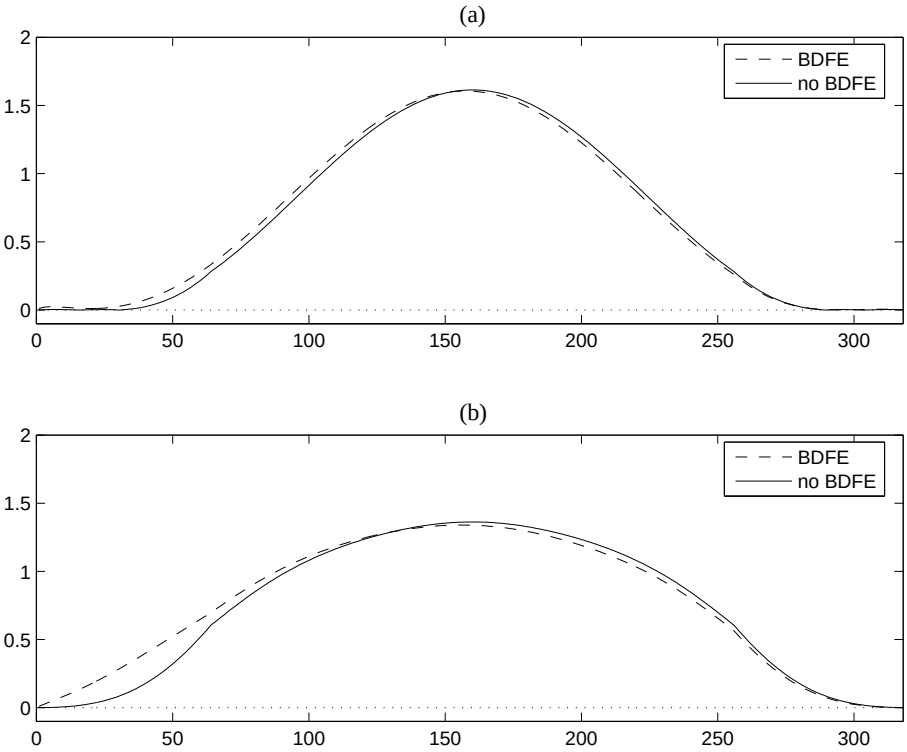

Fig. 2. Example window shapes for $PN = 256$, $N_h = 64$, SNR=10dB and (a) $f_{\mathrm{d}}T_{\mathrm{s}} = 0.001$, (b) $f_{\mathrm{d}}T_{\mathrm{s}} = 0.0075$.

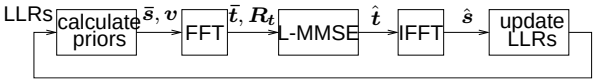

Fig. 3. Intraframe interference cancellation.

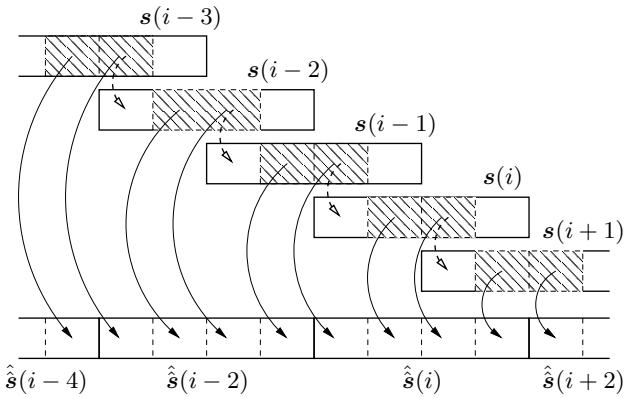

Fig. 4. Interframe detection process for $P = 2$. Solid arrows pass final hard estimates; dashed arrows pass soft initializations.

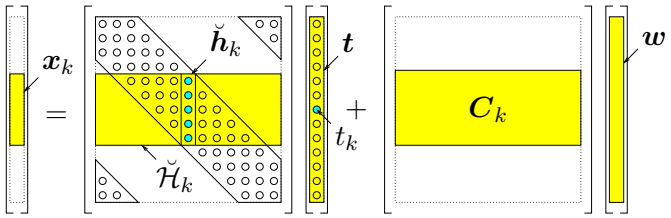

Fig. 5. Truncated observation model.

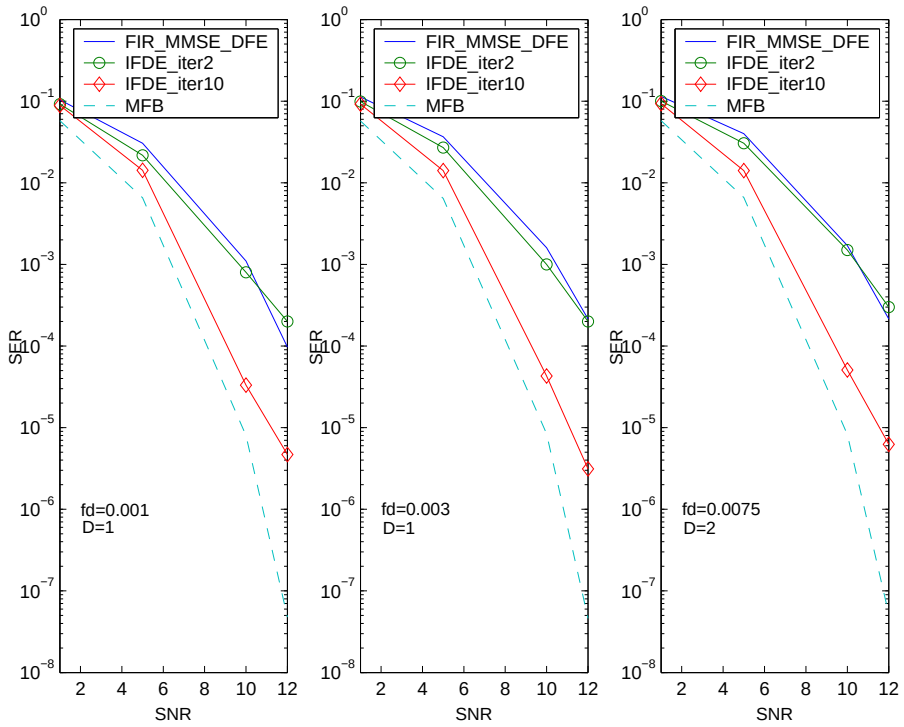

Fig. 6. Symbol error rate versus SNR for $N_h = 64$ and various $f_\mathrm{d}T_\mathrm{s}$.

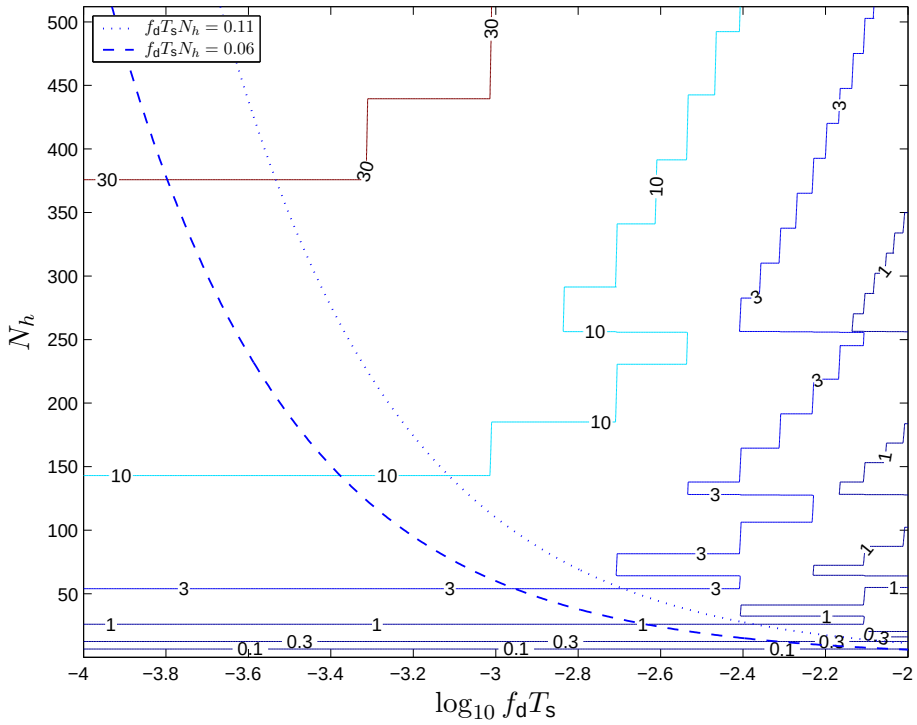

Fig. 7. Computational complexity ratio of FIR-MMSE-DFE to IFDE-BDFE-2.

$L_l^{(0)} = 0 \ \forall l$

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

  for $l = 0 \ldots PN-1$,

    $\bar{s}_l^{(n)} = \tanh(L_l^{(n)}/2)$

    $v_l^{(n)} = 1 - (\bar{s}_l^{(n)})^2$

  end

  $\bar{\boldsymbol{t}}^{(n)} = \boldsymbol{F}\bar{\boldsymbol{s}}^{(n)}$

  for $k = 0 \ldots PN-1$,

    $\boldsymbol{g}_k^{(n)} = \left(\boldsymbol{\mathcal{H}}_k \boldsymbol{F}\, \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{F}^H \boldsymbol{\mathcal{H}}_k^H + \sigma^2 \boldsymbol{C}_k \boldsymbol{C}_k^H\right)^{-1}$
    $\times \boldsymbol{\mathcal{H}}_k \boldsymbol{F}\, \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{F}^H \boldsymbol{i}_k$

    $\hat{t}_k^{(n)} = \bar{t}_k^{(n)} + \boldsymbol{g}_k^{(n)H}(\boldsymbol{x}_k - \boldsymbol{\mathcal{H}}_k \bar{\boldsymbol{t}}^{(n)})$

  end

  $\hat{\boldsymbol{s}}^{(n)} = \boldsymbol{F}^H \hat{\boldsymbol{t}}^{(n)}$

  $\boldsymbol{Q}^{(n)} = \boldsymbol{F}^H \left(\sum_{k=0}^{PN-1} \boldsymbol{\mathcal{H}}_k^H \boldsymbol{g}_k^{(n)} \boldsymbol{i}_k^H\right) \boldsymbol{F}$

  $\boldsymbol{P}^{(n)} = \left(\sum_{k=0}^{PN-1} \boldsymbol{C}_k^H \boldsymbol{g}_k^{(n)} \boldsymbol{i}_k^H\right) \boldsymbol{F}$

  for $l = 0 \ldots PN-1$,

    $$L_l^{(n+1)} = L_l^{(n)} + \frac{4\left(\mathrm{Re}\{Q_{l,l}^{(n)}(\hat{s}_l^{(n)} - \bar{s}_l^{(n)})\} + |Q_{l,l}^{(n)}|^2 \bar{s}_l^{(n)}\right)}{\boldsymbol{q}_l^{(n)H}\, \mathcal{D}(\boldsymbol{v}^{(n)}) \boldsymbol{q}_l^{(n)} - |Q_{l,l}^{(n)}|^2 v_l^{(n)} + \sigma^2 \|\boldsymbol{p}_l\|^2}$$

  end

end

TABLE I

SUMMARY OF ITERATIVE SYMBOL DETECTION.

	Step	Cost Per Step
1	$\boldsymbol{H}(i,\ell) = \boldsymbol{F}\boldsymbol{J}\,\mathcal{D}(\boldsymbol{b})\overline{\boldsymbol{H}}(i,\ell)\boldsymbol{F}$ for appropriate ℓ	$PNN_h + (N_h + \widetilde{D})PN\log_2 PN$
2	compute $\boldsymbol{x}(i)$ from (35)	$(\widetilde{D}L_{\text{pst}} + 1)PN + \frac{3}{2}PN\log_2 PN$
3	$\boldsymbol{\Sigma} = \sqrt{PN}\boldsymbol{C}_0\boldsymbol{C}_0^H$	$2PN + \frac{1}{2}PN\log_2 PN$
	$L_l^{(0)} = 0\ \forall l$	0
	for $n = 0, 1, \ldots, M-1$	
	for $l = 0, 1, \ldots, PN-1$	
	$\bar{s}_l^{(n)} = \tanh(L_l^{(n)}/2)$	1
	$v_l^{(n)} = 1 - (\bar{s}_l^{(n)})^2$	1
	end	
	$\bar{\boldsymbol{t}}^{(n)} = \boldsymbol{F}\bar{\boldsymbol{s}}^{(n)}$	$\frac{1}{2}PN\log_2 PN$
	$\boldsymbol{u}^{(n)} = \boldsymbol{F}\boldsymbol{v}^{(n)}$	$\frac{1}{2}PN\log_2 PN$
4	$\boldsymbol{R}_0^{(n)} = \boldsymbol{\mathcal{H}}_0\,\mathcal{C}(\boldsymbol{u}^{(n)})\boldsymbol{\mathcal{H}}_0^H + \boldsymbol{\Sigma}$	$3\widetilde{D}^3$
	for $k = 0, 1, \ldots, PN-1$	
5	compute $(\boldsymbol{R}_k^{(n)})^{-1}$	$\min\left\{2\widetilde{D}^2 + \frac{1}{3}\widetilde{D}^3,\ 7\widetilde{D}^2\right\}$
6	$\boldsymbol{g}_k^{(n)} = (\boldsymbol{R}_k^{(n)})^{-1}\boldsymbol{\mathcal{H}}_k\,\mathcal{C}(\boldsymbol{u}^{(n)})\boldsymbol{i}_k$	$2\widetilde{D}^2$
	$\hat{t}_k^{(n)} = \bar{t}_k^{(n)} + \boldsymbol{g}_k^{(n)H}(\boldsymbol{x}_k - \boldsymbol{\mathcal{H}}_k\bar{\boldsymbol{t}}^{(n)})$	$\widetilde{D}^2$
	end	
	$\hat{\boldsymbol{s}}^{(n)} = \boldsymbol{F}^H\hat{\boldsymbol{t}}^{(n)}$	$PN\log_2 PN$
7	compute $\{y_l^{(n)}\}_{l=0}^{PN-1}, \{\boldsymbol{\alpha}_d\}_{d=-2D}^{2D}$ from (36)-(38)	$5\widetilde{D}^2PN + (4\widetilde{D}^2 + 2\widetilde{D})PN\log_2 PN$
8	compute $\{z_l^{(n)}\}_{l=0}^{PN-1}$ from (39)-(41)	$\widetilde{D}^2PN + (\widetilde{D}^2 + \widetilde{D})PN\log_2 PN$
	for $l = 0, 1, \ldots, PN-1$	
9	$Q_{l,l}^{(n)} = \frac{1}{\sqrt{PN}}\sum_{d=-2D}^{2D}[\boldsymbol{\alpha}_d^{(n)}]_0\, e^{j\frac{2\pi}{PN}ld}$	$2\widetilde{D}$
	$L_l^{(n+1)} = L_l^{(n)} + 4\frac{\text{Re}\{Q_{l,l}^{(n)}(\hat{s}_l^{(n)} - \bar{s}_l^{(n)})\} + \lvert Q_{l,l}^{(n)}\rvert^2\bar{s}_l^{(n)}}{y_l^{(n)} - \lvert Q_{l,l}^{(n)}\rvert^2 v_l^{(n)} + \sigma^2 z_l^{(n)}}$	6
	end	
	end	

TABLE II

FAST IMPLEMENTATION OF THE ITERATIVE SYMBOL DETECTOR

IFDE-noBDFE: $3M\widetilde{D}^3/N + \left[2 + (N_h + \widetilde{D}) + M(5\widetilde{D}^2 + 3\widetilde{D} + 2)\right] P\log_2 PN + \left[3 + N_h + M\left(\min\{\frac{1}{3}\widetilde{D}^3, 5\widetilde{D}^2\} + 11\widetilde{D}^2 + 2\widetilde{D} + 8\right)\right] P$
IFDE-BDFE: $3M\widetilde{D}^3/N + \left[2 + (L_{\text{pst}} + 1)(N_h + \widetilde{D}) + M(5\widetilde{D}^2 + 3\widetilde{D} + 2)\right] P\log_2 PN + \left[3 + L_{\text{pst}}(\widetilde{D} + N_h) + N_h + M\left(\min\{\frac{1}{3}\widetilde{D}^3, 5\widetilde{D}^2\} + 11\widetilde{D}^2 + 2\widetilde{D} + 8\right)\right] P$
FIR-MMSE-DFE: $\frac{9}{2}N_f^2 - \frac{1}{2}N_f - 1$

TABLE III

RELATIVE ALGORITHM COMPLEXITY (PER SYMBOL).

Step	Cost
$\begin{bmatrix} \alpha_k & \boldsymbol{a}_k^H \\ \boldsymbol{a}_k & \boldsymbol{A}_k \end{bmatrix} = \boldsymbol{R}_k^{-1}$	0
$\boldsymbol{\Theta}_k^{-1} = \boldsymbol{A}_k - \alpha_k^{-1}\boldsymbol{a}_k\boldsymbol{a}_k^H$	$\widetilde{D}^2$
compute $\tilde{\boldsymbol{\theta}}_k$ and $\tilde{\theta}_k$	$3\widetilde{D}^2$
$\boldsymbol{b}_k = -\boldsymbol{\Theta}_k^{-1}\tilde{\boldsymbol{\theta}}_k$	$\widetilde{D}^2$
$\beta_k = \left(\tilde{\theta}_k - \tilde{\boldsymbol{\theta}}_k^H\boldsymbol{\Theta}_k^{-1}\tilde{\boldsymbol{\theta}}_k\right)^{-1}$	$\widetilde{D}^2$
$\boldsymbol{R}_{k+1}^{-1} = \begin{bmatrix} \boldsymbol{\Theta}_k^{-1} + \boldsymbol{b}_k\boldsymbol{b}_k^H\beta_k & \boldsymbol{b}_k\beta_k \\ \boldsymbol{b}_k^H\beta_k & \beta_k \end{bmatrix}$	$\widetilde{D}^2$

TABLE IV

RECURSIVE UPDATE OF $(\boldsymbol{R}_k^{(n)})^{-1}$