

Computationally Efficient Equalization for Asynchronous Cooperative Communications with Multiple Frequency Offsets

Huiming Wang[†], Xiang-Gen Xia[‡], and Qinye Yin[†]

Abstract

In cooperative communications, time and frequency synchronization is an important issue needed to be addressed in practice. Due to the nature of cooperative communications, multiple frequency offsets may occur and the traditional frequency offset compensations may not apply. For this problem, equalization for the time-varying channel has been used in the literature, where the equalization matrix inverse needs to be retaken every symbol. In this paper, we propose computationally efficient minimum mean square error (MMSE) and MMSE decision feedback equalizers (MMSE-DFE) when multiple frequency offsets are present, where the equalization matrix inverses do not need to be retaken every symbol. Our proposed equalization methods apply to linear convolutively coded cooperative systems, where linear convolutive space-time coding is used to achieve the full cooperative diversity when there are timing errors from the cooperative users or relay nodes, i.e., asynchronous cooperative communication systems.

Index Terms

Asynchronous cooperative networks, equalization, MMSE, MMSE-DFE, multiple carrier frequency offsets, space-time codes

I. INTRODUCTION

Cooperative communications have attracted considerable attention lately due to the potential *cooperative diversity* [1] - [4]. The basic idea to achieve the cooperative diversity is similar to achieving the spatial diversity in multiple antenna systems by utilizing the multiple transmissions and possibly space-time coding. It is well-understood that a major difference between cooperative and multiple antenna systems is that the multiple transmissions in cooperative systems may not be either time or frequency synchronized since the multiple transmissions are from different user or relay node locations while they are co-located in conventional multiple antenna systems.

[†]Huiming Wang and Qinye Yin are with the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, P. R. China. Email: xjbswhm@163.com. Their work was partially supported by the National Natural Science Foundation (No. 60572046 & No. 60502022 & No. 60772095) and the National Hi-Tech Research & Development Program (No. 2006AA01Z220) of China. This work was done when Huiming Wang was visiting the University of Delaware.

[‡]Xiang-Gen Xia is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716, USA. Email: xxia@ee.udel.edu. His work was supported in part by the Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-05-1-0161 and the National Science Foundation under Grant CCR-0325180.

When both time and frequency are synchronized at the relay nodes, the existing space-time coding for multiple antenna systems can be directly applied to cooperative communication systems adopting decode-and-forward (DF) protocol to achieve the cooperative diversity¹. When there are timing offsets from relay nodes, there have been several studies in the literature to achieve the asynchronous cooperative diversity, see for example [5] - [15]. In [7], a family of space-time trellis codes achieving full asynchronous cooperative diversity was proposed, which is based on the Hammons-El Gamal's nonlinear algebraic space-time codes [19]. This code family was systematically studied and constructed in [8]. In [15], a linear construction called *distributed linear convolutive space-time codes* (DLC-STC), which is also based on [7], has been proposed and shown to achieve the full asynchronous cooperative diversity not only with the maximum-likelihood (ML) receiver but also with the minimum mean square error (MMSE) equalizer² and MMSE decision feedback equalizer (MMSE-DFE). When the channels from relay nodes to destination node are frequency-selective fading, a distributed high-rate space-frequency code achieving both full asynchronous cooperative diversity and full multipath diversity was proposed in [13] by generalizing the idea of the OFDM transmission first proposed in [12] to achieve the full asynchronous cooperative diversity with Alamouti code.

All the aforementioned schemes are proposed to deal with the time asynchronous issue. On the other hand, due to the reason that the multiple transmissions from relay nodes are from different locations with different oscillators, they may have multiple different carrier frequency offsets (CFOs) that can not be compensated simultaneously at the receiver as in a conventional multiple antenna system. With multiple CFOs and different propagation delays, the channel becomes time-varying with intersymbol interference (ISI). To deal with this problem, equalizations have been proposed in the literature, see for example [16], [17], [18], where the equalization matrix inverses need to be retaken for every symbol (or OFDM symbol) even in a channel coherent time duration.

In this paper, we consider the equalization issue for cooperative communication systems with multiple CFOs. We propose computationally efficient MMSE and MMSE-DFE equalizers when multiple frequency offsets are present, where the equalization matrix inverses do not need to be retaken every symbol in a channel coherent time duration, which may therefore significantly reduce the computational

¹Space-time codes for conventional multiple antenna systems may not be applied directly in amplify-and-forward based cooperative systems, see [4].

²Space-time codes achieving full spatial diversity with linear receivers have been recently studied in [20], [21], [22].

complexity for the equalization. Our proposed equalization methods apply to linear convolutively coded cooperative systems, where the DLC-STC [15] can be used to achieve full asynchronous cooperative diversity when there are timing errors from cooperative users or relay nodes. Our proposed equalization methods also apply to frequency-selective fading channels (from relay nodes to destination node).

This paper is organized as follows. In Section II, we present the cooperative communication system model we use in this paper. At the transmitters (relay nodes), we adopt the notion of a general DLC-STC. In Section III, we derive linear MMSE equalizer and MMSE-DFE equalizer and propose recursive methods to equalize the received data with multiple different CFOs. We then study some properties of the proposed equalizers. In Section IV, we analyze the computational complexities of the proposed equalization schemes. In Section V, we present some simulation results.

Notations: Superscripts $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^H$ represent conjugate, transpose, and Hermitian, respectively. $E(\cdot)$ is the mathematical expectation. $\text{diag}(\cdot)$ is diagonal matrix with main diagonal (\cdot) . $|\cdot|$ and $\|\cdot\|$ are the modulus of a complex scalar and Euclidean norm of a vector, respectively. We denote the $N \times N$ identity matrix as \mathbf{I}_N and the $M \times N$ all-zero matrix as $\mathbf{0}_{M \times N}$. Matlab matrix representations are used, for example, $\mathbf{X}(r_1 : r_2, c_1 : c_2)$ denotes the submatrix of \mathbf{X} of the rows from r_1 to r_2 and the columns from c_1 to c_2 . \mathbb{C} is the complex number field. Function $Q(\cdot)$ represents a decision device that optimally maps soft symbol estimates into hard symbol estimates.

II. SYSTEM MODEL

The cooperative communication system considered in this paper is shown in Fig.1, where there are R relay nodes between the source node and the destination node. The DF protocol [2] is adopted. In the first phase, the source broadcasts the information sequence to potential relay nodes. At the relay node, the received signals are decoded and only the relay nodes that correctly decode the received signals will become active relay nodes to participate in the cooperative transmissions in the second phase. In the beginning of the second phase, the eligible relay nodes re-map the decoded information bits into symbols. Without loss of generality, we assume all R eligible relay nodes use the same signal constellation Γ , such as QPSK or QAM. Therefore, all the symbol sequences re-mapped at relays are the same, which can be denoted as $\mathbf{s} = [s_0, s_1, \dots, s_{N-1}]$. Consider a general distributed linear convolutive coding scheme for the R relay nodes. At the r -th relay node, the information symbol

sequence \mathbf{s} is linearly transformed into $\bar{\mathbf{c}}_r$ by an $N \times (N + L - 1)$ generating matrix $\bar{\mathbf{T}}^{(r)}$, i.e., $\bar{\mathbf{c}}_r = \mathbf{s}\bar{\mathbf{T}}^{(r)}$, where $\bar{\mathbf{T}}^{(r)}$, $r = 1, 2, \dots, R$, are Toeplitz matrices in the form:

$$\bar{\mathbf{T}}^{(r)} = \begin{bmatrix} \bar{t}_1^{(r)} & \bar{t}_2^{(r)} & \cdots & \bar{t}_L^{(r)} & 0 & 0 & \cdots & 0 \\ 0 & \bar{t}_1^{(r)} & \bar{t}_2^{(r)} & \cdots & \bar{t}_L^{(r)} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \bar{t}_1^{(r)} & \bar{t}_2^{(r)} & \cdots & \bar{t}_L^{(r)} \end{bmatrix} \quad (1)$$

where elements $\bar{t}_l^{(r)}$, $r = 1, 2, \dots, R$, $l = 1, 2, \dots, L$, are the generating polynomial coefficients for the linear convolutive code and can be designed to satisfy, for example, the *shift-full-rank property* [8] so as to achieve the full asynchronous cooperative diversity as studied in [15]. We would like to emphasize here that our following proposed recursive equalizers are independent of the above generating polynomial coefficients.

For convenience, in what follows we assume that channels from relay nodes to destination node are flat-fading and time-invariant during the transmission of one block. In fact, as we shall see later our proposed methods apply to frequency-selective channels as well. Denote the delays of R relay nodes as τ_r , $r = 1, 2, \dots, R$, respectively, and we only consider the case when the delay is a multiple of symbol duration. Without loss of generality, we assume $\tau_1 = 0$ and $\tau_1 < \tau_2 < \dots < \tau_R$. To deal with the timing errors, at the beginning and/or the end of each block, guard interval is inserted so that adjacent code blocks will not overlap with each other [7], [9], [10]. Therefore, we have $\delta \geq \tau_R$, where δ is the interval padding length. Here we assume $\delta = \tau_R$. Due to the zero padding operations at the relay nodes and the different delays between relay and destination nodes, the *equivalent code matrix* at the destination node can be written as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_R \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{c}}_1 & \overbrace{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \cdots \ 0}^{\tau_R} \\ \overbrace{0 \ \cdots \ 0}^{\tau_2} & \bar{\mathbf{c}}_2 & \overbrace{0 \ 0 \ \cdots \ 0}^{\tau_R - \tau_2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \overbrace{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \cdots \ 0}^{\tau_R} & \bar{\mathbf{c}}_R \end{bmatrix} \quad (2)$$

where $\mathbf{c}_r \triangleq [\mathbf{0}_{1 \times \tau_r} \ \bar{\mathbf{c}}_r \ \mathbf{0}_{1 \times (\tau_R - \tau_r)}]$, $r = 1, 2, \dots, R$. From (1), \mathbf{c}_r can be rewritten as:

$$\mathbf{c}_r = \mathbf{s}\mathbf{T}^{(r)} \quad (3)$$

where $\mathbf{T}^{(r)}$ is an $N \times (N + L - 1 + \tau_R)$ matrix defined as $\mathbf{T}^{(r)} \triangleq [\mathbf{0}_{N \times \tau_r} \ \bar{\mathbf{T}}^{(r)} \ \mathbf{0}_{N \times (\tau_R - \tau_r)}]$. It is a zero-padded version of $\bar{\mathbf{T}}^{(r)}$, which can be called as *equivalent generating matrix*.

Denote the channel coefficient and the CFO between the destination and the r -th relay node as h_r and Δf_r , respectively. Then the received signal at destination node in symbol interval k is:

$$y_k = \sum_{r=1}^R e^{j2\pi kT\Delta f_r} h_r c_{r,k} + n_k = \sum_{r=1}^R e^{j2\pi kT\Delta f_r} h_r \sum_{n=0}^{N-1} s_n t_{n,k}^{(r)} + n_k \quad (4)$$

where T is the symbol duration, $c_{r,k}$ is the k -th element of \mathbf{c}_r and $t_{n,k}^{(r)}$ is the (n, k) -th element of $\mathbf{T}^{(r)}$ in (3). The additive noise n_k is independent identically distributed (i.i.d.) complex Gaussian with zero-mean and variance σ_n^2 .

Reformulating (4) into vector-matrix form yields:

$$y_k = \mathbf{e}_k \bar{\mathbf{h}} \bar{\mathbf{s}}_k + n_k \quad (5)$$

where $\mathbf{e}_k = [e^{j2\pi kT\Delta f_1}, e^{j2\pi kT\Delta f_2}, \dots, e^{j2\pi kT\Delta f_R}]$ is CFO vector, $\bar{\mathbf{h}}$ is a $R \times (L + \tau_R)$ matrix as:

$$\bar{\mathbf{h}} = \begin{bmatrix} h_1 \bar{t}_1^{(1)} & h_1 \bar{t}_2^{(1)} & \dots & h_1 \bar{t}_L^{(1)} & \overbrace{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \dots \ 0}^{\tau_R} \\ \overbrace{0 \ \dots \ 0}^{\tau_2} & h_2 \bar{t}_1^{(2)} & h_2 \bar{t}_2^{(2)} & \dots & h_2 \bar{t}_L^{(2)} & \overbrace{0 \ 0 \ \dots \ 0}^{\tau_R - \tau_2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \overbrace{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \dots \ 0}^{\tau_R} & h_R \bar{t}_1^{(R)} & h_R \bar{t}_2^{(R)} & \dots & h_R \bar{t}_L^{(R)} \end{bmatrix} \quad (6)$$

and $\bar{\mathbf{s}}_k = [s_k, s_{k-1}, \dots, s_{k-(L+\tau_R)+1}]^T$.

Let $L' = L + \tau_R$. Considering consecutive N_f symbols, we get:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{s}_k + \mathbf{n}_k \quad (7)$$

where:

$$\mathbf{H}_k = \mathbf{P}_k \bar{\mathbf{H}} \quad (8)$$

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{e}_k & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} \\ \mathbf{0}_{1 \times R} & \mathbf{e}_{k-1} & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} & \mathbf{e}_{k-N_f+1} \end{bmatrix}_{N_f \times N_f R} \quad (9)$$

$$\bar{\mathbf{H}} = \begin{bmatrix} \bar{\mathbf{h}} & \mathbf{0}_{R \times 1} & \mathbf{0}_{R \times 1} & \mathbf{0}_{R \times 1} \\ \mathbf{0}_{R \times 1} & \bar{\mathbf{h}} & \mathbf{0}_{R \times 1} & \mathbf{0}_{R \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{R \times 1} & \mathbf{0}_{R \times 1} & \mathbf{0}_{R \times 1} & \bar{\mathbf{h}} \end{bmatrix}_{N_f R \times (N_f + L' - 1)} \quad (10)$$

$\mathbf{y}_k = [y_k, y_{k-1}, \dots, y_{k-N_f+1}]^T$, $\mathbf{s}_k = [s_k, s_{k-1}, \dots, s_{k-(N_f+L'-1)+1}]^T$, $\mathbf{n}_k = [n_k, n_{k-1}, \dots, n_{k-N_f+1}]^T$.

Eq. (7) is an equivalent received signal model of the distributed MISO system. We can find that the $N_f \times (N_f + L' - 1)$ matrix \mathbf{H}_k , which can be called the *equivalent channel matrix*, is composed of two parts. The first part is the $N_f \times N_f R$ matrix \mathbf{P}_k that is only associated with CFOs and is time-varying. The second part is the $N_f R \times (N_f + L' - 1)$ matrix $\bar{\mathbf{H}}$, which does not change in the block and is related to the channel coefficients h_r and the equivalent generating matrices $\mathbf{T}^{(r)}$, $r = 1, 2, \dots, R$, of the DLC-STC. Note that each row of \mathbf{H}_k only contains L' non-zero elements, which means that the length of the equivalent channel is L' . The received signal at symbol interval k can be considered as a linear convolution of the transmitted block s_k with the equivalent channel. In the following, we shall develop recursive equalizers for the signal model (7). From this model, one can see that, if the channels from relay nodes to destination node are not flat fading but frequency-selective fading, a similar model to (7) can be derived by changing the part in $\bar{\mathbf{H}}$ and all the following studies follow too. In other words, our proposed equalizers also work for frequency-selective fading channels.

III. RECURSIVE EQUALIZER DESIGNS

In this section, we assume that at the destination node, the channel information is known, including the delays τ_i , CFOs Δf_i , and the channel coefficients h_i , $i = 1, 2, \dots, R$, and we will derive our recursive equalization algorithms for the destination node based on (7). Different from the MMSE/MMSE-DFE equalizers for time-invariant channels [23], [24], ours are for time-varying channels. However, the key observation is that the varying of the channel is only due to the CFO matrix \mathbf{P}_k in (9), so the channels in two adjacent symbol intervals have some special relationship that we can fully take advantage of, in a way that the design result at symbol interval k can be exploited to design the equalizer at symbol interval $k + 1$ so that a new matrix inversion is not necessary.

A. MMSE Equalizer

At the receiver, we use an MMSE equalizer to compensate the channel and the CFOs. The finite length linear MMSE equalizer is a finite impulse response (FIR) filter with order N_f to minimize the mean square error at symbol interval k as:

$$\mathbf{f}_k^{MMSE} = \arg \min_{\mathbf{f}_k \in \mathbb{C}^{N_f \times 1}} E |s_{k-D} - \mathbf{f}_k^H \mathbf{y}_k|^2 \quad (11)$$

where \mathbf{f}_k is the filter coefficient vector and D is the estimation delay to make the filter a causal system, which satisfies $0 \leq D \leq (N_f + L' - 2)$. Here, we let $D = (N_f - 1)/2$. Assume the symbol sequence

and the additive noise are wide-sense stationary, mutually uncorrelated and white with variance σ_s^2 and σ_n^2 , respectively. The solution for (11) is well known [23]:

$$\mathbf{f}_k^{MMSE} = (\mathbf{H}_k \mathbf{H}_k^{\mathcal{H}} + c \mathbf{I}_{N_f})^{-1} \mathbf{H}_k \mathbf{i}_D \quad (12)$$

where the constant $c = \sigma_n^2 / \sigma_s^2$ and \mathbf{i}_D is an $(N_f + L' - 1) \times 1$ vector with 1 at the $(D + 1)$ -th element and all 0's elsewhere.

As we mentioned before, (12) is a time-varying filter due to the CFO matrix \mathbf{P}_k in (9). It seems that matrix inversion operation should be taken every symbol interval to calculate (12) and then to detect s_{k-D} by $\hat{s}_{k-D} = Q(\mathbf{f}_k^{\mathcal{H}} \mathbf{y}_k)$. However, we will show below that it is unnecessary due to the special time-varying property of the CFO matrix \mathbf{P}_k .

Denote $\mathbf{R}_k = \mathbf{H}_k \mathbf{H}_k^{\mathcal{H}} + c \mathbf{I}_{N_f}$. At time interval k and $k + 1$, \mathbf{R}_k and \mathbf{R}_{k+1} can be partitioned as:

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{\Xi} & \mathbf{r}_k \\ \mathbf{r}_k^{\mathcal{H}} & r_k \end{bmatrix}, \quad \mathbf{R}_{k+1} = \begin{bmatrix} r_{k+1} & \mathbf{r}_{k+1}^{\mathcal{H}} \\ \mathbf{r}_{k+1} & \mathbf{\Xi} \end{bmatrix} \quad (13)$$

where $\mathbf{\Xi}$ is an $(N_f - 1) \times (N_f - 1)$ sub-matrix, \mathbf{r}_k and \mathbf{r}_{k+1} are $(N_f - 1) \times 1$ column vectors. r_k and r_{k+1} are scalars. Assume that we have obtained \mathbf{R}_k^{-1} , which can be partitioned as:

$$\mathbf{R}_k^{-1} = \begin{bmatrix} \mathbf{\Theta} & \mathbf{w}_k \\ \mathbf{w}_k^{\mathcal{H}} & \theta_k \end{bmatrix} \quad (14)$$

where $\mathbf{\Theta} = \mathbf{R}_k^{-1}(1 : N_f - 1, 1 : N_f - 1)$, $\mathbf{w}_k = \mathbf{R}_k^{-1}(1 : N_f - 1, N_f)$ and $\theta_k = \mathbf{R}_k^{-1}(N_f, N_f)$. Then the following holds:

$$\mathbf{\Xi}^{-1} = \mathbf{\Theta} - \frac{(\mathbf{\Theta} \mathbf{r}_k) (\mathbf{\Theta} \mathbf{r}_k)^{\mathcal{H}}}{\nu_k^{-1} + \mathbf{r}_k^{\mathcal{H}} \mathbf{\Theta} \mathbf{r}_k} \quad (15)$$

where $\nu_k = 1/r_k$. To calculate \mathbf{R}_{k+1}^{-1} , we have the following:

$$\mathbf{R}_{k+1}^{-1} = \begin{bmatrix} \nu_{k+1} + \nu_{k+1}^2 \mathbf{r}_{k+1}^{\mathcal{H}} \mathbf{\Psi} \mathbf{r}_{k+1} & -\nu_{k+1} \mathbf{r}_{k+1}^{\mathcal{H}} \mathbf{\Psi} \\ -\nu_{k+1} \mathbf{\Psi} \mathbf{r}_{k+1} & \mathbf{\Psi} \end{bmatrix} \quad (16)$$

where $\nu_{k+1} = 1/r_{k+1}$ and

$$\mathbf{\Psi} = \mathbf{\Xi}^{-1} + \frac{(\mathbf{\Xi}^{-1} \mathbf{r}_{k+1}) (\mathbf{\Xi}^{-1} \mathbf{r}_{k+1})^{\mathcal{H}}}{\nu_{k+1}^{-1} - \mathbf{r}_{k+1}^{\mathcal{H}} \mathbf{\Xi}^{-1} \mathbf{r}_{k+1}}. \quad (17)$$

The detailed derivation is shown in Appendix I. Hence, at symbol interval $k + 1$, we can calculate \mathbf{R}_{k+1}^{-1} through (16) and (17) using $\mathbf{\Xi}^{-1}$ obtained at symbol interval k by (15). So matrix inversion is not required anymore except in the initial step in (14). Therefore, the computational complexity is greatly reduced, which will be shown in Section IV.

We summarize the above procedure as follows:

- 1) At time interval k , assume we have already obtained \mathbf{R}_k and \mathbf{R}_k^{-1} as (13) and (14);
- 2) Calculate Ξ^{-1} as (15) ;
- 3) At time interval $k + 1$, calculate \mathbf{r}_{k+1} and r_{k+1} as (13);
- 4) Calculate Ψ as (17) using Ξ^{-1} obtained in Step 2) and \mathbf{r}_{k+1} , r_{k+1} obtained in Step 3);
- 5) Finally, substitute (17) into (16) to obtain \mathbf{R}_{k+1}^{-1} . And then design \mathbf{f}_{k+1}^{MMSE} according to (12).

B. MMSE-DFE Equalizer

The linear MMSE equalizer is the optimal linear equalizer in the sense of minimum mean square error. On the other hand, the decision feedback equalizer (DFE) is a well known nonlinear equalizer that can outperform the MMSE equalizer. In the following, we will derive finite-length DFE equalizer based on the minimum mean square error criteria, i.e., MMSE-DFE. We will show that a similar recursive procedure also exists.

Let N_f and N_b be the lengths of the feedforward and feedback FIR filters, respectively. The design of DFE based on the MMSE criteria is to design \mathbf{f}_k and \mathbf{b}_k to satisfy:

$$\{\mathbf{f}_k^{MMSE-DFE}, \mathbf{b}_k^{MMSE-DFE}\} = \arg \min_{\mathbf{f}_k \in \mathbb{C}^{N_f \times 1}, \mathbf{b}_k \in \mathbb{C}^{N_b \times 1}} E |s_{k-D} - (\mathbf{f}_k^H \mathbf{y}_k + \mathbf{b}_k^H \hat{\mathbf{s}}_{k-D-1})|^2 \quad (18)$$

where \mathbf{f}_k and \mathbf{b}_k are the feedforward and feedback filter coefficient vectors, respectively, D is the estimation delay, $\hat{\mathbf{s}}_{k-D-1} = [\hat{s}_{k-D-1}, \hat{s}_{k-D-2}, \dots, \hat{s}_{k-D-N_b}]^T$ is the vector with already detected symbols. Generally, we assume the symbols in $\hat{\mathbf{s}}_{k-D-1}$ are correctly detected to ignore the error-propagation problem (a common assumption in DFE design) [24]. Although the selection of delay D will impact the MMSE performance of the equalizer, we shall assume that $D = N_f - 1$ in this paper, since it is always the case for most practical channel and noise scenarios and reasonably long feedforward filter [24]. Also, we let $N_b = L' - 1$ since the ISI of the equivalent channel only span $L' - 1$ symbols. In Appendix II, we derive the vector-matrix form of the MMSE-DFE equalizer slightly different from that in [23], [24], [25] since it is more convenient to give the recursive algorithm based on our derivation. In this case, we get:

$$\mathbf{f}_k^{MMSE-DFE} = (\mathbf{H}_k \mathbf{W} \mathbf{H}_k^H + c \mathbf{I}_{N_f})^{-1} \mathbf{H}_k \mathbf{i}_D \quad (19a)$$

$$\mathbf{b}_k^{MMSE-DFE} = -\mathbf{U} \mathbf{H}_k^H \mathbf{f}_k^{MMSE-DFE} \quad (19b)$$

where $\mathbf{U} = \begin{bmatrix} \mathbf{0}_{N_b \times (N_f + L' - 1 - N_b)} & \mathbf{I}_{N_b} \end{bmatrix}$, $\mathbf{W} = \mathbf{I}_{N_f + L' - 1} - \mathbf{U}^H \mathbf{U}$ and $c = \sigma_n^2 / \sigma_s^2$. Similar to the case of the MMSE equalizer, both \mathbf{f}_k and \mathbf{b}_k are time-varying. At each symbol interval, matrix inversion should be taken to calculate \mathbf{f}_k and \mathbf{b}_k then to detect s_{k-D} by $\hat{s}_{k-D} = Q (\mathbf{f}_k^H \mathbf{y}_k + \mathbf{b}_k^H \hat{s}_{k-D-1})$, if the above algorithm is used directly.

Similar to the previous derivation of the recursive algorithm for the MMSE equalizer, we also let $\mathbf{R}_k = \mathbf{H}_k \mathbf{W} \mathbf{H}_k^H + c \mathbf{I}_{N_f}$. If we define a matrix $\mathbf{V} = \begin{bmatrix} \mathbf{I}_{(N_f + L' - N_b)} & \mathbf{0}_{N_b \times (N_f + L' - N_b)}^H \end{bmatrix}^H$ of size $(N_f + L') \times (N_f + L' - N_b)$, then it is easy to verify that $\mathbf{W} = \mathbf{V} \mathbf{V}^H$. Hence, \mathbf{R}_k can be rewritten as:

$$\mathbf{R}_k = \bar{\mathbf{H}}_k \bar{\mathbf{H}}_k^H + c \mathbf{I}_{N_f} \quad (20)$$

where $\bar{\mathbf{H}}_k = \mathbf{H}_k \mathbf{V}$. $\bar{\mathbf{H}}_k$ and \mathbf{H}_{k+1} can be partitioned as, respectively:

$$\bar{\mathbf{H}}_k = \begin{bmatrix} \mathbf{B} & \mathbf{u}_k \\ \mathbf{0}^H & \alpha_k \end{bmatrix}, \quad \bar{\mathbf{H}}_{k+1} = \begin{bmatrix} \alpha_{k+1} & \mathbf{u}_{k+1}^H \\ \mathbf{0} & \mathbf{B} \end{bmatrix}, \quad (21)$$

where \mathbf{B} is an $(N_f - 1) \times (N_f - 1)$ sub-matrix, \mathbf{u}_k , \mathbf{u}_{k+1} and $\mathbf{0}$ are $(N_f - 1) \times 1$ column vectors and α_k , α_{k+1} are scalars. At time interval k , assume we have obtained \mathbf{R}_k^{-1} , which can be partitioned as:

$$\mathbf{R}_k^{-1} = \begin{bmatrix} \mathbf{Q} & \mathbf{v}_k \\ \mathbf{v}_k^H & \beta_k \end{bmatrix} \quad (22)$$

where \mathbf{Q} is an $(N_f - 1) \times (N_f - 1)$ sub-matrices, \mathbf{v}_k is $(N_f - 1) \times 1$ column vector and β_k is a scalar.

Then, the following holds:

$$\bar{\mathbf{Q}}^{-1} = \mathbf{Q} + \frac{(\mathbf{Q} \mathbf{u}_k) (\mathbf{Q} \mathbf{u}_k)^H}{b^{-1} - \mathbf{u}_k^H \mathbf{Q} \mathbf{u}_k} \quad (23)$$

where $\bar{\mathbf{Q}} \triangleq \mathbf{B} \mathbf{B}^H + c \mathbf{I}_{N_f - 1}$ and $b \triangleq 1 - |\alpha_k|^2 / (|\alpha_k|^2 + c)$. At time interval $k + 1$, the inverse matrix of \mathbf{R}_{k+1} can be calculated by:

$$\mathbf{R}_{k+1}^{-1} = \begin{bmatrix} \mu + \mu^2 \bar{\mathbf{u}}_{k+1}^H \bar{\Phi} \bar{\mathbf{u}}_{k+1} & -\mu \bar{\mathbf{u}}_{k+1}^H \bar{\Phi} \\ -\mu \bar{\Phi} \bar{\mathbf{u}}_{k+1} & \bar{\Phi} \end{bmatrix} \quad (24)$$

where $\mu \triangleq 1 / (|\alpha_{k+1}|^2 + \mathbf{u}_{k+1}^H \mathbf{u}_{k+1} + c)$, $\bar{\mathbf{u}}_{k+1} \triangleq \mathbf{B} \mathbf{u}_{k+1}$ and

$$\bar{\Phi} = \bar{\mathbf{Q}}^{-1} + \frac{(\bar{\mathbf{Q}}^{-1} \bar{\mathbf{u}}_{k+1}) (\bar{\mathbf{Q}}^{-1} \bar{\mathbf{u}}_{k+1})^H}{\mu^{-1} - \bar{\mathbf{u}}_{k+1}^H \bar{\mathbf{Q}}^{-1} \bar{\mathbf{u}}_{k+1}}. \quad (25)$$

The detailed derivation is shown in Appendix III. Similarly to the MMSE equalizer, the matrix inversion operation is only required once in the initial step, and the recursion will continue in the subsequent calculations.

The above procedure can be summarized as follows:

- 1) At time interval k , assume we have already obtained \mathbf{R}_k^{-1} as (22);
- 2) Calculate $\bar{\mathbf{Q}}^{-1}$ as (23);
- 3) At time interval $k + 1$, calculate Φ as (25) using $\bar{\mathbf{Q}}^{-1}$ obtained in Step 2);
- 4) Finally, substitute (25) into (24) to obtain \mathbf{R}_{k+1}^{-1} . Then, design \mathbf{f}_{k+1}^{MMSE} and \mathbf{b}_{k+1}^{MMSE} according to (19a) and (19b), respectively.

It is not hard to see that above derivations for the recursive MMSE/MMSE-DFE equalizers apply to any space-time coded cooperative network and any frequency-selective fading channel as long as the overall equivalent channel matrix is a convolutive matrix shown as in (7).

C. Some Remarks

In this part, we give some remarks of the above developed equalizers under multiple different CFOs.

Remark 1: Based on recently proposed design criterion [22], it is verified in [15] that without CFOs, both the MMSE equalizer and the MMSE-DFE equalizer can achieve asynchronous full cooperative diversity with elaborately designed coefficients $\bar{t}_l^{(r)}$, $l = 1, \dots, L$, in generating matrices $\bar{\mathbf{T}}^{(r)}$, $r = 1, \dots, R$. However, when considering CFOs, we obtain a time-varying MMSE/MMSE-DFE equalizer due to the time-varying equivalent channel. Generally it is difficult to do the performance analysis to the time-varying equalizers. In Section V, simulations show that the presence of CFOs will degrade the equalization performance compared with the case when the system is perfectly frequency-synchronized and the asynchronous full cooperative diversity may not be achieved when CFOs exist.

Remark 2: Denote two CFOs sets of R relay nodes as $\mathcal{F}_R = \{\Delta f_1, \Delta f_2, \dots, \Delta f_R\}$ and $\mathcal{F}'_R = \{\Delta f'_1, \Delta f'_2, \dots, \Delta f'_R\}$, respectively. Then, we can find that the performances of the MMSE/MMSE-DFE equalizer with these two CFOs sets are the same when the two CFOs sets only differ by a constant shift, i.e., $\mathcal{F}'_R = \mathcal{F}_R - \Delta f$, where Δf is an arbitrary constant. Since we always can multiply both sides of (7) with $\mathbf{E} = \text{diag}(e^{-j2\pi kT\Delta f} \mathbf{1}_{N_f \times 1})$ to cancel a common term $e^{j2\pi kT\Delta f}$ in \mathbf{P}_k before equalization, where $\mathbf{1}_{N_f \times 1}$ is an $N_f \times 1$ column vector with all one elements. In fact, we do not have to do this since we show in Appendix IV that, the MMSE/MMSE-DFE equalizers have the inherent property to cancel the common frequency offset Δf_{min} , where Δf_{min} is the minimum CFO in set \mathcal{F}_R . It means that, the absolute values of the CFOs are not important, only the *relative values* matter. For example, under the CFOs set $\mathcal{F}_3 = \{0.3, 0.32, 0.37\}$ the performance of the MMSE/MMSE-DFE

equalizer is the same as the one with CFOs set $\mathcal{F}'_3 = \{0, 0.02, 0.07\}$.

Remark 3: The simulation results in Section V also show that generally there is no deterministic relationship between the value of CFOs and the equalization performance. In some cases, smaller CFOs may result in better performance while in other cases it turns out just the opposite. An interesting phenomenon is that in some cases, more relay nodes may have worse performance when multiple CFOs are present.

IV. COMPLEXITY ANALYSIS

In Section III, we derived FIR linear MMSE equalizer and MMSE-DFE equalizer for the asynchronous cooperative system with multiple CFOs. They are both *serial* equalizers, which means that at each time interval, the equalizer only estimate one symbol. However, recall that the symbols are transmitted from relay nodes block by block. Moreover, guard intervals are inserted to overcome the inter-block interference. Therefore, a *block* equalizer [25], [26] can be used to detect symbols as well, i.e., the destination node receives the whole block and then estimate all symbols in the block simultaneously and the matrix inverse is done once for a block. In this section, we first recall the block MMSE and MMSE-DFE equalizers, and then compare their computational complexities with our proposed recursive algorithms.

Considering all received symbols in one block, we get the following data model:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$$

where $\mathbf{y} = [y_1, y_2, \dots, y_{N+L'-1}]^T$, \mathbf{H} is the $(N + L' - 1) \times N$ equivalent channel matrix, which also contains the CFO matrix, channel coefficients and generating matrix, $\mathbf{s} = [x_1, x_2, \dots, x_N]^T$ and $\mathbf{n} = [n_1, n_2, \dots, n_{N+L'-1}]^T$.

A. Block MMSE and MMSE-DFE Equalizers

The block MMSE equalizer is to find equalization matrix \mathbf{F} to satisfy the MMSE criteria, i.e.,

$$\mathbf{F}^{MMSE} = \arg \min_{\mathbf{F} \in \mathbb{C}^{N \times (N+L'-1)}} E \|\mathbf{s} - \mathbf{F}\mathbf{y}\|^2$$

and the solution is [25]:

$$\mathbf{F}^{MMSE} = (\mathbf{H}^H \mathbf{H} + c\mathbf{I}_N)^{-1} \mathbf{H}^H$$

where $c = \sigma_n^2/\sigma_s^2$. After equalization, we get $\hat{\mathbf{s}} = Q(\mathbf{F}^{MMSE}\mathbf{y})$. Similarly, the block MMSE-DFE equalizer is to design feedforward filter-matrix \mathbf{F} and feedback filter-matrix \mathbf{B} to satisfy:

$$\{\mathbf{F}^{MMSE-DFE}, \mathbf{B}^{MMSE-DFE}\} = \arg \min_{\mathbf{F} \in \mathbb{C}^{N \times (N+L'-1)}, \mathbf{B} \in \mathbb{C}^{N \times N}} \|\mathbf{s} - (\mathbf{F}\mathbf{y} - \mathbf{B}\hat{\mathbf{s}})\|^2.$$

In order to feedback decisions in a causal way, we require \mathbf{B} to be a zero diagonal upper triangular matrix. We also assume past decisions are correct, i.e., $\hat{\mathbf{s}} = \mathbf{s}$. Let $\bar{\mathbf{R}} = \mathbf{H}^H\mathbf{H} + c\mathbf{I}_N$, and the ‘‘Cholesky’’ factorization (lower-diagonal-upper) of $\bar{\mathbf{R}}$ be $\bar{\mathbf{R}} = \mathbf{L}\mathbf{D}\mathbf{L}^H$, where \mathbf{L} is a monic lower triangular matrix and \mathbf{D} is a diagonal matrix. Then, the block MMSE-DFE can be given by [25]:

$$\begin{aligned} \mathbf{B}^{MMSE-DFE} &= \mathbf{L}^H - \mathbf{I}_N \\ \mathbf{F}^{MMSE-DFE} &= \mathbf{L}^H (\mathbf{H}^H\mathbf{H} + c\mathbf{I}_N)^{-1} \mathbf{H}^H = \mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{H}^H. \end{aligned}$$

B. Complexity Analysis

We now calculate the numbers of complex operations (we consider complex multiplication (CM) and complex division (CD) here) for the recursive algorithms and compare them with that of the block ones.

As to the recursive MMSE equalizer, assuming we have obtained \mathbf{R}_k^{-1} , the computational load to calculate \mathbf{R}_{k+1}^{-1} and to get the soft estimation $\tilde{\mathbf{s}}_{k+1}$ is listed in Table I. Therefore, to detect the whole block \mathbf{s} with N symbols, total computational complexity is in the order of $\mathcal{O}(N_f^2N)$. Similarly, for the recursive MMSE-DFE equalizer, the computational complexity to obtain the feedforward filter is almost the same with that of the recursive MMSE equalizer, To obtain the feedback filter, another N_bN_f CM are needed. Hence, a total of $4(N_f - 1)^2 + 2N_f^2 + N_bN_f + 4N_f + N_b - 3$ CM and $(N_f - 1)^2$ CD are required to get $\tilde{\mathbf{s}}_{k+1}$. To detect the whole block \mathbf{s} , the complexity is about $\mathcal{O}(N_f^2N)$, which is in the same order as that of the recursive MMSE receiver.

For the block MMSE equalizer, the calculation of $\bar{\mathbf{R}} = \mathbf{H}^H\mathbf{H} + c\mathbf{I}_N$ and the matrix inverse require $(N + L' - 1)^2N$ CM and N^3 CM, respectively. To get \mathbf{F}^{MMSE} and then equalize the block, another $(N + L' - 1)N + N^2$ CM are required. Hence the total computational load to detect the block is $(N + L' - 1)^2N + N^3 + (N + L' - 1)N + N^2$ CM which is in the order of $\mathcal{O}(N^3)$. Similarly, it is not hard to obtain the total complexity of block MMSE-DFE to be in the order of $\mathcal{O}(N^3)$ as well. The complexities are also summarized in Table I. From the comparison, one can see that when $N \gg N_f$, the computational load of recursive equalizers are much lower compared with the block ones.

V. SIMULATION RESULTS

In this section, we present some simulation results to illustrate the equalization performance of our proposed linear MMSE and MMSE-DFE equalizers. Assume the carrier frequency is 2.5GHz, the symbol duration is $T = 20\mu s$, and the oscillator's stability is 10ppm. Hence the CFO can be as large as $2.5\text{GHz} \times 10\text{ppm} = 25\text{kHz}$ and normalized CFO $\Delta\phi \triangleq \Delta f T \in [-0.5, 0.5]$. The constellation we used is QPSK and each block contains 80 symbols. The coefficients in generating matrix (1) are designed according to [15]. For $R = 2$, the coefficient vector are $\mathbf{t}^{(1)} = [1/\sqrt{2}, 1/\sqrt{2}]$ and $\mathbf{t}^{(2)} = [1/\sqrt{2}, -1/\sqrt{2}]$. For $R = 3$, they are $\mathbf{t}^{(1)} = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]$, $\mathbf{t}^{(2)} = [1/\sqrt{6}, -2/\sqrt{6}, 1/\sqrt{6}]$ and $\mathbf{t}^{(3)} = [1/\sqrt{2}, 0, 1/\sqrt{2}]$. Assume that the channel between any relay node and the destination node is quasi-static Rayleigh flat fading (channel coefficients are complex Gaussian random variables with zero mean and normalized variance $1/R$), the delays from the relays are uniformly distributed in $[0, \tau_{max}]$ and in simulations we set $\tau_{max} = 3$. The lengths of MMSE equalizer and the feedforward filter of MMSE-DFE equalizer are fixed to 21 and 10, respectively. The bit error rate (BER) vs. bit signal-noise-ratio (E_b/N_0) curves are plotted to demonstrate the performance.

Fig.2 and Fig.3 illustrate the performances of the recursive MMSE/MMSE-DFE equalizers with different sets of CFOs and different numbers of relay nodes, $R = 2$ and $R = 3$, respectively. We can see that without CFOs, both these two cases can achieve full diversity order even with the timing error. Although both equalization performances will degrade in the presence of CFOs, the MMSE-DFE equalizer outperforms the linear MMSE equalizer significantly. It is also shown that the MMSE equalizer will encounter error floor at moderate SNR (about 20dB) while the MMSE-DFE equalizer will not. So MMSE-DFE equalizer is more robust when CFOs are present.

Fig.4 shows the property discussed in remark 2 of Section III.C. The performances with CFOs sets $\mathcal{F}_3^1 = \{0.2, 0.27, 0.31\}$, $\mathcal{F}_3^2 = \{-0.2, -0.13, -0.09\}$ and $\mathcal{F}_3^3 = \{0, 0.07, 0.11\}$ are the same for MMSE or MMSE-DFE equalizers.

Fig.5 compares the performance of our proposed serial MMSE-DFE equalizer (SDFE) and the block MMSE-DFE equalizer (BDFE) in Section IV. Because the length of feedforward FIR filter of our serial MMSE-DFE is $N_f = 10$ while that of block equalizer is the block length $N = 80$, the performance of our scheme is a bit inferior to the block equalizer (less than 0.3 dB). However, considering the lower

computational complexity, our recursive serial MMSE-DFE equalizer is a better tradeoff.

Fig.6 gives an example to show that there is no particular relationship between the value of CFOs and the equalization performance. The MMSE-DFE equalization is considered. The performance with CFO set $\{-0.5, 0.37, 0.4\}$ is worse than that with $\{0.34, 0.37, 0.4\}$ while better than the case with $\{-0.35, 0.37, 0.4\}$. Another observation is that more relay nodes will not necessarily improve the performance in the presence of CFOs. Assuming at some time there are two relay nodes with normalized CFOs $\{0.37, 0.4\}$ and then a third relay node is available. If the normalized CFO of this relay is -0.5 or 0.34 , then it will improve the performance. However, if the CFO is -0.35 , the performance is worse than the two relay case.

VI. CONCLUSION

In this paper, we have investigated the equalization issue for linear convolutively coded cooperative network with multiple different carrier frequency offsets and possibly different time delays. We have proposed recursive algorithms for MMSE and MMSE-DFE equalizers where equalization filter matrix inversions are not needed for every symbol in a channel coherent time duration.

APPENDIX I

DERIVATION OF RECURSIVE ALGORITHM FOR MMSE EQUALIZER

We firstly recall two important formulas, which will be used in our following derivations. The first one is the inversion formula [27] for a partitioned matrix, which can be expressed as:

$$\begin{aligned} & \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{-1} \\ = & \begin{bmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1} \mathbf{A}_{12} (\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} \mathbf{A}_{21} \mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1} \mathbf{A}_{12} (\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} \\ -(\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} \mathbf{A}_{21} \mathbf{A}_{11}^{-1} & (\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} \end{bmatrix} \end{aligned} \quad (26)$$

$$= \begin{bmatrix} (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21})^{-1} & -\mathbf{A}_{11}^{-1} \mathbf{A}_{12} (\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} \\ -(\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} \mathbf{A}_{21} \mathbf{A}_{11}^{-1} & (\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} \end{bmatrix} \quad (27)$$

The second one is the Woodbury fomula, which is given by:

$$(\mathbf{A} + \gamma \mathbf{u} \mathbf{v}^{\mathcal{H}})^{-1} = \mathbf{A}^{-1} - \frac{\gamma}{1 + \gamma \mathbf{v}^{\mathcal{H}} \mathbf{A}^{-1} \mathbf{u}} \mathbf{A}^{-1} \mathbf{u} \mathbf{v}^{\mathcal{H}} \mathbf{A}^{-1} \quad (28)$$

where \mathbf{u} and \mathbf{v} are column vectors with proper dimensions.

It is easy to see \mathbf{P}_k in (9) can be partitioned as:

$$\mathbf{P}_k = \begin{bmatrix} \bar{\mathbf{P}}_k & \mathbf{0}_{(N_f-1) \times R} \\ \mathbf{0}_{1 \times (N_f-1)R} & \mathbf{e}_{k-N_f+1} \end{bmatrix}$$

where $\bar{\mathbf{P}}_k = \mathbf{P}_k(1 : N_f - 1, 1 : (N_f - 1)R)$. Correspondingly, $\bar{\mathbf{H}}$ in (10) can be partitioned as:

$$\bar{\mathbf{H}} = \begin{bmatrix} \bar{\mathbf{H}}_{11} & \mathbf{0}_{(N_f-1)R \times 1} \\ \bar{\mathbf{H}}_{21} & \bar{\mathbf{H}}_{22} \end{bmatrix}$$

where $\bar{\mathbf{H}}_{11} = \bar{\mathbf{H}}(1 : (N_f - 1)R, 1 : N_f + L' - 2)$, $\bar{\mathbf{H}}_{21} = \bar{\mathbf{H}}((N_f - 1)R + 1 : N_f R, 1 : N_f + L' - 2)$, and $\bar{\mathbf{H}}_{22}$ is a $R \times 1$ vector. Hence, \mathbf{H}_k in (8) becomes:

$$\mathbf{H}_k = \begin{bmatrix} \bar{\mathbf{P}}_k \bar{\mathbf{H}}_{11} & \mathbf{0}_{(N_f-1) \times 1} \\ \mathbf{h}_k^{\mathcal{H}} & e_k \end{bmatrix}$$

where $\mathbf{h}_k^{\mathcal{H}} = \mathbf{e}_{k-N_f+1} \bar{\mathbf{H}}_{21}$ is a $1 \times (N_f + L' - 2)$ row vector and $e_k = \mathbf{e}_{k-N_f+1} \bar{\mathbf{H}}_{22}$ is a scalar. Now we have

$$\begin{aligned} \mathbf{R}_k &= \mathbf{H}_k \mathbf{H}_k^{\mathcal{H}} + c \mathbf{I}_{N_f} = \begin{bmatrix} \bar{\mathbf{P}}_k \bar{\mathbf{H}}_{11} & \mathbf{0}_{(N_f-1) \times 1} \\ \mathbf{h}_k^{\mathcal{H}} & e_k \end{bmatrix} \begin{bmatrix} \bar{\mathbf{H}}_{11}^{\mathcal{H}} \bar{\mathbf{P}}_k^{\mathcal{H}} & \mathbf{h}_k \\ \mathbf{0}_{1 \times (N_f-1)} & e_k^* \end{bmatrix} + c \mathbf{I}_{N_f} \\ &= \begin{bmatrix} \Xi & \mathbf{r}_k \\ \mathbf{r}_k^{\mathcal{H}} & r_k \end{bmatrix} \end{aligned} \quad (29)$$

where $\Xi = \bar{\mathbf{P}}_k \bar{\mathbf{H}}_{11} \bar{\mathbf{H}}_{11}^{\mathcal{H}} \bar{\mathbf{P}}_k^{\mathcal{H}} + c \mathbf{I}_{N_f-1}$, $\mathbf{r}_k = \bar{\mathbf{P}}_k \bar{\mathbf{H}}_{11} \mathbf{h}_k$ and $r_k = \mathbf{h}_k^{\mathcal{H}} \mathbf{h}_k + |e_k|^2 + c$.

If we partition \mathbf{R}_k^{-1} as (14), then applying the inversion formula (27) to \mathbf{R}_k in (29), we obtain:

$$\Theta = \left(\Xi - \frac{\mathbf{r}_k \mathbf{r}_k^{\mathcal{H}}}{r_k} \right)^{-1}$$

or equivalently

$$\Xi^{-1} = \left(\Theta^{-1} + \frac{\mathbf{r}_k \mathbf{r}_k^{\mathcal{H}}}{r_k} \right)^{-1} = \Theta - \frac{(\Theta \mathbf{r}_k) (\Theta \mathbf{r}_k)^{\mathcal{H}}}{\nu_k^{-1} + \mathbf{r}_k^{\mathcal{H}} \Theta \mathbf{r}_k} \quad (30)$$

where $\nu_k = 1/r_k$. (30) is obtained by using Woodbury formula (28).

At symbol interval $k + 1$, substituting k with $k + 1$ in (9), \mathbf{P}_{k+1} can be partitioned as:

$$\mathbf{P}_{k+1} = \begin{bmatrix} \mathbf{e}_{k+1} & \mathbf{0}_{1 \times (N_f-1)R} \\ \mathbf{0}_{(N_f-1) \times R} & \bar{\mathbf{P}}_k \end{bmatrix}$$

where $\bar{\mathbf{P}}_k = \mathbf{P}_{k+1}(2 : N_f, R + 1 : N_f R)$. Note that the left-upper sub-matrix of \mathbf{P}_k is the same as the right-lower sub-matrix of \mathbf{P}_{k+1} . Correspondingly, $\bar{\mathbf{H}}$ in (10) can be re-partitioned as:

$$\bar{\mathbf{H}} = \begin{bmatrix} \bar{\mathbf{H}}'_{11} & \bar{\mathbf{H}}'_{12} \\ \mathbf{0}_{(N_f-1)R \times 1} & \bar{\mathbf{H}}'_{22} \end{bmatrix}$$

where $\bar{\mathbf{H}}'_{11} = \bar{\mathbf{H}}(1 : R, 1)$, $\bar{\mathbf{H}}'_{22} = \bar{\mathbf{H}}(R + 1 : N_f R, 2 : N_f + L' - 1)$. Note due to the special shift structure of $\bar{\mathbf{H}}$, we have $\bar{\mathbf{H}}_{11} = \bar{\mathbf{H}}'_{22}$. Now \mathbf{H}_{k+1} in (8) becomes:

$$\mathbf{H}_{k+1} = \begin{bmatrix} e_{k+1} & \mathbf{h}_{k+1}^{\mathcal{H}} \\ \mathbf{0}_{(N_f-1) \times 1} & \bar{\mathbf{P}}_k \bar{\mathbf{H}}'_{22} \end{bmatrix}$$

where $e_{k+1} = \mathbf{e}_{k+1} \bar{\mathbf{H}}'_{11}$ is a scalar and $\mathbf{h}_{k+1}^{\mathcal{H}} = \mathbf{e}_{k+1} \bar{\mathbf{H}}'_{12}$ is a $1 \times (N_f + L' - 2)$ row vector.

Similar to (29), we have:

$$\mathbf{R}_{k+1} = \begin{bmatrix} e_{k+1} & \mathbf{h}_{k+1}^{\mathcal{H}} \\ \mathbf{0}_{(N_f-1) \times 1} & \bar{\mathbf{P}}_k \bar{\mathbf{H}}'_{22} \end{bmatrix} \begin{bmatrix} e_{k+1}^* & \mathbf{0}_{1 \times (N_f-1)} \\ \mathbf{h}_{k+1} & (\bar{\mathbf{H}}'_{22})^{\mathcal{H}} \bar{\mathbf{P}}_k^{\mathcal{H}} \end{bmatrix} + c \mathbf{I}_{N_f} = \begin{bmatrix} r_{k+1} & \mathbf{r}_{k+1}^{\mathcal{H}} \\ \mathbf{r}_{k+1} & \Xi \end{bmatrix} \quad (31)$$

where $r_{k+1} = \mathbf{h}_{k+1}^{\mathcal{H}} \mathbf{h}_{k+1} + |e_{k+1}|^2 + c$, $\mathbf{r}_{k+1} = \bar{\mathbf{P}}_k \bar{\mathbf{H}}'_{22} \mathbf{h}_{k+1}$ and $\Xi = \bar{\mathbf{P}}_k \bar{\mathbf{H}}'_{11} \bar{\mathbf{H}}_{11}^{\mathcal{H}} \bar{\mathbf{P}}_k^{\mathcal{H}} + c \mathbf{I}_{N_f-1} = \bar{\mathbf{P}}_k \bar{\mathbf{H}}'_{22} (\bar{\mathbf{H}}'_{22})^{\mathcal{H}} \bar{\mathbf{P}}_k^{\mathcal{H}} + c \mathbf{I}_{N_f-1}$ due to $\bar{\mathbf{H}}_{11} = \bar{\mathbf{H}}'_{22}$.

Applying the inversion formula (26) to \mathbf{R}_{k+1} in (31), \mathbf{R}_{k+1}^{-1} can be written as:

$$\mathbf{R}_{k+1}^{-1} = \begin{bmatrix} \nu_{k+1} + \nu_{k+1}^2 \mathbf{r}_{k+1}^{\mathcal{H}} \Psi \mathbf{r}_{k+1} & -\nu_{k+1} \mathbf{r}_{k+1}^{\mathcal{H}} \Psi \\ -\nu_{k+1} \Psi \mathbf{r}_{k+1} & \Psi \end{bmatrix}$$

where $\nu_{k+1} = 1/r_{k+1}$ and

$$\Psi = (\Xi - \nu_{k+1} \mathbf{r}_{k+1} \mathbf{r}_{k+1}^{\mathcal{H}})^{-1} = \Xi^{-1} + \frac{(\Xi^{-1} \mathbf{r}_{k+1}) (\Xi^{-1} \mathbf{r}_{k+1})^{\mathcal{H}}}{\nu_{k+1}^{-1} - \mathbf{r}_{k+1}^{\mathcal{H}} \Xi^{-1} \mathbf{r}_{k+1}}$$

where the second equation is obtained by using Woodbury formula (28).

APPENDIX II DERIVATION OF MMSE-DFE EQUALIZER

In this section, we derive the MMSE-DFE equalizer in (19a) and (19b). Our derivation is slightly different from that in [23], [24] and [25] because it is more convenient to derive the recursive MMSE-DFE algorithm based on our derivation here. From (18), define an augmented filter coefficients vector as $\mathbf{w}_k \triangleq [\mathbf{f}_k^{\mathcal{H}} \quad \mathbf{b}_k^{\mathcal{H}}]^{\mathcal{H}}$, and a corresponding augmented input vector as $\bar{\mathbf{y}}_k \triangleq [\mathbf{y}_k^{\mathcal{H}} \quad \hat{\mathbf{s}}_{k-D-1}^{\mathcal{H}}]^{\mathcal{H}}$. Then, (18) becomes:

$$\mathbf{w}_k^{MMSE-DFE} = \arg \min_{\mathbf{w}_k \in \mathbb{C}^{(N_f+N_b) \times 1}} E |s_{k-D} - \mathbf{w}_k^{\mathcal{H}} \bar{\mathbf{y}}_k|^2.$$

This is a standard linear optimum filtering problem [28]. By the orthogonality principle, we have:

$$(\mathbf{w}_k^{MMSE-DFE})^{\mathcal{H}} E [\bar{\mathbf{y}}_k \bar{\mathbf{y}}_k^{\mathcal{H}}] = E [s_{k-D} \bar{\mathbf{y}}_k^{\mathcal{H}}]. \quad (32)$$

We assume the symbol sequence and the additive noise are wide-sense stationary, mutually uncorrelated and white with variance σ_s^2 and σ_n^2 . Hence $E[s_i s_i] = \sigma_s^2$ and $E[s_i s_j] = 0$ for any $i \neq j$. Under the correct past decision assumption and $D = N_f - 1$, substituting (7) into (32), we have:

$$E [s_{k-D} \bar{\mathbf{y}}_k^{\mathcal{H}}] = E \left[\begin{array}{cc} s_{k-D} \mathbf{y}_k^{\mathcal{H}} & s_{k-D} \mathbf{s}_{k-D-1}^{\mathcal{H}} \end{array} \right] = \sigma_s^2 \left[(\mathbf{H}_k \mathbf{i}_D)^{\mathcal{H}} \quad \mathbf{0}_{1 \times N_b} \right] \quad (33)$$

and

$$E [\bar{\mathbf{y}}_k \bar{\mathbf{y}}_k^{\mathcal{H}}] = E \left(\begin{bmatrix} \mathbf{y}_k \\ \mathbf{s}_{k-D-1} \end{bmatrix} \begin{bmatrix} \mathbf{y}_k^{\mathcal{H}} & \mathbf{s}_{k-D-1}^{\mathcal{H}} \end{bmatrix} \right) = \begin{bmatrix} \sigma_s^2 \mathbf{H}_k \mathbf{H}_k^{\mathcal{H}} + \sigma_n^2 \mathbf{I}_{N_f} & \sigma_s^2 \mathbf{H}_k \mathbf{U}^{\mathcal{H}} \\ \sigma_s^2 \mathbf{U} \mathbf{H}_k^{\mathcal{H}} & \sigma_s^2 \mathbf{I}_{N_b} \end{bmatrix} \quad (34)$$

where $\mathbf{U} = \begin{bmatrix} \mathbf{0}_{N_b \times (N_f + L' - 1 - N_b)} & \mathbf{I}_{N_b} \end{bmatrix}$. Then, substituting (33) and (34) into (32), we get:

$$\mathbf{f}_k^{\mathcal{H}} (\mathbf{H}_k \mathbf{H}_k^{\mathcal{H}} + \sigma_n^2 / \sigma_s^2 \mathbf{I}_{N_f}) + \mathbf{b}_k^{\mathcal{H}} \mathbf{U} \mathbf{H}_k^{\mathcal{H}} = (\mathbf{H}_k \mathbf{i}_D)^{\mathcal{H}} \quad (35)$$

$$\mathbf{f}_k^{\mathcal{H}} \mathbf{H}_k \mathbf{U}^{\mathcal{H}} + \mathbf{b}_k^{\mathcal{H}} = \mathbf{0}_{1 \times N_b}. \quad (36)$$

From (35) and (36), It is easy to obtain $\mathbf{f}_k^{\text{MMSE-DFE}}$ and $\mathbf{b}_k^{\text{MMSE-DFE}}$ as (19a) and (19b).

APPENDIX III DERIVATION OF RECURSIVE ALGORITHM FOR MMSE-DFE EQUALIZER

Rewrite \mathbf{R}_k in (20) here:

$$\mathbf{R}_k = \bar{\mathbf{H}}_k \bar{\mathbf{H}}_k^{\mathcal{H}} + c \mathbf{I}_{N_f} \quad (37)$$

Note that we have $N_b = L' - 1$, $\bar{\mathbf{H}}_k = \mathbf{H}_k \mathbf{V}$, and $\mathbf{V} = \begin{bmatrix} \mathbf{I}_{(N_f + L' - N_b)} & \mathbf{0}_{N_b \times (N_f + L' - N_b)}^{\mathcal{H}} \end{bmatrix}^{\mathcal{H}}$. It is easy to find that $\bar{\mathbf{H}}_k$ can be partitioned as:

$$\bar{\mathbf{H}}_k = \begin{bmatrix} \mathbf{B} & \mathbf{u}_k \\ \mathbf{0}^H & \alpha_k \end{bmatrix} \quad (38)$$

where \mathbf{B} is a $(N_f - 1) \times (N_f - 1)$ sub-matrix and α_k is a scalar. Then, substitute (38) into (37), we have:

$$\mathbf{R}_k^{-1} = \begin{bmatrix} \mathbf{B} \mathbf{B}^H + \mathbf{u}_k \mathbf{u}_k^H + c \mathbf{I}_{N_f - 1} & \alpha_k^* \mathbf{u}_k \\ \alpha_k \mathbf{u}_k^H & |\alpha_k|^2 + c \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{Q} & \mathbf{v}_k \\ \mathbf{v}_k^H & \beta_k \end{bmatrix}.$$

Using inversion fomula (27), we have

$$\mathbf{Q} = \left(\mathbf{B} \mathbf{B}^H + c \mathbf{I}_{N_f - 1} + \left(1 - \frac{|\alpha_k|^2}{|\alpha_k|^2 + c} \right) \mathbf{u}_k \mathbf{u}_k^H \right)^{-1}.$$

For convenience, let $\bar{\mathbf{Q}} \triangleq \mathbf{B} \mathbf{B}^H + c \mathbf{I}_{N_f - 1}$ and $b \triangleq 1 - |\alpha_k|^2 / (|\alpha_k|^2 + c)$. Using Woodbury formula (28) again, we obtain:

$$\bar{\mathbf{Q}}^{-1} = (\mathbf{Q}^{-1} - b \mathbf{u}_k \mathbf{u}_k^H)^{-1} = \mathbf{Q} + \frac{(\mathbf{Q} \mathbf{u}_k) (\mathbf{Q} \mathbf{u}_k)^H}{b^{-1} - \mathbf{u}_k^H \mathbf{Q} \mathbf{u}_k}$$

At time interval $k + 1$, $\bar{\mathbf{H}}_{k+1}$ can be partitioned as:

$$\bar{\mathbf{H}}_{k+1} = \begin{bmatrix} \alpha_{k+1} & \mathbf{u}_{k+1}^H \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \quad (39)$$

Then, substituting (39) into (37), and using inversion formula (26), we have:

$$\begin{aligned} \mathbf{R}_{k+1}^{-1} &= \begin{bmatrix} |\alpha_{k+1}|^2 + \mathbf{u}_{k+1}^H \mathbf{u}_{k+1} + c & \mathbf{u}_{k+1}^H \mathbf{B}^H \\ \mathbf{B} \mathbf{u}_{k+1} & \mathbf{B} \mathbf{B}^H + c \mathbf{I}_{N_f-1} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mu + \mu^2 \bar{\mathbf{u}}_{k+1}^H \Phi \bar{\mathbf{u}}_{k+1} & -\mu \bar{\mathbf{u}}_{k+1}^H \Phi \\ -\mu \Phi \bar{\mathbf{u}}_{k+1} & \Phi \end{bmatrix} \end{aligned}$$

where $\mu = 1/(|\alpha_{k+1}|^2 + \mathbf{u}_{k+1}^H \mathbf{u}_{k+1} + c)$, $\bar{\mathbf{u}}_{k+1} = \mathbf{B} \mathbf{u}_{k+1}$ and

$$\Phi = (\bar{\mathbf{Q}} - \mu \bar{\mathbf{u}}_{k+1} \bar{\mathbf{u}}_{k+1}^H)^{-1} = \bar{\mathbf{Q}}^{-1} + \frac{(\bar{\mathbf{Q}}^{-1} \bar{\mathbf{u}}_{k+1}) (\bar{\mathbf{Q}}^{-1} \bar{\mathbf{u}}_{k+1})^H}{\mu^{-1} - \bar{\mathbf{u}}_{k+1}^H \bar{\mathbf{Q}}^{-1} \bar{\mathbf{u}}_{k+1}}$$

where the second equation is again obtained by using Woodbury formula (28).

APPENDIX IV

When CFOs set is \mathcal{F} , without loss of generality, we assume $\Delta f_1 \leq \Delta f_2 \leq \dots \leq \Delta f_R$, so $\Delta f_{min} = \Delta f_1$. From (9), the CFOs matrix \mathbf{P}_k can be rewritten as:

$$\mathbf{P}_k = \mathbf{E} \bar{\mathbf{P}}_k$$

where $\mathbf{E} = \text{diag}(e^{j2\pi kT \Delta f_1} \mathbf{1}_{N_f \times 1})$, $\mathbf{1}_{N_f \times 1}$ is a $N_f \times 1$ column vector with all elements are 1's,

$$\bar{\mathbf{P}}_k = \begin{bmatrix} \bar{\mathbf{e}}_k & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} \\ \mathbf{0}_{1 \times R} & \bar{\mathbf{e}}_{k-1} & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} & \mathbf{0}_{1 \times R} & \bar{\mathbf{e}}_{k-N_f+1} \end{bmatrix}$$

and $\bar{\mathbf{e}}_k = [1, e^{j2\pi kT(\Delta f_2 - \Delta f_1)}, e^{j2\pi kT(\Delta f_3 - \Delta f_1)}, \dots, e^{j2\pi kT(\Delta f_R - \Delta f_1)}]$. Note that $\bar{\mathbf{P}}_k$ is exactly the CFOs matrix with the CFOs set $\mathcal{F}' = \{0, \Delta f_2 - \Delta f_1, \Delta f_3 - \Delta f_1, \dots, \Delta f_R - \Delta f_1\}$.

It is easy to see that $\mathbf{E} \mathbf{E}^H = \mathbf{E}^H \mathbf{E} = \mathbf{I}_{N_f}$. Substituting \mathbf{P}_k into (12), we obtain:

$$\mathbf{f}_k^{MMSE} = \mathbf{E} (\bar{\mathbf{P}}_k \bar{\mathbf{H}} \bar{\mathbf{H}}^H \bar{\mathbf{P}}_k^H + c \mathbf{I}_{N_f})^{-1} \bar{\mathbf{P}}_k \bar{\mathbf{H}} \mathbf{i}_D$$

for the MMSE equalizer. Hence, the soft estimation of the MMSE equalizer with CFOs set \mathcal{F} is

$$\tilde{s}_k = (\mathbf{f}_k^{MMSE})^H \mathbf{y}_k = (\bar{\mathbf{P}}_k \bar{\mathbf{H}} \mathbf{i}_D)^H (\bar{\mathbf{P}}_k \bar{\mathbf{H}} \bar{\mathbf{H}}^H \bar{\mathbf{P}}_k^H + c \mathbf{I}_{N_f})^{-1} (\bar{\mathbf{P}}_k \bar{\mathbf{H}} + \mathbf{n}_k).$$

It is easy to find that $\bar{\mathbf{P}}_k \bar{\mathbf{H}} + \mathbf{n}_k$ is the received vector and $(\bar{\mathbf{P}}_k \bar{\mathbf{H}} \bar{\mathbf{H}}^H \bar{\mathbf{P}}_k^H + c \mathbf{I}_{N_f})^{-1} \bar{\mathbf{P}}_k \bar{\mathbf{H}} \mathbf{i}_D$ is the MMSE equalizer with CFO sets \mathcal{F}' , respectively. Therefore, the soft estimation of s_k is exactly the same for both \mathcal{F} and \mathcal{F}' .

For the MMSE-DFE equalizer, the soft estimation is $\tilde{s}_{k-D} = \mathbf{f}_k^H \mathbf{y}_k + \mathbf{b}_k^H \hat{\mathbf{s}}_{k-D-1}$, the proof of first part $\mathbf{f}_k^H \mathbf{y}_k$ is almost the same with the MMSE case. For the second part, it is easy to see that $\mathbf{b}_k^{MMSE-DFE} =$

$-\mathbf{U}\bar{\mathbf{H}}^H\bar{\mathbf{P}}_k^H(\bar{\mathbf{P}}_k\bar{\mathbf{H}}\mathbf{V}\mathbf{V}^H\bar{\mathbf{H}}^H\bar{\mathbf{P}}_k^H + c\mathbf{I}_{N_f})^{-1}\bar{\mathbf{P}}_k\bar{\mathbf{H}}\mathbf{i}_D$ for both CFOs sets \mathcal{F} and \mathcal{F}' . Besides, $\hat{\mathbf{s}}_{k-D-1}$ is the same under the assumption of correct past detection. Hence, the soft estimation of s_{k-D} is the same as well.

Therefore, for both equalizers, the soft estimation is the same. Consequently, after hard decision $\hat{\mathbf{s}} = Q(\tilde{\mathbf{s}})$, the symbol error rate (bit error rate) performances are the same.

REFERENCES

- [1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperative diversity - Part I: System description, Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, pp. 1927 – 1948, Nov. 2003.
- [2] J. N. Laneman and W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2415 – 2425, Oct. 2003.
- [3] M. Janani, A. Hedayat, T. E. Hunter, and A. Nosratinia, "Coded cooperation in wireless communications: Space-time transmission and iterative decoding," *IEEE Trans. Signal Processing*, vol. 52, pp. 362 – 371, Feb. 2004.
- [4] Y. Jing and B. Hassibi, "Distributed space-time coding in wireless relay networks," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 3524–3536, Dec. 2006.
- [5] X. Li, "Space-time coded multi-transmission among distributed transmitters without perfect synchronization," *IEEE Signal Processing Lett.*, vol. 11, pp. 948–951, Dec. 2004.
- [6] S. Wei, D. Goeckel, and M. Valenti, "Asynchronous cooperative diversity," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 1547–1557, June 2006.
- [7] Y. Li and X.-G. Xia, "A family of distributed space-time trellis codes with asynchronous cooperative diversity," *IEEE Trans. Commun.*, vol. 55, pp. 790–800, Apr. 2007.
- [8] Y. Shang and X.-G. Xia, "Shift full rank matrices and applications in space-time trellis codes for relay networks with asynchronous cooperative diversity," *IEEE Trans. Inform. Theory*, vol. 52, pp. 3153–3167, July 2006.
- [9] A. R. Hammons, "Algebraic space-time codes for quasi-synchronous cooperative diversity," in *Proc. IEEE Conf. Wireless Networks, Commun. Mobile Comput.*, Maui, HI., pp. 11-15, June 2005.
- [10] M. O. Damen and A. R. Hammons, "Delay-tolerant distributed TAST codes for cooperative diversity," *IEEE Trans. Inform. Theory*, vol. 53, pp.3755-3773, Oct. 2007.
- [11] Y. Shang and X.-G. Xia, "Limited-shift-full-rank matrices with applications in asynchronous cooperative communications," *IEEE Trans. Inform. Theory*, vol. 53, pp.4199-4126, Nov. 2007.
- [12] Y. Mei, Y. Hua, A. Swami, and B. Daneshrad, "Combating synchronization errors in cooperative relays," in *Proc. IEEE ICASSP 2005*, vol. 3, pp. 369-372, Mar. 2005.
- [13] Y. Li, W. Zhang, and X.-G. Xia, "Distributive high-rate full-diversity space-frequency codes for asynchronous cooperative communications," in *Proc. IEEE ISIT'06*, Seattle, USA, July 2006, pp. 2612–2616.
- [14] Z. Li and X.-G. Xia, "A simple Alamouti space-time transmission scheme for asynchronous cooperative systems," *IEEE Signal Process. Letts.*, Nov. 2007.
- [15] X. Guo and X.-G. Xia, "Distributed linear convolutive space-time codes for asynchronous cooperative communication networks," preprint, 2007.
- [16] Z. Li, D. Qu and G. Zhu, "An equalization technique for distributed STBC-OFDM system with multiple carrier frequency offsets," in *Proc. IEEE WCNC 2006*, vol. 2, pp. 839-843, April 2006.
- [17] D. Veronesi and D. L. Goeckel, "Multiple frequency offset compensation in cooperative wireless systems," in *Proc. IEEE Globecom'06*, San Francisco, USA, pp. 1–5, Nov. 2006.
- [18] F. Tian, X.-G. Xia, and P. C. Ching, "Signal detection for space-frequency coded cooperative communication system with multiple carrier frequency offsets," in *Proc. IEEE Wireless Communications Networking Conf. (WCNC)*, Hong Kong, pp. 1221-1225, Mar. 2007.
- [19] A. R. Hammons and H. El Gamal, "On the theory of space-time codes for PSK modulation," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 524–542, Oct. 2000.
- [20] J. Liu, J.-K. Zhang, and K. M. Wong, "On the design of minimum BER linear space-time block codes for MIMO systems equipped with MMSE receivers," *IEEE Trans. Signal Processing*, vol. 54, pp. 3147–3158, Aug. 2006.
- [21] J.-K. Zhang, J. Liu, and K. M. Wong, "Linear Toeplitz space time block codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT'05)*, Adelaide, Australia, Sept. 4-9, 2005.
- [22] Y. Shang and X.-G. Xia, "A criterion and design for space-time block codes achieving full diversity with linear receivers," in *IEEE Int. Symp. Inform. Theory (ISIT'07)*, Nice, France, 24-29, pp. 2906–2910, Jun. 2007.
- [23] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.
- [24] N. Al-Dhahir and J. M. Cioffi, "MMSE decision-feedback equalizers: finite-length results," *IEEE Trans. Inform. Theory*, vol. 41, pp. 961–975, July 1995.

- [25] G. Leus and M. Moonen, *Signal Processing for Mobile Communications Handbook*, M. Ibnkahla, Ed. New York: CRC, 2005.
- [26] G. Kaleh, "Channel equalization for block transmission systems," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 110–121, Jan. 1995.
- [27] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge Univ. Press, 1985.
- [28] S. Haykin, *Adaptive filter theory*, 4th ed. New Jersey: Prentice Hall, 2001.

TABLE I
COMPUTATIONAL COMPLEXITY OF THE PROPOSED RECURSIVE MMSE EQUALIZER AND COMPARISON OF COMPUTATIONAL COMPLEXITIES OF RECURSIVE /BLOCK ALGORITHMS

Recursive algorithm steps	Complex multiples	Complex divisions	Order
Calculate Ξ^{-1} in 2)	$3(N_f - 1)^2/2 + (N_f - 1)$	$(N_f - 1)^2/2$	
Calculate \mathbf{r}_{k+1} and r_{k+1} in 3)	$N_f(N_f - L' + 1)$		
Calculate Ψ in 4)	$3(N_f - 1)^2/2 + (N_f - 1)$	$(N_f - 1)^2/2$	
Calculate \mathbf{R}_{k+1}^{-1} in 5)	$(N_f - 1)^2 + (N_f - 1)$		
Calculate \mathbf{f}_k^{MMSE} in 5)	N_f^2		
Estimate s_{k+1}	N_f		
Total of Serial MMSE	$4(N_f - 1)^2 + N_f(N_f - L' + 1) + N_f^2 + 4N_f - 3$	$(N_f - 1)^2$	$\mathcal{O}(N_f^2 N)$
Serial MMSE-DFE	$4(N_f - 1)^2 + 2N_f^2 + N_b N_f + 4N_f + N_b - 3$	$(N_f - 1)^2$	$\mathcal{O}(N_f^2 N)$
Block MMSE	$(N + L' - 1)^2 N + N^3 + (N + L' - 1)N + N^2$		$\mathcal{O}(N^3)$
Block MMSE-DFE	$(N + L' - 1)^2 N + N^3 + N^2 + (N - L' + 1)N + 2N$		$\mathcal{O}(N^3)$

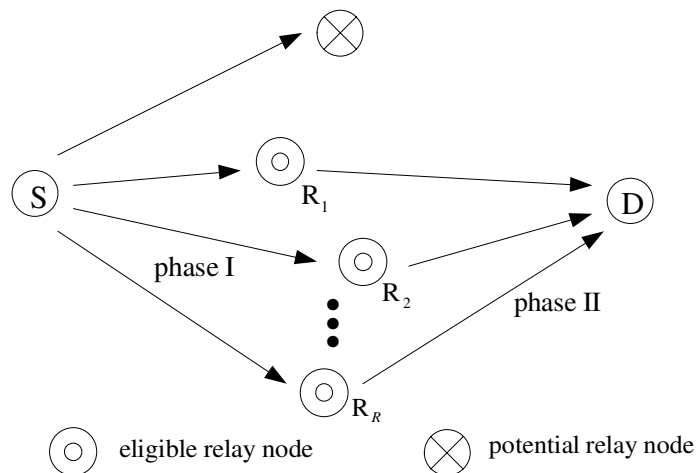


Fig. 1. Cooperative communication network model.

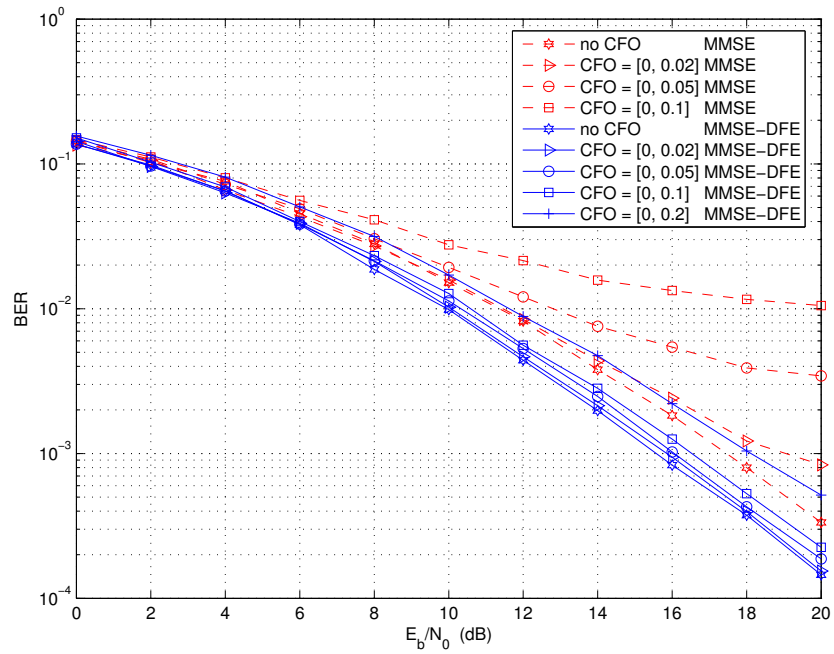


Fig. 2. BER performance vs. E_b/N_0 for two relay nodes with different CFOs.

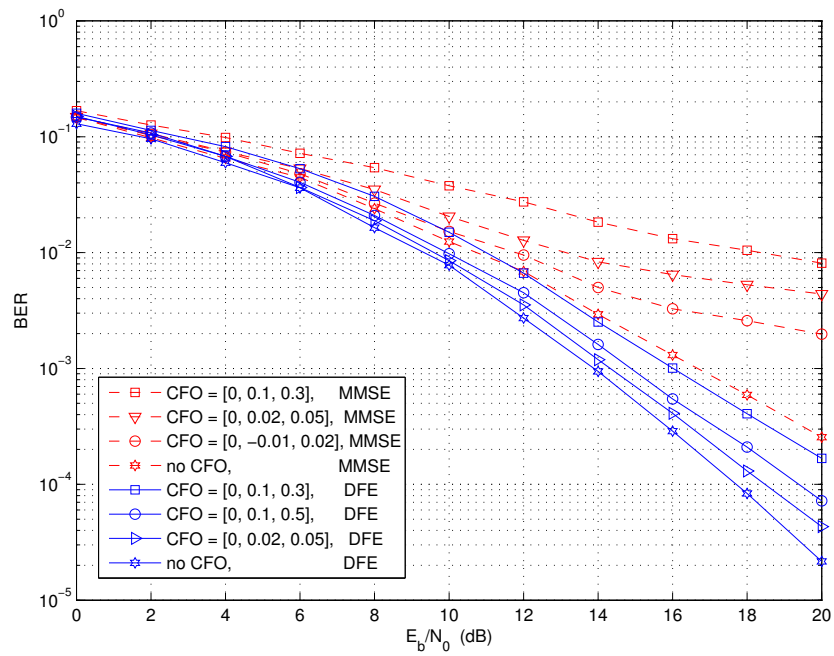


Fig. 3. BER performance vs. E_b/N_0 for three relay nodes with different CFOs.

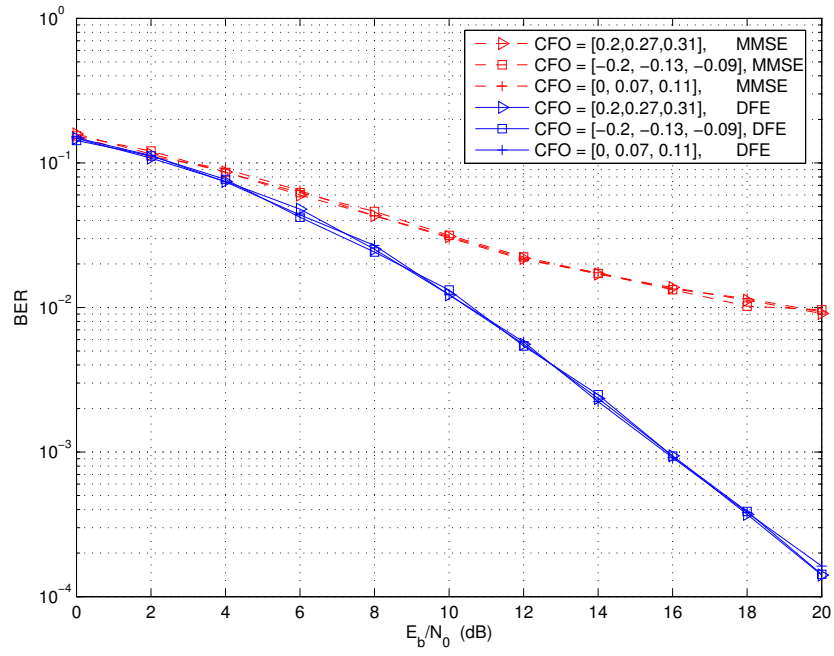


Fig. 4. The property of the MMSE/MMSE-DFE equalizer: the equalization performance is the same under two CFO sets with a constant difference.

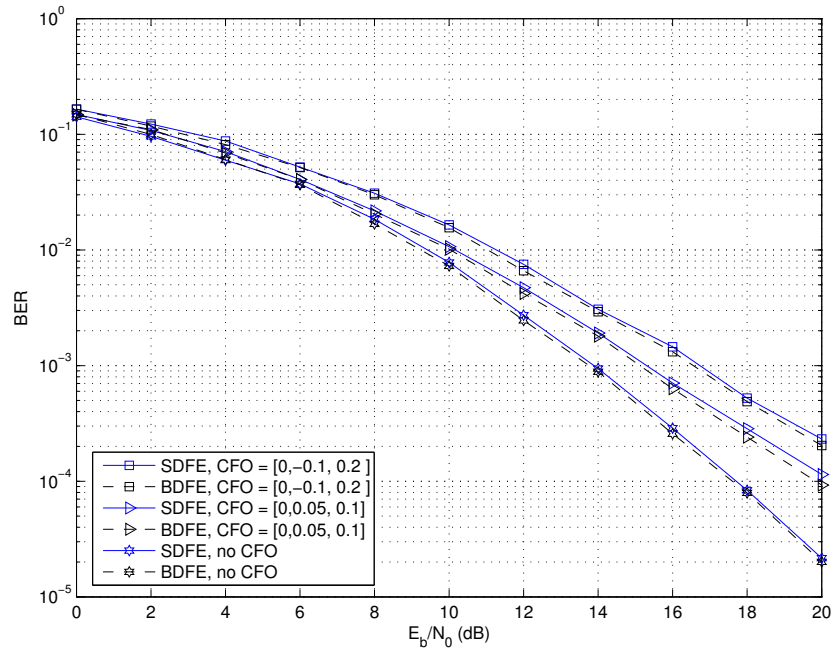


Fig. 5. The performance comparison of block MMSE-DFE equalizer and our serial MMSE-DFE equalizer.

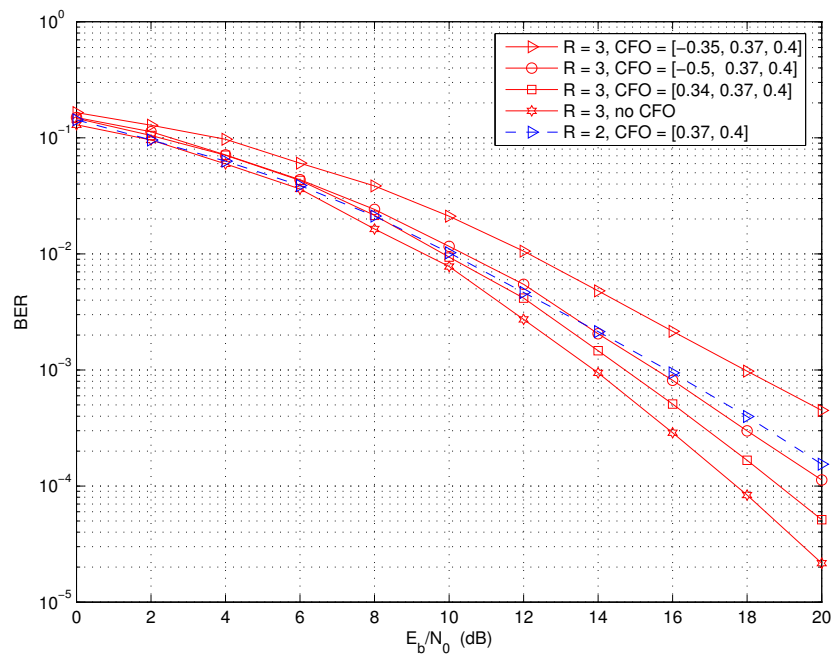


Fig. 6. Comparisons of the BER performance for different number of relay nodes and different CFOs. With different CFOs, the performance with three relay nodes may be worse than that with two relay nodes.