Enhanced Data Detection in OFDM Systems Using Factor Graph

Jindan Yang*, Qinghua Guo†*, Defeng (David) Huang* and Sven Nordholm‡
*School of EECE, The University of Western Australia, Perth, WA
†School of ECTE, University of Wollongong, Wollongong, NSW
‡Department of ECE, Curtin University, Perth, WA
Email: 20877549@student.uwa.edu.au, qinghua_guo@uow.edu.au
david.huang@uwa.edu.au, s.nordholm@curtin.edu.au

Abstract—In OFDM systems, the cyclic prefix (CP) eliminates the interblock interference and enables the use of the computationally efficient single-tap equalizers at the receiver. While posing a loss in both spectrum efficiency and power efficiency, the CP brings extra information about the data which can be used for detection. Therefore, instead of discarding the CP as in the conventional OFDM systems, this paper takes advantage of the prefix redundancy and utilizes it in the soft-input soft-output equalizer of a turbo equalization system. By using factor graph, an equalization algorithm is developed, and with proper approximation, the complexity of the proposed algorithm is reduced to $O(2R\log_2 N + \frac{4RG}{N}\log_2 G + \frac{2RG}{N})$ per data symbol for $R$ iterations, where $N$ is the length of the block and $G$ is equal to $P + L - 1$ with $P$ the CP length and $L$ the channel length. Simulation results show that the turbo equalization system converges within two iterations and the proposed equalization approach achieves significant gain compared to the conventional approach.

Index Terms—OFDM system, cyclic prefix, iterative receiver, factor graph.

I. INTRODUCTION

ORTHOGONAL frequency division multiplexing (OFDM) [1] has been used in a variety of applications and standards, such as digital audio broadcasting (DAB), digital video broadcasting (DVB), IEEE 802.11a, MMAC and HIPERLAN/2. Based on a cyclic prefix (CP) assisted scheme, these systems first insert a repetition of the end of each block into its front at the transmitter as the CP and then discard it at the receiver. The main purpose of using CP is to convert the frequency selective channel into a set of parallel frequency flat channels so that the computationally efficient single-tap equalizers can be used. On the other hand, due to the extra transmission time and energy induced by the CP, a loss occurs in both spectrum efficiency and power efficiency.

Some studies attempt to avoid the use of the CP [2], while others, considering the compatibility with the existing systems, tend to exploit the redundancy introduced by the CP instead [3], [4]. Since the CP carries information about the data, research has been carried out on utilizing it for equalization purpose. For instance, [5]–[8] were conducted in the context of non-iterative systems, while inspired by turbo equalization [9], [10]–[14] were set in iterative systems, where extrinsic information is exchanged between the equalizer and the decoder in an iterative fashion at the receiver.

More specifically, [5] assumes that the length of the CP is designed to be longer than the channel, and it uses a portion of the CP, which is free of interference from the preceding OFDM symbol, to improve the data recovery. As an extension, [6]–[8] exploit the whole CP by performing interference cancellation on the CP first. For iterative receivers, [10] utilizes the CP to jointly estimate the channel as well as the data using the EM algorithm, while [11]–[13] make use of the CP only for data detection using various methods such as the ML search, genetic algorithm and Newton’s method. The main issue with [5]–[8] is the limited performance gain, and with [10]–[13], the complexity of the algorithms is quite high. In [14], we propose a low-complexity equalization algorithm for the iterative SC-FDE systems where CP is also employed. With complexity of $O(N\log N)$ per data block per iteration where $N$ is the length of the block, a gain of 0.7dB is achieved under both AWGN and ISI channels when the CP ratio is 1/4.

In this paper, we consider the coded OFDM systems in the context of turbo equalization. By using factor graph, we propose a low-complexity equalization algorithm which exploits the redundancy induced by the CP and enhances the data detection. With proper approximation, the complexity is $O(2R\log_2 N + \frac{4RG}{N}\log_2 G + \frac{2RG}{N})$ per data symbol for $R$ iterations, where $N$ is the length of the block and $G$ is equal to $P + L - 1$ with $P$ the CP length and $L$ the channel length. Simulation results show that significant gain is achieved compared to the conventional algorithm and the turbo equalization system converges fast (within two iterations).

The rest of this paper is organized as follows. Section II describes the system models. Section III presents the models in a factor graph and an equalization algorithm is developed. Section IV shows the simulation results and Section V concludes the paper.

Notations: We use upper (lower) boldface letters to denote matrices (column vectors) and italics to denote scalars. The superscript $^*$ and $^H$ represent the transpose and the conjugate...
transpose, respectively. $F_N$ represents a normalized $N \times N$ discrete Fourier transform (DFT) matrix with the $(m,n)$th element given by $N^{-1/2}e^{-j2\pi mmn/N}$, where $j = \sqrt{-1}$. $\text{Diag}(a)$ is a diagonal matrix with the entries of $a$ on its diagonal. $(A)_{\text{diag}}$ returns a matrix with the diagonal entries of $A$ on its diagonal and zeros off its diagonal. $I$ and $0$ denote an identity matrix and an all-zero vector/matrix, with proper sizes.

II. System Model

Consider a coded OFDM system as depicted in Fig. 1. The information bits are encoded, interleaved and mapped onto a $M$-QAM symbol sequence, $\mathbf{u} = [u_0, u_1, ..., u_{Z-1}]^T$. This sequence is divided into $K$ blocks, with the $k$th block denoted by $\mathbf{u}_k = [u_{kN}, u_{kN+1}, ..., u_{(k+1)N-1}]^T$, $k = 0, ..., K-1$, where we assume $Z = KN$. Applying an $N$-point IFFT on $\mathbf{u}_k$ results in $\mathbf{x}_k = F_N^H\mathbf{u}_k$, and copying the last $P$ samples of $\mathbf{x}_k$ to its front forms the cyclic prefix (CP), $\hat{\mathbf{x}}_k = A\mathbf{x}_k$, $A = \begin{bmatrix} 0_{P \times (N-P)} & I_P \end{bmatrix}$. For the CP to be effective, $P \geq L - 1$ is required, where $L$ is the length of the multipath channel. After inserting the CP, the block is of length $M$, $M = N + P$, and denoted by $\mathbf{s}_k = [x_{(k+1)N-P}, ..., x_{(k+1)N-1}, x_{kN}, ..., x_{(k+1)N-1}]^T$. Then the sequence $\mathbf{s} = [s_0, s_1, ..., s_{J-1}]^T = [s_0^T, s_1^T, ..., s_{K-1}^T]^T$, where $J = KM$, is transmitted over the channel. At the receiver side, the observation is $\mathbf{r} = [r_0, r_1, ..., r_{V-1}]^T$, where $V = J + L - 1$, and

$$r_j = \sum_{l=0}^{L-1} h_{j,l} \cdot s_{j-l} + n_j, \quad j = 0, ..., V - 1 \quad (1)$$

where $\{h_{j,l}\}$ is the channel state information (CSI) at time $j$, $l = 0, ..., L - 1$, and $\{n_j\}$ is the additive white Gaussian noise (AWGN) with zero mean and variance $2\sigma^2$. In this paper, we assume that the channel is quasi-static, which means the channel remains the same within the duration of a block but it varies from block to block.

As shown in Fig. 2, the counterpart of $\mathbf{s}_k$ at the receiver is $\mathbf{r}_k = [r_{kM}, r_{kM+1}, ..., r_{(k+1)M-1}]^T$, which is $L - 1$ symbols longer than $\mathbf{s}_k$ due to the multipath channel. In conventional OFDM systems, the CP observation is removed and only the part $\mathbf{y}_k = [r_{kM+P}, r_{kM+P-1}, ..., r_{(k+1)M-1}]^T$ is further processed. This part can be modeled as the circular convolution of the data block $\mathbf{x}_k$ and the channel $\mathbf{h}_k$.

$$\mathbf{y}_k = \mathbf{h}_k \otimes \mathbf{x}_k + \mathbf{n}_k, \quad k = 0, ..., K - 1 \quad (2)$$

where “$\otimes$” denotes the circular convolution, $\mathbf{h}_k$ is the multipath channel that $\mathbf{x}_k$ undergoes, $\mathbf{h}_k = [h_k^0, h_k^1, ..., h_k^{L-1}]^T$, and $\mathbf{n}_k$ is the AWGN noise vector with mean $0$ and covariance $2\sigma^2I$.

The rest part of $\mathbf{r}_k$ is a combination of the CP observation and the first $L - 1$ received symbols of the next block. With length of $G = P + L - 1$, we denote this part by $\tilde{\mathbf{y}}_k = [r_{kM}, ..., r_{kM+P-1}, r_{(k+1)M}, ..., r_{(k+1)M+L-2}]^T$. Due to the fact that $\mathbf{x}_k$ is a replica of the last $P$ symbols of $\mathbf{x}_k$, this part can be modeled as

$$\tilde{\mathbf{y}}_k = \mathbf{h}_k \ast \mathbf{x}_k + \mathbf{v}_k, \quad k = 0, ..., K - 1 \quad (3)$$

with

$$\mathbf{v}_k = w_{\text{tail}}^{k-1} + w_{\text{head}}^{k+1} + \tilde{\mathbf{n}}_k \quad (4)$$

$$w_{\text{tail}}^{k-1} = [\text{tail}(\mathbf{h}_k - \mathbf{x}_k), 0, ..., 0]^T \quad (5)$$

$$w_{\text{head}}^{k+1} = [0, ..., 0, \text{head}(\mathbf{h}_k + \mathbf{x}_k)]^T \quad (6)$$

and

$$w_{\text{tail}}^{k-1} = \mathbf{0}, \quad w_{\text{head}}^{k+1} = \mathbf{0} \quad (7)$$

where “$\ast$” denotes the linear convolution, $w_{\text{tail}}^{k-1}$ and $w_{\text{head}}^{k+1}$ denote the interference from adjacent blocks $\mathbf{s}_{k-1}$ and $\mathbf{s}_{k+1}$, respectively, $\tilde{\mathbf{n}}_k$ denotes the AWGN noise vector of length $G$, and tail(·) and head(·) are two truncation functions which return the $L - 1$ tail and the $L - 1$ head symbols of the sequence in the parentheses, respectively.

As linear convolution after proper zero-padding can be transformed into circular convolution, and circular convolution in the time domain is equivalent to multiplication in the frequency domain, we can rewrite (2) and (3) as

$$\mathbf{y}_k = F_N^H \mathbf{H}_k F_N \mathbf{x}_k + \mathbf{n}_k \quad (8)$$

$$\tilde{\mathbf{y}}_k = F_N^H \tilde{\mathbf{h}}_k F_N \mathbf{Q} \tilde{\mathbf{x}}_k + \mathbf{v}_k \quad (9)$$

where $\mathbf{Q} = \begin{bmatrix} I_P & 0_{P \times (L-1)} \end{bmatrix}$ is to adapt $\tilde{\mathbf{x}}_k$ for the DFT operation by adding $L - 1$ zeros to its end, and $\mathbf{H}_k$ and $\tilde{\mathbf{h}}_k$ are the $N$-point and $G$-point DFTs of the vector $\mathbf{h}_k$, respectively,

$$\mathbf{H}_k = \text{Diag}\left(\sqrt{N}F_N \mathbf{h}_k\right), \quad \tilde{\mathbf{h}}_k = \begin{bmatrix} h_k^T & \mathbf{0}_{N-L} \end{bmatrix}^T \quad (10)$$

$$\tilde{\mathbf{h}}_k = \text{Diag}\left(\sqrt{G}F_G \mathbf{h}_k\right), \quad \tilde{\mathbf{h}}_k = \begin{bmatrix} h_k^T & \mathbf{0}_{G-L} \end{bmatrix}^T \quad (11)$$

Substituting $\mathbf{x}_k = F_N^H \mathbf{u}_k$ and $\tilde{\mathbf{x}}_k = A \mathbf{x}_k$ into (8) and (9), respectively, we get

$$\mathbf{y}_k = F_N^H \mathbf{H}_k \mathbf{u}_k + \mathbf{n}_k \quad (12)$$

$$\tilde{\mathbf{y}}_k = F_G^H \tilde{\mathbf{h}}_k \mathbf{B} \mathbf{u}_k + \mathbf{v}_k \quad (13)$$
with $B = F_G Q A F_N^H$. The term $v_k$ can also be written in a similar form as

$$
v_k = \tilde{n}_k + C F^H \tilde{H}_{k-1} B u_{k-1} + C H F^H \tilde{H}_{k+1} B u_{k+1} \tag{14}
$$

where the matrix $C$ and its Hermitian transpose $C^H$ function as tail ($\cdot$) and head ($\cdot$), respectively, and

$$
C = \begin{bmatrix}
0_{(L-1) \times P} & I_{L-1} \\
0_P & 0_{P \times (L-1)}
\end{bmatrix}. \tag{15}
$$

III. FACTOR GRAPH BASED EQUALIZATION ALGORITHMS

In this section, we present the system models in a Forney-style factor graph (FFG) [15], where an edge or a half-edge represents a variable and a node represents a factor (or a function). Variables in the graph of this paper are all vectors, and messages about them are assumed to be Gaussian so only the mean vector and covariance matrix are needed to represent each message. To update these messages, the Gaussian message passing (GMP) technique from [16] is employed, along with the computation rules from [15].

The notation system associated with an FFG is as follows. Let $x$ be a vector variable, represented by a directed edge, i.e., an edge with an arrow, in an FFG. We use $\tilde{m}_x$ and $\tilde{V}_x$ to denote the mean vector and the covariance matrix of the message that flows in the direction as the arrow indicates, which is called the forward message, while $\tilde{m}_x$ and $\tilde{V}_x$ are for the message in the opposite direction, called the backward message. The marginal message is the product of these two messages and denoted by $m_x$ and $V_x$.

As illustrated in Fig. 3, the right branch and the left branch represent the model (12) and the model (13), respectively. Since the conventional equalizer only utilizes the $y_k$ part of the observation, it corresponds to the right branch of the graph only; whereas the proposed equalizer makes use of both the $y_k$ and $\hat{y}_k$ parts of the received signal, therefore both branches of the graph.

A. Message passing for the conventional equalizer

1) As the noise is with mean zero and variance $2\sigma^2$, based on rules (II.10) and (II.8) in [15], the backward message at point $e_k$ is

$$
\tilde{m}_{e_k} = y_k, \quad \tilde{V}_{e_k} = \beta I_N \tag{16}
$$

where $\beta = 2\sigma^2$. According to rules (III.6) and (III.5) in [15], the backward message at point $b_k$ is

$$
\tilde{W}_{b_k} \tilde{m}_{b_k} = \beta^{-1} H_k^H F_N y_k, \quad \tilde{W}_{b_k} = \beta^{-1} H_k^H H_k \tag{17}
$$

where $\tilde{W}_{b_k}$ is the weight matrix and $\tilde{W}_{b_k} = \tilde{V}_{b_k}^{-1}$.  

2) Combining the backward message at point $b_k$ with the $a$ priori message of $u_k$, we have the $a$ posteriori message or marginal message of $u_k$ as

$$
\begin{align*}
\tilde{W}_{u_k} m_{u_k} &= \tilde{W}_{b_k} \tilde{m}_{b_k} + \tilde{V}_{a_k}^{-1} \tilde{m}_{a_k}, \tag{19} \\
\tilde{W}_{u_k} &= \tilde{W}_{b_k} + \tilde{V}_{a_k}^{-1}. \tag{20}
\end{align*}
$$

For the $a$ priori information, we set $\tilde{m}_{a_k} = 0$ and $\tilde{V}_{a_k} = I$ at the first iteration, and at other iterations, its covariance matrix $\tilde{V}_{a_k}$ is modeled as a diagonal matrix since symbols within a block are independent of each other thanks to the use of the interleaver.

3) To extract the extrinsic information of each symbol in $u_k$, we use the relations (14) and (15) in [18]. Since $\tilde{V}_{b_k}$ and $\tilde{V}_{a_k}$ are diagonal, the extrinsic variances and means of the data symbols in the $k$th block, $k = 0, \ldots, K-1$, can be given by

$$
\begin{align*}
V_{k}^{ext} &= \left[ \tilde{W}_{u_k} - \tilde{V}_{a_k}^{-1} \right]^{-1} = \tilde{V}_{b_k}, \tag{21} \\
m_k^{ext} &= V_k^{ext} \left[ \tilde{W}_{u_k} m_{u_k} - \tilde{V}_{a_k}^{-1} \tilde{m}_{a_k} \right] = \tilde{m}_{b_k}. \tag{22}
\end{align*}
$$

B. Message passing for the proposed equalizer

1) The forward message at point $e_k$ is the same as the $a$ posteriori message of $u_k$ obtained from the right branch. Using (19), (20) and (17), (18), we have

$$
\begin{align*}
\tilde{W}_{a_k} \tilde{m}_{a_k} &= \tilde{V}_{a_k}^{-1} \tilde{m}_{a_k} + \beta^{-1} H_k^H F_N y_k, \tag{23} \\
\tilde{W}_{c_k} &= \tilde{V}_{a_k}^{-1} + \beta^{-1} H_k^H H_k. \tag{24}
\end{align*}
$$

2) Based on rules (II.10) and (II.8) in [15], the backward message at point $e_k$ is

$$
\tilde{m}_{e_k} = \hat{y}_k - \tilde{m}_{v_k}, \quad \tilde{V}_{e_k} = \tilde{V}_{v_k}. \tag{25}
$$

The mean vector of $v_k$ in (14) is updated by $\tilde{m}_{a_{k-1}}$ and $\tilde{m}_{a_{k+1}}$, which are the latest mean vectors about $u_{k-1}$ and $u_{k+1}$,

$$
\tilde{m}_{v_k} = C F^H e_{k-1} + C H F^H e_{k+1} \tag{26}
$$

The message passing presented here is only an alternative description of the conventional soft-output equalization algorithm [17], without changing its working principles.
with $\xi_k = \tilde{H}_k \mathbf{M}m_{c_k}$. The covariance matrix is approximated by \(^2\)

$$
\tilde{\mathbf{V}}_{v_k} = \lambda_k \mathbf{I}_G
$$

(27)

where

$$
\begin{align*}
\lambda_k &= 2\sigma^2 + \frac{1}{G} \sum_{i=0}^{L-1} \left[ \gamma_{k+1}(L - 1 - i) \left| h_{k+1}^i \right|^2 \
+ \gamma_{k-1} \left| h_{k-1}^i \right|^2 \right] , \\
\gamma_k &= \frac{1}{N} \sum_{i=0}^{N-1} (\tilde{\mathbf{W}}_{c_k})_{i,i} .
\end{align*}
$$

(28)

(29)

Therefore, according to rules (III.6) and (III.5) in [15], the backward message at point $c_k$ is given by

$$
\begin{align*}
\tilde{\mathbf{W}}_{c_k} m_{c_k} &= \lambda_k^{-1} \mathbf{B}^H \tilde{\mathbf{H}}_k^H \mathbf{F}_G (\tilde{y}_k - \tilde{m}_{v_k}) , \\
\mathbf{W}_{c_k} &= \lambda_k^{-1} \mathbf{1}_k
\end{align*}
$$

(30)

(31)

where $\mathbf{W}_k = \mathbf{B}^H \tilde{\mathbf{H}}_k^H \tilde{\mathbf{H}}_k \mathbf{B}$.

3) For the a posteriori message at point $c_k$, the covariance matrix is

$$
\begin{align*}
\mathbf{V}_{c_k} &= \left( \mathbf{W}_{c_k} + \tilde{\mathbf{W}}_{c_k} \right)^{-1} \\
&= \left( \mathbf{V}_{a_k}^{-1} + \beta^{-1} \mathbf{H}_k^H \mathbf{H}_k + \lambda_k^{-1} \mathbf{W}_k \right)^{-1} .
\end{align*}
$$

(32)

For the mean vector, we substitute rule (I.1) into rule (I.6) in [15], and get $\mathbf{m} = \tilde{\mathbf{m}} + \mathbf{V} (\tilde{\mathbf{W}} \tilde{\mathbf{m}} - \tilde{\mathbf{W}} \mathbf{m})$. Thus, the a posteriori mean vector at point $c_k$ is

$$
\begin{align*}
\mathbf{m}_{c_k} &= \tilde{\mathbf{m}}_{c_k} + \mathbf{V}_{c_k} (\mathbf{W}_{c_k} \tilde{\mathbf{m}}_{c_k} - \tilde{\mathbf{W}}_{c_k} \tilde{\mathbf{m}}_{c_k}) \\
&= \tilde{\mathbf{m}}_{c_k} + \lambda_k^{-1} \mathbf{V}_{c_k} \mathbf{B}^H \tilde{\mathbf{H}}_k^H \\
&\quad \cdot \left[ \mathbf{F}_G (\tilde{y}_k - \tilde{m}_{v_k}) - \mathbf{H}_k \mathbf{B} \tilde{\mathbf{m}}_{c_k} \right] .
\end{align*}
$$

(33)

According to rules (II.5) and (II.6) in [15], the a posteriori message of $\mathbf{u}_k$ is

$$
\mathbf{m}_{u_k} = \mathbf{m}_{c_k} , \quad \mathbf{V}_{u_k} = \mathbf{V}_{c_k} .
$$

(34)

4) To calculate the extrinsic information of each data symbol, we extract the a posteriori mean/variance of the symbol, and subtract its a priori mean/variance using the relationships (14) and (15) in [18] as follows.

$$
\begin{align*}
\mathbf{V}_{k}^{ext} &= \left( \mathbf{V}_{u_k} \right)_{diag} - \tilde{\mathbf{V}}_{a_k}^{-1} , \\
\mathbf{m}_{k}^{ext} &= \mathbf{V}_{k}^{ext} \left[ \left( \mathbf{V}_{u_k} \right)_{diag} \mathbf{m}_{u_k} - \tilde{\mathbf{V}}_{a_k}^{-1} \tilde{\mathbf{m}}_{a_k} \right] .
\end{align*}
$$

(35)

(36)

Note that the a posteriori covariance matrix in this case is not diagonal, and that the a posteriori variances of the data symbols are given by the diagonal entries of the a posteriori covariance matrix.

\(^2\)This approximation can be found in other works, such as [19]. The impact of this approximation diminishes gradually as $\lambda_k$ becomes smaller and smaller with the iteration.

<table>
<thead>
<tr>
<th>Variables/Terms</th>
<th>Complexity per data symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{H}_k$, $\mathbf{H}_k$</td>
<td>$O(\log_2 N + \frac{1}{2} \log_2 G)$</td>
</tr>
<tr>
<td>$\mathbf{F}_k$</td>
<td>$O(\log_2 N + \frac{1}{2} \log_2 G)$</td>
</tr>
<tr>
<td>$\mathbf{H}_k^H \mathbf{F}_N \mathbf{y}_k$</td>
<td>$O(\log_2 N + 1)$</td>
</tr>
<tr>
<td>$\xi_k$</td>
<td>$O(R \log_2 N + \frac{1}{2} \log_2 G + \frac{3N}{2} R)$</td>
</tr>
<tr>
<td>$\mathbf{m}_{k}^{ext}$</td>
<td>$O(R \log_2 N + \frac{1}{2} \log_2 G + \frac{3N}{2} R)$</td>
</tr>
</tbody>
</table>

C. Complexity reduction

For the proposed algorithm, due to the existence of the full matrix $\Psi_k$ in (32), the calculation of $\mathbf{V}_{u_k}$ involves matrix inversion which is computationally demanding. To reduce the complexity, we make the following approximation (see (32) and (34))

$$
\mathbf{V}_{u_k} = \left( \tilde{\mathbf{W}}_{c_k} + \lambda_k^{-1} \mathbf{F}_k \right)^{-1}
$$

(37)

where $\Phi_k = (\Psi_k)_{diag}$. As a result, $\mathbf{V}_{u_k}$ becomes a diagonal matrix and the matrix inversion that originally occurs is turned into scalar reciprocals. Also, the calculation of $\mathbf{m}_{u_k}$ becomes (see (33) and (34))

$$
\begin{align*}
\mathbf{m}_{u_k} &= \tilde{\mathbf{m}}_{c_k} + \lambda_k^{-1} \left( \mathbf{W}_{c_k} + \lambda_k^{-1} \mathbf{F}_k \right)^{-1} \mathbf{B}^H \tilde{\mathbf{H}}_k^H \\
&\quad \cdot \left[ \mathbf{F}_G (\tilde{y}_k - \mathbf{C}^H \xi_k - \mathbf{C}^H \mathbf{F}_k \xi_{k+1} - \xi_k) \right] .
\end{align*}
$$

(38)

Substituting (37) and (38) into (35) and (36), respectively, we have the extrinsic variances and means of the data symbols as

$$
\begin{align*}
\mathbf{V}_{k}^{ext} &= \left( \beta^{-1} \mathbf{H}_k^H \mathbf{H}_k + \lambda_k^{-1} \mathbf{F}_k \right)^{-1} , \\
\mathbf{m}_{k}^{ext} &= \mathbf{V}_{k}^{ext} \left[ \beta^{-1} \mathbf{H}_k^H \mathbf{F}_N \mathbf{y}_k + \lambda_k^{-1} \mathbf{F}_k \tilde{\mathbf{m}}_{c_k} + \lambda_k^{-1} \mathbf{B}^H \tilde{\mathbf{H}}_k^H \\
&\quad \cdot \left[ \mathbf{F}_G (\tilde{y}_k - \mathbf{C}^H \xi_k - \mathbf{C}^H \mathbf{F}_k \xi_{k+1} - \xi_k) \right] .
\end{align*}
$$

(39)

(40)

With this approximation the complexity of the proposed algorithm is listed in Table I, where the complex multiplications are counted for $R$ iterations. Dominated by the calculation of $\xi_k$ and $\mathbf{m}_{k}^{ext}$ which needs to be done every block every iteration, the overall complexity is $O(2R \log_2 N + \frac{4RG}{N} \log_2 G + \frac{2RG}{N} R)$ per data symbol for $R$ iterations.

IV. Simulation Results

In the simulation, a rate-1/2 convolutional code (23,35), a S-random interleaver, a 4QAM modulator with Gray mapping and the BCJR-based decoder are employed. Each data block has 64 symbols before CP insertion, i.e. $N = 64$, and with 1/4 ratio, the CP length is 16, i.e. $P = 16$. Three types of channels are simulated, AWGN channel, Proakis C channel, and a random 17-tap channel. For the former two, each block experiences exactly the same channel. For the random 17-tap channel, coefficients are generated independently for each block. The energy of the channel is normalized to 1, i.e. $\frac{1}{L} \sum_{i=0}^{L-1} | h_{k}^i |^2 = 1$, for $k = 0, \ldots, K - 1$, and uniform power
delay profile applies. Perfect CSI is assumed available at the receiver.

Fig. 4 compares the BER performance of the proposed algorithm and the conventional algorithm under AWGN channel and the random 17-tap channel. For the conventional algorithm, the BER performance does not improve with the iteration and only the curve for the first iteration is shown. For the proposed algorithm, the BER curves decline dramatically with the iteration and converge within two iterations. At the BER of $10^{-5}$, a gain of 0.97dB is obtained compared to the conventional algorithm under AWGN channel, and a 2dB gain under the random 17-tap channel. As the channel deteriorates, an even greater gain is observed as in Fig. 5, where the proposed algorithm outperforms the conventional algorithm by 6dB at the BER of $10^{-5}$ under the Proakis C channel.

V. CONCLUSION

We have proposed a low-complexity equalization algorithm using factor graph, which takes advantage of the CP observation and enhances the data detection in the OFDM systems. Models for both the CP and non-CP parts of observation are presented in an FFG and the message passing process is described in detail for the FFG. Proper approximation is made to reduce the complexity. Simulation results are provided to verify the performance improvement.

REFERENCES


