# Turbo Equalization and its Applications

Jindan Yang

*This thesis is presented for the degree of Doctor of Philosophy*

*School of Electrical, Electronic and Computer Engineering*

THE UNIVERSITY OF WESTERN AUSTRALIA

*Achieve International Excellence*

*November 2014*

# Abstract

In communication systems, the transmitted signals often experience various forms of corruption, and a very destructive one is the inter-symbol interference (ISI), resulting from the signals travelling through multiple paths and arriving with different delays. To compensate for the multipath propagation effect, equalization is often required at the receiver and as a very powerful candidate, the turbo equalization technique iteratively exchanges soft information between the equalizer and the decoder, offering excellent performance.

Meantime, single-carrier frequency domain equalization (SC-FDE) and orthogonal frequency division multiplexing (OFDM) have become two of the most important modulation schemes in communications. Both schemes are based on guard interval assisted block transmissions and both can be implemented by fast Fourier transform/inverse fast Fourier transform (FFT/IFFT) operations, featuring very low complexity. Cyclic prefix (CP), as a commonly used guard interval, is typically discarded at the receiver. However, the fact that it is a repetition of part of the data suggests that it contains useful information and can be exploited for enhancing the system performance.

This thesis considers the application of turbo equalization in SC-FDE and OFDM systems and based on the linear minimum mean square error (LMMSE) principle, various equalization algorithms are proposed with the aim of exploiting the prefix redundancy for data detection and channel estimation. To facilitate the description of system models and to develop low-complexity algorithms, Forney-style factor graphs (FFG) and the Gaussian message passing (GMP) technique in factor graphs are employed.

Compared to the conventional CP-discarding schemes, the proposed CP-exploiting schemes provide significant improvement in system performance, while exhibiting low complexity. For the iterative SC-FDE system, a gain of 0.7dB is achieved under both the additive white Gaussian noise (AWGN) channel and the ISI channel when the CP ratio is 1/4 and the channel is perfectly known at the receiver, with complexity of $\mathcal{O}(NlogN)$ per data block per iteration, where $N$ is the length of the data block.

For the turbo OFDM system, when the channel is known, a gain of 0.97dB under the AWGN channel and 2dB under the ISI channel are achieved through making use of the CP for data detection. Approximations are discussed and a reduced-complexity version of the proposed algorithm is developed, with complexity of $\mathcal{O}(NlogN)$ per data block per iteration. A thorough SNR analysis is provided to verify the performance improvement, and the sensitivity of the proposed reduced-complexity algorithm to channel estimation error is also examined.

When the channel is unknown, a joint channel estimation, data detection and decoding approach is proposed for the turbo OFDM system, using the message passing expectation maximization (EM) algorithm. After convergence, a minimum square error of $10^{-4}$ is achieved for the channel estimation performance and the data detection performance is shown to be close to that when the channel is known. Instead of only using the means as in the original message passing EM algorithm, both the means and variances of the channel estimates are accommodated in the system models, which greatly improves the system performance at early iterations. Discussion of the different ways of using CP shows that using CP for data detection is more beneficial than using it for channel estimation, considering the performance gain that can be obtained and the complexity cost that comes with it. A novel scheduling scheme is proposed based on that discussion, offering an attractive tradeoff between performance and complexity.

With the popularity of turbo equalization and the SC-FDE and OFDM systems, the proposed algorithms in this thesis are believed to be very useful in practice. Also, with a general nature, the proposed algorithms can be extended to other communication systems where similar redundancy is present and exploitation of the redundancy is reasonable.

# ABSTRACT

# List of Publications

[1] J. Yang, Q. Guo, D. Huang, and S. Nordholm, "Exploiting cyclic prefix in Turbo FDE systems using factor graph," in *IEEE Wireless Communications and Networking Conference (WCNC), 2013*, pp. 2536-2541.

[2] J. Yang, Q. Guo, D. Huang, and S. Nordholm, "Enhanced data detection in OFDM systems using factor graph," in *IEEE International Conference on Wireless Communications and Signal Processing (WCSP), 2013*, pp. 1-5.

[3] J. Yang, Q. Guo, D. Huang, and S. Nordholm, "A factor graph approach to exploiting cyclic prefix for equalization in OFDM systems," in *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 4972-4983, 2013.

[4] J. Yang, Q. Guo, D. Huang, and S. Nordholm, "Exploiting cyclic prefix for joint detection, decoding and channel estimation in OFDM via EM algorithm and message passing," in *IEEE International Conference on Communications (ICC), 2014*, accepted.

[5] J. Yang, Q. Guo, D. Huang, and S. Nordholm, "Exploiting cyclic prefix for iterative OFDM receiver design via message passing based EM algorithm," in *IEEE Trans. Commun.*, submitted.

# Acknowledgements

I would like to express my gratitude to my supervisors, Dr. David (Defeng) Huang and Dr. Qinghua Guo for their support during the course of my studies. This thesis would have been impossible without their enthusiasm, dedication and assistance.

I would also like to thank my colleagues in the Signal Processing Wireless Communication Laboratory (SPWCL) research group at the University of Western Australia. Their insightful discussions and social activities have been invaluable and have greatly enriched my learning experience.

Most importantly, I thank my family for their understanding and support throughout the course of my studies and I am forever grateful.

I acknowledge that this work was supported by iVEC through the use of advanced computing resources located at iVEC@Murdoch.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **AR** | Auto Regressive |
| **AWGN** | Additive White Gaussian Noise |
| **BER** | Bit Error Rate |
| **CIR** | Channel Impulse Response |
| **CP** | Cyclic Prefix |
| **CSI** | Channel State Information |
| **DAB** | Digital Audio Broadcasting |
| **DFE** | Decision Feedback Equalizer |
| **DFT** | Discrete Fourier Transform |
| **DVB** | Digital Video Broadcasting |
| **ECC** | Error Correcting Code |
| **EM** | Expectation Maximization |
| **FDE** | Frequency-Domain-Equalization |
| **FDMA** | Frequency Division Multiple Access |
| **FFG** | Forney-style Factor Graph |
| **FFT** | Fast Fourier Transform |
| **GMP** | Gaussian Message Passing |
| **IBI** | Inter-Block Interference |

| | |
|---|---|
| **IC** | Interference Cancellation |
| **IDFT** | Inverse Discrete Fourier Transform |
| **IFFT** | Inverse Fast Fourier Transform |
| **ISI** | Inter-Symbol Interference |
| **LDPC** | Low Density Parity Check code |
| **LLR** | Log-Likelihood Ratio |
| **LMMSE** | Linear Minimum Mean Square Error |
| **LS** | Least Square |
| **LTE** | Long Term Evolution |
| **MAP** | Maximum *a posteriori* |
| **MIMO** | Multiple-Input Multiple-Output |
| **ML** | Maximum Likelihood |
| **MMSE** | Minimum Mean Square Error |
| **MRC** | Maximum Ratio Combining |
| **MSE** | Mean Square Error |
| **NRNSC** | Non-Recursive Non-Systematic Convolutional code |
| **OFDM** | Orthogonal Frequency Division Multiplexing |
| **PAPR** | Peak to Average Power Ratio |
| **QAM** | Quadrature Amplitude Modulation |
| **QoS** | Quality of Service |
| **SC-FDE** | Single-Carrier Frequency-Domain-Equalization |
| **SISO** | Single-Input Single-Output |
| **SNR** | Signal to Noise Ratio |

| | |
|---|---|
| **SOVA** | Soft Output Viterbi Algorithm |
| **WSSUS** | Wide Sense Stationary Uncorrelated Scattering |
| **ZF** | Zero Forcing |

# Chapter 1

# Introduction

In communication systems, the transmitted signal often experience various forms of corruption, including path loss, noise and fading [1,2]. Path loss refers to the power reduction of an electromagnetic wave as it travels through space. Noise includes the ambient noise generated by the atmosphere, and the thermal noise arising from the electronic components such as the amplifiers at the receiver. Fading is the attenuation of the transmitted signal in amplitude and phase, caused by multipath propagation or time variation of the channel.

Multipath propagation occurs when the transmitted signal arrives at the receiver via multiple paths at different delays, resulting in adjacent symbols interfering with each other at the receiver, commonly referred as inter-symbol interference (ISI). Equivalently, the coherence bandwidth of the channel is smaller than the bandwidth of the signal in frequency domain, and the fading effects vary among different frequency components of the signal, known as frequency-selective fading. The distortion of the transmitted signal caused by the ISI in time domain or the frequency selectivity of the channel in frequency domain is a major concern for modern communication systems, and equalizers are often deployed to compensate for the multipath propagation effects.

The commonly used equalizers in communications are divided into two categories.

1. The linear filtering based equalizers [3]. For example, the zero forcing (ZF) equalizer applies the inverse of the frequency response of the channel to the received signals. The minimum mean square error (MMSE) equalizer minimizes the variance of the difference between the transmitted signal and the signal at the equalizer output. The decision feedback equalizer (DFE) uses the feedback of the detected symbols and adds a filtered version of previous symbol estimates to the original filter output.
2. The trellis based equalizers [4]. For example, the Viterbi equalizer finds the sequence that maximizes the maximum likelihood (ML) probability, and the MAP equalizer finds the symbol that maximizes the maximum *a posteriori* (MAP) probability.

These aforementioned equalizers only take into account the channel information, but as the focus of this thesis, the turbo equalizer also considers the coding information, and through iteration the equalizer refines its estimates of data symbols by taking in the soft extrinsic information from the decoder.

## 1.1 Turbo Equalization

Consider a communication system as depicted in Fig. 1.1, where the error correction code (ECC) adds redundancy to the information bits, the interleaver permutes the coded bits and breaks long error bursts into short ones, and the mapper maps the interleaved coded bits into complex symbols which contain more data and are more spectrally efficient.

Optimal receiver as seen in Fig. 1.1(b) jointly takes all the factors into account including the channel coding, interleaving, mapping and the channel information, and based on different criteria, it makes decision based on the MAP probability of

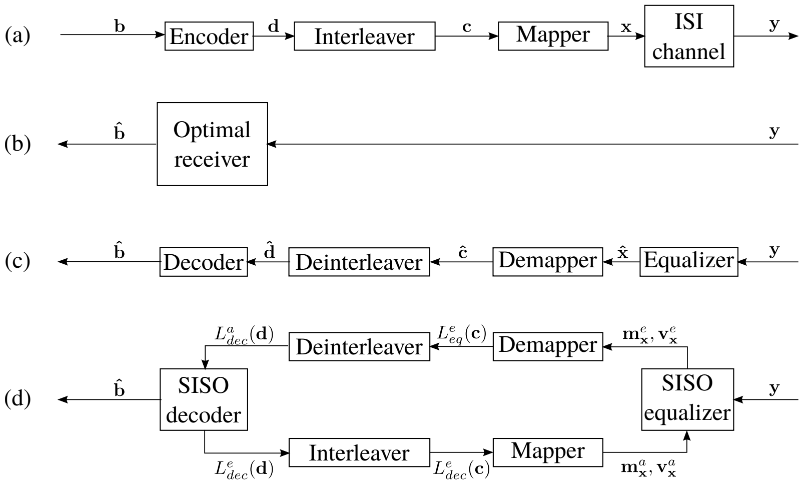


**Figure 1.1:** Transceiver structure of the turbo equalization system. (a) Transmitter and the channel (b) Optimal receiver (c) A contemporary receiver with one-time separate equalization and decoding (d) Turbo receiver

the sequence or the bit. The optimal sequence receiver calculates the most likely sequence $\hat{\mathbf{b}}$ that maximizes the probability $P(\mathbf{b} = \hat{\mathbf{b}}|\mathbf{y})$, and the optimal bit receiver calculates the most likely bit $\hat{b}_i$ that maximizes the probability $P(b_i = \hat{b}_i|\mathbf{y})$, $i = 0, ..., Z-1$. As the length of $\mathbf{b}$ grows, the computational complexity of both optimal receivers rapidly becomes intractable. In an attempt to reduce the complexity, traditional receivers split the task into smaller ones, and equalization and decoding are performed separately, as shown in Fig. 1.1(c). This sub-optimal approach ignores the dependence between the coding, interleaving, mapping and the channel and thus yields significant loss in performance [5].

In 1993, turbo code was introduced by Berrou *et al.* [6], where at the receiver, soft information is exchanged iteratively between two decoders and a near-Shannon performance is delivered. Inspired by the turbo codes, turbo equalization [5, 7–11] was proposed as a new type of receiver design, where the channel is viewed as a rate-1 convolutional code and the receiver is considered as a similar problem to turbo decoding. The turbo receiver consists of a soft-input soft-output (SISO) equalizer and a SISO decoder, interconnected by a de-mapper/mapper and a de-interleaver/interleaver. In analogy to turbo decoding, the SISO equalizer is the decoder that "decodes" the channel distortion and the SISO decoder is the one that "decodes" the channel code. Soft extrinsic information is exchanged iteratively between the SISO equalizer and the SISO decoder in the form of log-likelihood ratios (LLRs) of the coded bits. The de-mapper/mapper and the de-interleaver/interleaver in between facilitate the exchange, by performing bit-symbol conversions and permuting the coded bit LLRs to the right order, respectively. It is shown that through iterative processing, the turbo receiver offers excellent performance while exhibiting much lower complexity than the optimal receivers, making it a strong candidate for the future communication techniques [5, 7–11].

Early designs of turbo receivers employ the soft-output Viterbi algorithm (SOVA) or the MAP algorithm in the equalizer, such as in [12] and [13]. The SOVA algorithm [14, 15] is a modified SISO version of the Viterbi algorithm (VA) [16] which can take soft input but only outputs hard decisions. Both VA and SOVA algorithms perform maximum likelihood sequence detection and outputs an estimate of the symbol sequence, $\hat{\mathbf{x}}$, which maximizes $P(\mathbf{y}|\mathbf{x} = \hat{\mathbf{x}})$. The MAP algorithm [17, 18] performs MAP symbol detection and outputs an estimate of each symbol, $\hat{x}_i$, which maximizes the probability $P(x_i = \hat{x}_i|\mathbf{y})$. In practice, two variations of the MAP algorithm are more preferable, namely the Log-MAP algorithm and the Max-Log-MAP algorithm [19,20]. These two variations are numerically more stable, and require less computation effort since calculations are carried out in logarithmic domain and some nested multiplications are replaced by additions.

Both the SOVA equalizer and the MAP equalizer are trellis based and when the constellation size $M$ is large and/or the channel impulse response $L$ is long, the number of the trellis states $M^L$ grows exponentially and the computational complexity becomes prohibitively high. Consequently, alternative techniques have been proposed, for instance, [7] replaces the MAP equalizer with a soft interference canceler (IC) and [9, 10] replace it with a linear MMSE (LMMSE) equalizer. The turbo receiver in [9, 10] shows a close performance to the receiver with the MAP equalizer, but requires much lower complexity, offering an effective trade-off between the complexity and performance. In this thesis, we adopt the MMSE-equalizer based turbo receiver in [9, 10].

Before we describe the transceiver structure of the considered turbo equalization system, the notation systems used in this thesis are as follows.

**Notations:**

Upper (lower) boldface letters denote matrices (column vectors) and italics denote scalars. $(\mathbf{a})_i$ and $(\mathbf{A})_{i,j}$ denote the $i$th entry of vector $\mathbf{a}$ and the $(i, j)$th entry of matrix $\mathbf{A}$, respectively. The superscripts $^T$, $^*$ and $^H$ denote the transpose, conjugate and conjugate transpose, respectively. $\mathbf{F}_N$ denotes a normalized $N \times N$ discrete Fourier transform (DFT) matrix with the $(m, n)$th element given by $\frac{1}{\sqrt{N}} e^{-j2\pi mn/N}$, where $j = \sqrt{-1}$. $Diag\,(\mathbf{a})$ denotes a diagonal matrix with the entries of $\mathbf{a}$ on its diagonal. $\big(\mathbf{A}\big)_{diag}$ returns a matrix with the diagonal entries of $\mathbf{A}$ on its diagonal and zeros off its diagonal. $\mathrm{tr}(\cdot)$ denotes the trace of the matrix in the parentheses. $\mathbf{I}$ and $\mathbf{0}$ denote an identity matrix and an all-zero vector/matrix, with proper sizes. $(\cdot)_{\{\cdot\}}$ returns a vector with the $\{\cdot\}$th entries of the vector in $(\cdot)$ if it is a vector in $(\cdot)$, or a diagonal matrix with the $\{\cdot\}$th diagonal entries of the matrix in $(\cdot)$ on its diagonal if it is a matrix in $(\cdot)$.

### 1.1.1 Transmitter

As shown in Fig. 1.1(a), let $\mathbf{b}$ be an information bit sequence, $\mathbf{b} = \big[b_0, ..., b_{Z-1}\big]^T$ of length $Z$. With coding rate $r$, the encoder outputs the coded bit sequence

$\mathbf{d} = \big[d_0, ..., d_{U-1}\big]^T$ of length $U = Z/r$. The interleaved coded bit sequence is $\mathbf{c} = \prod(\mathbf{d})$, where $\prod(\cdot)$ denotes the interleaving operation. Assuming an $M$-ary symbol constellation is used and $U$ is a multiple of $Q = log_2 M$, the bit sequence $\mathbf{c}$ is partitioned into $N$ subsequences, $\mathbf{c} = \big[\mathbf{c}_0, ..., \mathbf{c}_{N-1}\big]^T$, where $N = U/Q$. Each subsequence $\mathbf{c}_n = \big[c_{n,0}, ..., c_{n,Q-1}\big]^T$, $n = 0, ..., N-1$, is mapped to a symbol $x_n$, where $x_n \in \boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}} = \big\{\mathcal{A}_i, i = 0, ..., M-1\big\}$ is the symbol constellation. Each constellation point $\mathcal{A}_i$ corresponds to a binary vector $\mathbf{a}_i = \big[a_{i,0}, ..., a_{i,Q-1}\big]^T$, and when $\mathbf{c}_n = \mathbf{a}_i$, we have $x_n = \mathcal{A}_i$. After the mapping, the symbol sequence $\mathbf{x} = \big[x_0, ..., x_{N-1}\big]^T$ is formed and will be transmitted over the channel.

Without loss of generality, this thesis employs the non-recursive non-systematic convolutional (NRNSC) codes [21], the S-random interleaving [22] and the QAM modulation with Gray mapping [23]. The encoder starts with the zero state and by feeding $K$ zeros bits into the encoder at the tail of each sequence $\mathbf{b}$, the encoder is also terminated at the zero state, where $K$ is the number of registers of the encoder. The $K$ tailing bits may have the effect of improving the overall bit error rate (BER) slightly because it is known at the receiver that the $K$ tail bits are zero and the last trellis state is also zero. But this effect is negligible since the length of $\mathbf{b}$ is normally much greater than $K$, i.e. $Z \gg K$. Also, the $M$ constellation points are assumed to occur equally likely and the symbol energy is normalized to 1, i.e. $\frac{1}{M}\sum_{i=0}^{M-1}\mathcal{A}_i = 0$ and $\frac{1}{M}\sum_{i=0}^{M-1}|\mathcal{A}_i|^2 = 1$.

Note that the transmitter structure shown in Fig. 1.1(a) is a transmitter for a typical coded system. When the SC-FDE and OFDM systems are considered later in Chapter 2, 3 and 4, modules such as ‘Add CP’ and/or ‘IFFT’ will be added in the transmitter structure, and slight adjustments may apply to the notations of certain signals, for example, $\mathbf{x}$ denotes the mapped symbol sequence in Fig. 1.1(a) but in Fig. 4.1 of Chapter 4, it denotes the symbol sequence after inverse fast Fourier transform (IFFT) operation.

### 1.1.2 Channel

An ISI channel with $L$ paths is considered in this thesis and the received symbol sequence $\mathbf{y} = \left[y_0, ..., y_V\right]^T$ is modelled as the linear convolution of the data and the channel, i.e.

$$y_i = \sum_{l=0}^{L-1} h_{i,l} x_{i-l} + n_i, \quad i = 0, ..., V, \quad V = N + L - 1 \tag{1.1}$$

where $h_{i,l}$ is the $l$th tap of the ISI channel at time index $i$, $l = 0, ..., L-1$, and $n_i$ is the complex additive white Gaussian noise (AWGN) with zero mean and variance $2\sigma^2$.

Each tap of the channel $h_{i,l}$ is modelled with Rayleigh distribution and the energy of the channel is normalized to 1, $\mathrm{E}\left[\frac{1}{L}\sum_{l=0}^{L-1} |h_{i,l}|^2\right] = 1$. The coherence time of the channel is considered to be much greater than the symbol time duration, and the Quasi-static assumption is adopted [24], i.e. the channel taps remain the same within a block but may vary from block to block.

### 1.1.3 Receiver

This thesis assumes perfect coherent detection at the receiver, i.e. practical issues such as imperfect synchronization, non-linearity in amplifiers are not considered. The convolutional code trellis, interleaving pattern, symbol constellation and the noise variance $2\sigma^2$ are assumed to be perfectly known at the receiver. The coded bits $\mathbf{d}$, interleaved coded bits $\mathbf{c}$ and the data symbols $\mathbf{x}$ are assumed to be independent, thanks to the use of the interleaver/de-interleaver [4]. The probability of each data symbol $x_n$, $n = 0, ..., N-1$, is assumed to be with Gaussian distribution, i.e. $P(x_n) \propto \exp\left[-(x_n - m_n)^2/v_n\right]$, parameterized by its mean $m_n$ and $v_n$. As illustrated in Fig. 1.1(d), each module of the turbo receiver functions as follows.

### Equalizer

The purpose of the equalizer is to calculate the extrinsic means $\mathbf{m}^e = \left[m_0^e, ..., m_{N-1}^e\right]^T$ and variances $\mathbf{v}^e = \left[v_0^e, ..., v_{N-1}^e\right]^T$ of the data symbols $\mathbf{x}$, based on the observation $\mathbf{y}$, the channel $\mathbf{h} = \left\{h_l, l = 0, ..., L-1\right\}$ and the *a priori* means $\mathbf{m}^a = \left[m_0^a, ..., m_{N-1}^a\right]^T$ and variances $\mathbf{v}^a = \left[v_0^a, ..., v_{N-1}^a\right]^T$ of the data symbols.

For the first iteration, there is no information coming from the decoder and the *a priori* information of the equalizer is initialized by $\mathbf{m}^a = \mathbf{0}_{N\times 1}$ and $\mathbf{v}^a = \mathbf{I}_{N\times 1}$. The equalizer first calculates the *a posteriori* mean vector $\mathbf{m}^p$ and covariance matrix $\mathbf{V}^p$ of the data symbol $\mathbf{x}$, based on MMSE principle as in [9, 10]. Then the extrinsic mean $m_n^e$ and variance $v_n^e$ of each symbol $x_n$, $n = 0, ..., N-1$, are calculated as follows [25].

$$v_n^e = \left(1/v_n^p - 1/v_n^a\right)^{-1} \tag{1.2}$$

$$m_n^e = v_n^e\left(m_n^p/v_n^p - m_n^a/v_n^a\right) \tag{1.3}$$

where $m_n^p$ is the $n$-th entry of the vector $\mathbf{m}^p$ and $v_n^p$ is the $n$-th diagonal entry of the matrix $\mathbf{V}^p$.

Note that the calculation of $\mathbf{m}^p$ and $\mathbf{V}^p$ is based on block-by-block processing while the calculation of $\mathbf{m}^e$ and $\mathbf{v}^e$ is based on symbol-by-symbol processing. Also, the algorithm of calculating $\mathbf{m}^p$ and $\mathbf{V}^p$ differs as the system models change, which will be demonstrated later in this thesis, by showing the different equalization algorithms in the SC-FDE system and in the OFDM system, for the conventional CP-discarding equalizers and for the proposed CP-exploiting equalizers.

### Demapper

The purpose of the demapper is to convert the extrinsic means $\mathbf{m}^e$ and variances $\mathbf{v}^e$ of the data symbols $\mathbf{x}$ into the LLRs $L_{eq}^e(\mathbf{c})$ of the interleaved coded bits $\mathbf{c}$.

The LLRs of the bits $c_{n,q}$, $q = 0, ..., Q-1$, in subsequence $\mathbf{c}_n$ (which is mapped to the symbol $x_n$), $n = 0, ..., N-1$, are calculated as

$$\begin{aligned} L_{eq}^{e}(c_{n,q}) &= \ln \frac{P(c_{n,q}=0)}{P(c_{n,q}=1)}, \\ &= \ln \frac{\sum_{\forall \mathcal{A}_i \in \boldsymbol{\mathcal{A}}:\, a_{i,q}=0} \left[ P(x_n = \mathcal{A}_i) \prod_{\forall q':\, q' \neq q} P(c_{n,q'} = a_{i,q'}) \right]}{\sum_{\forall \mathcal{A}_i \in \boldsymbol{\mathcal{A}}:\, a_{i,q}=1} \left[ P(x_n = \mathcal{A}_i) \prod_{\forall q':\, q' \neq q} P(c_{n,q'} = a_{i,q'}) \right]} \end{aligned} \tag{1.4}$$

where $\forall \mathcal{A}_i \in \boldsymbol{\mathcal{A}} : a_{i,q} = 0$ denotes all the constellation points $\mathcal{A}_i \in \boldsymbol{\mathcal{A}}$ that contain a '0' at the $q$th bit position, and $\forall \mathcal{A}_i \in \boldsymbol{\mathcal{A}} : a_{i,q} = 1$ denotes all the constellation points $\mathcal{A}_i \in \boldsymbol{\mathcal{A}}$ that contain a '1' at the $q$th bit position. The probability of the data symbol $x_n$ being the constellation point $\mathcal{A}_i$ is given by

$$P(x_n = \mathcal{A}_i) \propto \exp\left[ -\frac{(\mathcal{A}_i - m_n^e)^2}{v_n^e} \right], \quad i = 0, ..., M-1. \tag{1.5}$$

The probability of $c_{n,q}$ is calculated using the interleaved extrinsic LLRs $L_{dec}^{e}(\mathbf{c})$ from the decoder from previous iteration (which is the *a priori* LLRs of $\mathbf{c}$ for the demapper in this iteration) as

$$P(c_{n,q} = 0) = \frac{\exp(L_{dec}^{e}(c_{n,q}))}{1 + \exp(L_{dec}^{e}(c_{n,q}))}, \quad P(c_{n,q} = 1) = \frac{1}{1 + \exp(L_{dec}^{e}(c_{n,q}))}. \tag{1.6}$$

For the first iteration, $P(c_{n,q} = 0) = P(c_{n,q} = 1) = 1/2$.

Therefore, the extrinsic LLRs of the bit sequence $\mathbf{c}$ from the equalizer are formed as $L_{eq}^{e}(\mathbf{c}) = \left[ L_{eq}^{e}(c_{0,0}), ..., L_{eq}^{e}(c_{0,Q-1}), ..., L_{eq}^{e}(c_{N-1,0}), ..., L_{eq}^{e}(c_{N-1,Q-1}) \right]^T$.

### De-interleaver

The de-interleaver inverses the interleaving operation on $L_{eq}^{e}(\mathbf{c})$, i.e. $L_{dec}^{a}(\mathbf{d}) = \prod^{-1}(L_{eq}^{e}(\mathbf{c}))$, where $\prod^{-1}$ denotes the de-interleaving operation and $L_{dec}^{a}(\mathbf{d})$ is the *a priori* information of the decoder.

### Decoder

The purpose of the decoder is to calculate the extrinsic LLRs $L_{dec}^{e}(\mathbf{d})$ of the coded bits $\mathbf{d}$ based on the code trellis and the *a priori* LLRs $L_{dec}^{a}(\mathbf{d})$. This thesis employs the BCJR algorithm [17] to perform MAP decoding. Both the *a posteriori* LLRs of the coded bits $\mathbf{d}$ and the *a posteriori* LLRs of the information bits $\mathbf{b}$ are calculated. By subtracting the *a priori* LLRs $L^a_{dec}(\mathbf{d})$ from the *a posteriori* LLRs of $\mathbf{d}$, the extrinsic LLRs $L^e_{dec}(\mathbf{d})$ are obtained and output to the interleaver. When certain stopping criteria of the iteration is met, the decoder will make hard decision based on the *a posteriori* LLRs of the information bits $\mathbf{b}$.

### Interleaver

The interleaver performs $L^e_{dec}(\mathbf{c}) = \prod(L^e_{dec}(\mathbf{d}))$ and feeds $L^e_{dec}(\mathbf{c})$ into the mapper.

### Mapper

The purpose of the mapper is to convert the extrinsic LLRs $L^e_{dec}(\mathbf{c})$ of the coded bits $\mathbf{c}$ into the means $\mathbf{m}^a$ and variances $\mathbf{v}^a$ of the data symbols $\mathbf{x}$, which will serve as the *a priori* information of the equalizer. For each symbol $x_n$, $n = 0, ..., N-1$, the calculation of the mean $m_n$ and variance $v_n$ proceeds as follows.

$$m^a_n = \mathrm{E}(x_n) = \sum_{i=0}^{M-1} \mathcal{A}_i \, P(x_n = \mathcal{A}_i), \tag{1.7}$$

$$v^a_n = \mathrm{Cov}(x_n, x^*_n) = \sum_{i=0}^{M-1} |\mathcal{A}_i|^2 \, P(x_n = \mathcal{A}_i) - |m_n|^2. \tag{1.8}$$

The probability of $x_n$ being $\mathcal{A}_i$ is given by

$$P(x_n = \mathcal{A}_i) = \prod_{q=0}^{Q-1} P(c_{n,q} = a_{i,q}) \tag{1.9}$$

where $a_{i,q}$ is the $q$-th bit of the vector $\mathbf{a}_i$ (which corresponds to the constellation point $\mathcal{A}_i$) and $a_{i,q} \in \{0, 1\}$. The probability of $c_{n,q}$ is calculated as in (1.6) but using the interleaved extrinsic LLRs $L^e_{dec}(\mathbf{c})$ from the decoder from this iteration.

The means $\mathbf{m}^a$ and variances $\mathbf{v}^a$ of the data symbols $\mathbf{x}$ are passed on to the equalizer, and the iteration continues until certain stopping criteria is met and the decoder makes hard decisions $\hat{\mathbf{b}}$ on the transmitted bits. The stopping criteria can be *a*) when a predefined number of iterations are exhausted, *b*) when the BER performance satisfies the QoS requirement, or *c*) simply when the system has converged and no further improvement is seen in the performance if the iteration goes on.

## 1.2 Factor Graph Approach

Graphical models [26] in general and factor graphs [27–29] in particular provide a visual way to describe structured systems and to develop algorithms for those systems. In [30], Tanner first introduced graphs to describe families of codes, to which the low-density parity-check (LDPC) codes [31] belong. Wiberg *et al.* then generalized the "Tanner graphs" in [32, 33] and suggested their applications beyond coding. Taking one step further, factor graphs [27–29] have applied these graphical models to general functions.

In signal processing, digital communication, artificial intelligence and many other research areas, there are a wide variety of algorithms which deal with complicated global functions and depend on a large number of variables. More often than not, these functions can factor into products of simpler local functions, each of which only depends on a subset of the variables. Such factorization can be visualized in a bipartite graph, called factor graph [27–29].

Depending on the notation system of the graph, factor graph can have different styles [27–29]. This thesis adopts the Forney-style factor graph (FFG) [28, 29]. Different from the original factor graph [27] which have both variable nodes and factor nodes, the FFG only consists of factor nodes, and the variables are represented by edges. The definition of an FFG is that, *a*) there is a unique node for every factor (i.e. local function), *b*) there is a unique edge or half-edge (when the variable is only connected to one node) for every variable, and *c*) the node representing factor $f$ is connected with the edge (or half-edge) representing variable $x$, if and only if $f$ is a function of $x$.

The sum-product algorithm [27] is a message-passing algorithm that operates in factor graphs and computes various marginal functions associated with the global function. The update rule of the sum-product algorithm is that, the message out of the factor $f$ along the edge $x$ is the product of $f$ with all the incoming messages along all the edges except $x$, summarized over all the variables associated with $f$ except $x$. It is important to note that any message along the edge $x$ does not depend on any other variable and is a function of $x$ only.

For linear Gaussian models, the sum-product algorithm preserves Gaussianity, i.e. if the incoming messages of a node are members of the exponential family [3], so are the outgoing messages of that node. Let $\mathbf{x}$ be a vector variable with multi-variable Gaussian distribution. Then the message of $\mathbf{x}$ can be parameterized by either the mean vector $\mathbf{m}$ and the covariance matrix $\mathbf{V}$ or the transformed mean $\mathbf{Wm}$ and the weight matrix $\mathbf{W}$. For certain messages, $\mathbf{V}$ (or $\mathbf{W}$) can be singular and its counterpart dual, $\mathbf{Wm}$ and $\mathbf{W}$ (or $\mathbf{m}$ and $\mathbf{V}$), needs to be used (which can happen quite frequently but is seldom an issue [29]). Computation rules for Gaussian message passing through equality constraint nodes, adder nodes and multiplier nodes with known (constant) coefficients have been tabulated in [29]. For multiplier nodes with unknown coefficients, [34] has presented the EM algorithm as a message passing algorithm as well.

This thesis are based on the notations and computation rules in [29, 34]. A variable is represented by a directed edge, i.e. an edge with an arrow, in an FFG. The arrow on the edge shows the natural direction of information flow in the system and helps with the notation of the multiple messages on the edge. For a vector variable $\mathbf{x}$, $\overrightarrow{\mathbf{m}}_{\mathbf{x}}$ and $\overrightarrow{\mathbf{V}}_{\mathbf{x}}$ denote the mean vector and the covariance matrix of the forward message of $\mathbf{x}$ (the message that flows in the direction of the edge, whichever that direction is), and $\overleftarrow{\mathbf{m}}_{\mathbf{x}}$ and $\overleftarrow{\mathbf{V}}_{\mathbf{x}}$ denote the mean vector and the covariance matrix of the backward message of $\mathbf{x}$ (which flows in the opposite direction). The product of the forward and backward message of $\mathbf{x}$ is the marginal message of $\mathbf{x}$, denoted by $\mathbf{m}_{\mathbf{x}}$, $\mathbf{V}_{\mathbf{x}}$. In occasions where a variable is shared by many factors, clones of the variable are used. For example, $\mathbf{x}'$ and $\mathbf{x}''$ are two clones of the variable $\mathbf{x}$ and they are essentially equivalent, i.e. $\mathbf{x} = \mathbf{x}' = \mathbf{x}''$.

If the factor graph is in the form of a tree, i.e. it does not have cycles, the propagation of the messages can start from the leaf nodes or edges and work along the tree until finishing at the root nodes or edges (the definition of leaf and root depends on the marginal function of interest). In a factor graph with cycles, however, things are much more complex. The transmission of a message on any edge of a cycle from a node $f$ will trigger a chain of pending messages to be transmitted around the cycle, which then will return to $f$ and trigger $f$ to send another message on the same edge. Such propagation goes on and on without natural termination, and consequently the sum-product algorithm is applied around the cycles in an iterative fashion [27].

In a cycle-free factor graph, the result of the sum-product algorithm is the exact marginal function. When the factor graph has one or more cycles, the sum-product algorithm only leads to an approximation of the true marginal function. This approximate function, however, can be quite good. Applications such as the decoding of LDPC codes [31] and the turbo equalization [9,10,35] can be viewed as examples of the sum-product algorithm operating iteratively on graphs with cycles, and they have been shown to be able to provide excellent performance. This thesis considers the turbo equalization system and focuses on the equalizer module at the receiver. The whole system can be viewed as a graph with cycles but the equalizer considered in this thesis corresponds to a smaller graph that is cycle-free. The equalizer processes signals on a block basis (group basis in Chapter 4) and there is no dependence between the blocks (groups) except when interference cancellation is involved.

## 1.3 Cyclic Prefix in SC-FDE and OFDM

Single carrier transmission with frequency domain equalization (SC-FDE) [36–38] and orthogonal frequency division multiplexing (OFDM) [39] are two of the most important techniques in digital communications. Both schemes are based on block transmissions and can be implemented using fast Fourier transform/inverse fast Fourier transform (FFT/IFFT) operations, and both schemes use guard intervals to mitigate the inter-block interference (IBI). The main difference of the two schemes lies in the placement of the IFFT module in the transceiver structure. In OFDM, the IFFT module is at the transmitter to multiplex the data into parallel sub-carriers while in SC-FDE, the IFFT module is at the receiver to convert FDE signals back into time domain.

The primary advantage of SC-FDE over OFDM is that SC-FDE has lower peak-to-average power ratio (PAPR) due to its inherent single carrier structure. It is also because of this advantage that single carrier frequency division multiple access (SC-FDMA), the multi-user version of SC-FDE, becomes a favorite transmission scheme for uplink wireless communications such as 3GPP long term evolution (LTE) [40–42], where power efficiency is of paramount importance for mobile terminals. On the other hand, OFDM is more preferable in many applications than SC-FDE because of its ability to cope with severe channel conditions such as narrow-band interference, and also its superior performance when coupled with strong error correction codes (ECC) and higher signal constellations. A wide variety of applications and wireless broad-band standards have adopted the OFDM scheme, for example, the digital audio broadcasting (DAB) [43], digital video broadcasting (DVB) [44], IEEE 802.11a

[45], MMAC [46], HIPERLAN/2 [47] and the downlink transmission of LTE [40–42].

As a type of guard interval widely used in SC-FDE and OFDM, the cyclic prefix (CP) refers to the prefixing of each block with a repetition of the last several symbols of the block. The length of the CP must be at least equal to the maximum spread length of the multipath channel for it to be effective, and at the receiver side, the observation corresponding to the CP part is normally discarded. The use of CP allows the linear convolution of a multipath channel and the data to be modelled as circular convolution, which in turn can be transformed to point-wise multiplication in frequency domain (from time domain). The transformation to and back from frequency domain can be easily achieved via the FFT and IFFT operations, which only require a complexity of $\mathcal{O}(NlogN)$ per block ($N$ is the length of the block). Compared to traditional time domain processing, SC-FDE and OFDM have a much lower complexity at the receiver, thanks to the use of the CP.

On the other hand, the extra transmission time and energy induced by the CP renders a loss in both spectrum efficiency and power efficiency, especially when the channel has a long delay spread. However, since the CP inherently is a repetition of the last several symbols of each block, it contains useful information about the transmitted data and many studies have been carried out to exploit the redundancy to improve equalization [48–60], channel estimation [61–65] and synchronization [66–68], *etc.* This thesis will focus on utilizing the CP for equalization and channel estimation.

Some studies are carried out in the context of non-iterative systems. For example, [48] investigates the exploitation of CP in the SC-FDE system, where two independent estimates are obtained from the two observations of the CP and are combined together based on the maximum ratio combining (MRC) rule. For OFDM systems, [49] and [50] assume that the length of the CP is designed to be longer than the maximum delay spread of the channel, and a portion of the CP, which is free of interference from the preceding OFDM symbol, is used to improve the data detection. As an extension, [51–56] exploit the whole CP after performing interference cancellation. Also, various methods such as the ML search, genetic algorithm and Newton's method are compared in [57, 58]. Unfortunately, the performance gain of these works is quite limited, for example, [52] shows 0.58dB gain over the conventional OFDM system when the CP ratio is 1/4. Moreover, a very high complexity is exhibited, for instance, the minimum mean square error (MMSE) method in [55] features cubic complexity. In [59] and [60], a more sophisticated receiver is designed and a greater performance improvement is achieved, but again it comes with a dramatic increase in complexity. In the light of turbo equalization [7,9,10], the methods in [48–60] can all be extended to iterative receivers with proper adjustment. Also, the CP can be utilized for joint channel estimation and data detection as in [62–65] when the channel is unknown.

## 1.4 Thesis Contribution

This thesis considers the application of turbo equalization in SC-FDE and OFDM systems and based on the minimum mean square error (MMSE) principle, various equalization algorithms are proposed in order to exploit the prefix redundancy for data detection and/or channel estimation. To facilitate the description of system models and to develop low-complexity algorithms, Forney-style factor graphs (FFG) and the Gaussian message passing (GMP) technique in factor graphs are employed.

The contributions of this thesis are summarized as follows.

1. We have proposed a low-complexity equalization algorithm for the iterative SC-FDE system [69]. Compared to the conventional CP-discarding scheme, a gain of 0.7dB is achieved under both the additive white Gaussian noise

(AWGN) channel and the ISI channel when the CP ratio is 1/4 and the channel is perfectly known at the receiver, with complexity of $\mathcal{O}(NlogN)$ per data block per iteration, where $N$ is the length of the data block.

2. We have proposed a low-complexity equalization algorithm for the turbo OFDM system [70, 71]. Compared to the conventional CP-discarding scheme, a gain of 0.97dB under the AWGN channel and 2dB under the ISI channels are achieved when the CP ratio is 1/4 and the channel is known, with complexity of $\mathcal{O}(NlogN)$ per data block per iteration. We have shown that when CP is exploited for equalization in OFDM systems, the iterations between decoder and equalizer are necessary for taking full advantage of the CP, since the use of CP in equalization makes different sub-carriers dependent. We have provided an SNR analysis, which to the best of our knowledge has not appeared in the literature before, to verify the performance improvement of the proposed equalizer over the conventional equalizer. We have also investigated the sensitivity of the proposed equalizer to the quality of the available channel state information (CSI), and in the presence of channel estimation errors, the proposed equalizer shows robustness in terms of BER performance.

3. We have proposed a joint channel estimation, data detection and decoding approach for the turbo OFDM system [72, 73]. The proposed approach employs the expectation maximization (EM) algorithm via message passing and can use CP for both data detection and channel estimation. After convergence, a minimum square error of $10^{-4}$ is achieved for the channel estimation performance and the data detection performance is also close to that when the channel is known. We have accommodated both the means and variances of the channel estimate in the message passing EM algorithm, contrary to only using the means in previous literature. We have discussed the different ways

of using CP, i.e. for equalization/channel estimation only or both, and shown that using CP for equalization is more beneficial considering the performance gain that can be obtained and the complexity cost that comes with it. We have discussed the different scheduling schemes in the factor graph and proposed a novel scheme with complexity of $\mathcal{O}(NlogN)$ per block per iteration whose BER performance is close to that of the known channel case.

## 1.5 Thesis Outline

The rest of this thesis is organized as follows.

Chapter 2 considers the MMSE-based turbo SC-FDE system when the channel is known. First, the conventional turbo SC-FDE system model is interpreted as an FFG and an equalization algorithm is derived. Then the normally discarded CP part is presented similarly in an FFG and an algorithm that integrates the two FFGs is developed accordingly. Complexity evaluation and simulation results are provided.

Chapter 3 considers the coded OFDM systems where turbo equalization is employed and the channel is known. Both the CP and the non-CP parts of the received signal are presented in an FFG and an equalization algorithm is proposed to exploit the CP redundancy. An SNR analysis is provided to verify the improvement of the proposed algorithm over the conventional algorithm. Approximations are discussed and a reduced-complexity version of the proposed algorithm is proposed. Impacts of the approximations are investigated in terms of both complexity reduction and BER performance. The sensitivity of the proposed reduced-complexity algorithm to channel estimation error is also examined.

Chapter 4 considers joint channel estimation, data detection and decoding in coded OFDM systems where turbo equalization is employed. First, models for the CP observation, non-CP observation and the time correlation of the time-varying channel are presented in three FFGs, and message passing process is described for each graph. Modification in the message passing EM algorithm is made to accommodate both the means and variances of the channel estimates in the system models. When three graphs are combined, a larger EM algorithm is formed, and different uses of the CP and scheduling schemes for message passing between graphs are discussed. A novel scheduling scheme is proposed as an effective trade-off between the system performance and complexity. Complexity evaluation and simulation results are provided.

Chapter 5 draws conclusions of this thesis and discusses future work.

# Chapter 2

# Exploiting Cyclic Prefix in Turbo FDE Systems

SC-FDE is one of the most important techniques to combat the severe frequency selectivity of the channels in high-rate digital communications [37] and turbo equalization is proved to be able to provide extraordinary performance [4]. When turbo equalization is employed in SC-FDE systems, namely, turbo FDE [74], the complexity of the receiver can be reduced to $\mathcal{O}(NlogN)$ per data block per iteration, where $N$ is the length of data block. As a result, turbo FDE can achieve superior performance with limited complexity, making it a promising technique for communications under frequency selective channels. In a turbo FDE system, the CP is typically discarded at the receiver before any further processing. In this chapter, on the contrary, we take advantage of the redundancy imposed by the CP and make use of all the observations for equalization purpose. To the best of our knowledge, little research has been done in this regard for turbo FDE systems.

We consider the MMSE-based turbo FDE systems. First, we interpret the conventional turbo FDE system as an FFG, and an equalization algorithm is derived based on the GMP technique [29]. Then the normally discarded CP part is presented similarly using an FFG. An algorithm that integrates the two FFGs is developed accordingly. As a result, two extrinsic messages about the data are obtained rather than one. A symbol-wise combination of the two messages produces a better estimation of the data symbols. With proper approximations, the proposed algorithm is evaluated to achieve the same order of complexity as that of the conventional one, i.e. $\mathcal{O}(NlogN)$ per block per iteration. Simulation results verify that at 1/4 CP ratio, a gain of around 0.7dB is obtained for both 16QAM and 64QAM systems compared with the conventional algorithm.

## 2.1 System Model

Consider a turbo FDE system as shown in Fig. 2.1. The information bits are encoded, interleaved and then applied to a mapper/modulator. Let $\mathbf{x}$ be the QAM modulated symbol sequence, $\mathbf{x} = [x_0, x_1, ..., x_Z]^T$, where $Z$ is the length of the sequence. By partitioning the frame $\mathbf{x}$ into $K$ length-$N$ blocks (assume $Z = KN$), we have $\mathbf{x} = \left[\mathbf{x}_0^T, ..., \mathbf{x}_k^T, ..., \mathbf{x}_{K-1}^T\right]^T$, where $\mathbf{x}_k = [x_{k,0}, x_{k,1}, ..., x_{k,N-1}]^T$, $k = 0, ..., K-1$. The cyclic prefix (CP) of each block, $\tilde{\mathbf{x}}_k$, is formed by

$$\tilde{\mathbf{x}}_k = \mathbf{A} \cdot \mathbf{x}_k = [x_{k,N-P}, x_{k,N-P+1}, ..., x_{k,N-1}]^T \tag{2.1}$$

where $\mathbf{A} = \left[\begin{array}{cc} \mathbf{0}_{P\times(N-P)} & \mathbf{I}_P \end{array}\right]_{P\times N}$ and it extends the block $\mathbf{x}_k$ to $\mathbf{s}_k = \left[\tilde{\mathbf{x}}_k^T, \mathbf{x}_k^T\right]^T$. For the CP to be effective, we require $P \geqslant L-1$, where $L$ is the length of the channel.

Given the transmitted sequence $\mathbf{s} = \left[\mathbf{s}_0^T, \mathbf{s}_1^T, ..., \mathbf{s}_{K-1}^T\right]^T = \left[s_0, s_1, ..., s_{K(N+P)}\right]^T$, the received symbols are

$$r_i = \sum_{l=0}^{L-1} h_{i,l} s_{i-l} + n_i, \quad i = 0, ..., V \tag{2.2}$$

where $V = Z + KP + L - 1$, $h_{i,l}$ is the $l$th tap of the channel at time index $i$, and $n_i$

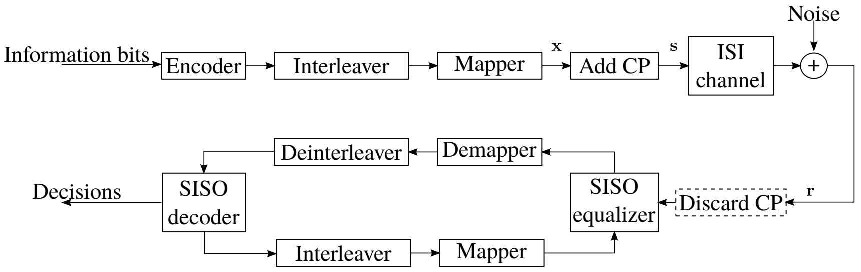


**Figure 2.1:** Block diagram of a turbo FDE system.

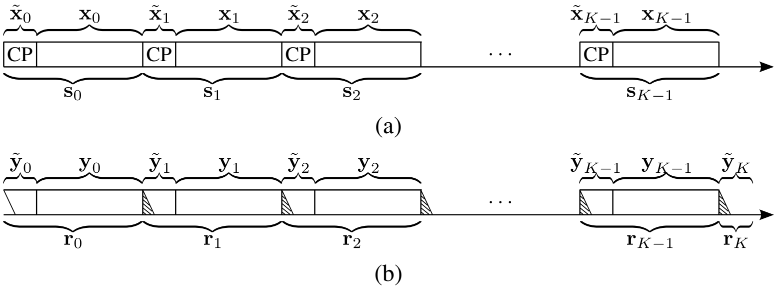


**Figure 2.2:** Data structure of a turbo FDE system. (a) Transmitted signal (b) Received signal

is the complex additive white Gaussian noise (AWGN) with zero mean and variance $2\sigma^2$.

In the following, we assume that the channel is Quasi-static, i.e. the $L$ channel taps vary from block to block but remain the same within a block. As illustrated in Fig. 2.2, corresponding to $\mathbf{s}_k = \left[\tilde{\mathbf{x}}_k^T, \mathbf{x}_k^T\right]^T$, each received block can be split into two parts, $\mathbf{r}_k = \left[\tilde{\mathbf{y}}_k^T, \mathbf{y}_k^T\right]^T$. The term $\mathbf{y}_k$ is given by

$$\mathbf{y}_k = \mathbf{x}_k \circledast \mathbf{h}_k + \mathbf{n}_k, \quad k = 0, ..., K-1 \tag{2.3}$$

where "$\circledast$" denotes the circular convolution, $\mathbf{h}_k$ represents the channel that $\mathbf{x}_k$ undergoes, and $\mathbf{n}_k$ is the noise vector with mean vector $\mathbf{0}$ and covariance matrix $2\sigma^2\mathbf{I}$. The other term $\tilde{\mathbf{y}}_k$ is

$$\tilde{\mathbf{y}}_k = Tail\left(\tilde{\mathbf{x}}_{k-1} * \mathbf{h}_{k-1}\right) + Body\left(\tilde{\mathbf{x}}_k * \mathbf{h}_k\right) + \tilde{\mathbf{n}}_k, \tag{2.4}$$

for $k = 1, ..., K-1$. Here, "$*$" denotes the linear convolution and $\tilde{\mathbf{n}}_k$ is the noise term of length $P$. $Tail(\cdot)$ and $Body(\cdot)$ are two truncation functions, which return the last $L-1$ symbols and the first $P$ symbols of the vector in $(\cdot)$, respectively. $Tail(\cdot)$ also pads $P-L+1$ zeros at the end in order to be of the same length as $Body(\cdot)$ and $\tilde{\mathbf{n}}_k$. Specially, for $k=0$, we have $\tilde{\mathbf{y}}_0 = Body\left(\tilde{\mathbf{x}}_0 * \mathbf{h}_0\right) + \tilde{\mathbf{n}}_0$, and for $k=K$, $\tilde{\mathbf{y}}_K = Tail\left(\tilde{\mathbf{x}}_{K-1} * \mathbf{h}_{K-1}\right) + \tilde{\mathbf{n}}_K$, where $\tilde{\mathbf{y}}_K$ and $\tilde{\mathbf{n}}_K$ are also of length $P$.

Alternatively, we can write (2.4) in matrix form as

$$\tilde{\mathbf{y}}_k = \mathbf{T} \cdot \left(\tilde{\mathbf{x}}_{k-1} * \mathbf{h}_{k-1}\right) + \mathbf{B} \cdot \left(\tilde{\mathbf{x}}_k * \mathbf{h}_k\right) + \tilde{\mathbf{n}}_k \tag{2.5}$$

where $\mathbf{T}$ and $\mathbf{B}$ are two matrices, both of size $P \times G$, $G = P + L - 1$

$$\mathbf{T} = \left[ \begin{array}{c|c} \mathbf{0}_{P\times P} & \begin{array}{c} \mathbf{I}_{L-1} \\ \\ \mathbf{0}_{(P-L+1)\times(L-1)} \end{array} \end{array} \right]_{P\times G}, \quad \mathbf{B} = \left[ \begin{array}{cc} \mathbf{I}_P & \mathbf{0}_{P\times(L-1)} \end{array} \right]_{P\times G}. \tag{2.6}$$

Linear convolution after proper zero-padding and circular convolution in time domain can be transformed into multiplication in frequency domain by discrete Fourier transform (DFT) operations. Therefore, (2.3) and (2.5) can be rewritten as

$$\mathbf{y}_k = \mathbf{F}_N^H \mathbf{H}_k \mathbf{F}_N \mathbf{x}_k + \mathbf{n}_k, \tag{2.7}$$

$$\tilde{\mathbf{y}}_k = \mathbf{T} \cdot \left(\mathbf{F}_G^H \tilde{\mathbf{H}}_{k-1} \mathbf{F}_G \mathbf{Q} \mathbf{A} \mathbf{x}_{k-1}\right) + \mathbf{B} \cdot \left(\mathbf{F}_G^H \tilde{\mathbf{H}}_k \mathbf{F}_G \mathbf{Q} \mathbf{A} \mathbf{x}_k\right) + \tilde{\mathbf{n}}_k \tag{2.8}$$

where $\mathbf{H}_k$ and $\tilde{\mathbf{H}}_k$ are the DFTs of the channel $\mathbf{h}_k$ with sizes $N \times N$ and $G \times G$ respectively,

$$\mathbf{H}_k = Diag\left(\sqrt{N}\mathbf{F}_N \hat{\mathbf{h}}_k\right), \qquad \text{with} \quad \hat{\mathbf{h}}_k = \begin{bmatrix} \mathbf{h}_k \\ \\ \mathbf{0} \end{bmatrix}_{N\times 1}, \tag{2.9}$$

$$\tilde{\mathbf{H}}_k = Diag\left(\sqrt{G}\mathbf{F}_G \check{\mathbf{h}}_k\right), \qquad \text{with} \quad \check{\mathbf{h}}_k = \begin{bmatrix} \mathbf{h}_k \\ \\ \mathbf{0} \end{bmatrix}_{G\times 1}, \tag{2.10}$$

and the use of $\mathbf{Q}$ is to add $L-1$ zeros at the end of $\tilde{\mathbf{x}}_k$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{I}_P \\ \\ \mathbf{0}_{(L-1)\times P} \end{bmatrix}_{G\times P}. \tag{2.11}$$

In fact, (2.7) represents the model of the conventional FDE systems, where CP is discarded and only $\mathbf{y}_k$ is utilized for further processing. On the other hand, (2.8) offers a description of the CP part, which is normally ignored. As we can see from (2.7) and (2.8), each data block $\mathbf{x}_k$ actually experiences two channels and two observations are obtained at the receiver, $\mathbf{y}_k$ and $\tilde{\mathbf{y}}_k$. In the next section, we will present models (2.7) and (2.8) in factor graphs [29] and use the Gaussian message passing technique [29] to exploit all the observations for equalization purpose.

## 2.2 Graph Based Equalization Algorithms

### 2.2.1 Conventional Equalizer

Fig. 2.3 depicts the FFG of the $k$th block of the system model (2.7) and Fig. 2.4 presents the block structure of the whole system for model (2.7). As can be seen from Fig. 2.4, each block is independent of others. Based on the message passing rules in [29], the conventional equalization algorithm is derived and proceeds in the following steps S1-S4.

**S1. Initialization**

In each iteration, the soft decoder outputs the extrinsic log-likelihood ratios (LLRs) about the coded bits and the LLRs are then interleaved and converted into probabilities about the symbols. Parameterized by mean vector $\mathbf{m}_k^{apri}$ and covariance matrix $\mathbf{V}_k^{apri}$ for block $k$, $k = 0, ..., K-1$, these probabilities of the data symbols are fed into the equalizer as the *a priori* information [9,10,25].

It is reasonable to model the *a priori* covariance matrix $\mathbf{V}_k^{apri}$ as a scaled identity

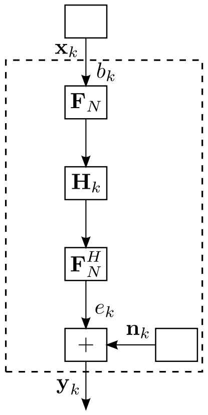


**Figure 2.3:** FFG of the $k$th block of the system model (2.7).

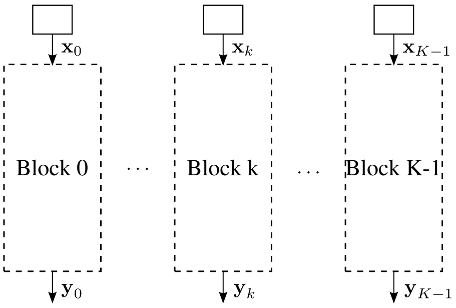


**Figure 2.4:** Block structure for the system described by (2.7).

matrix, since $a$) symbols within a block can be considered independent of each other due to the employment of the interleaver, rendering the matrix diagonal, $b$) it is a common practice [9, 10, 25] to approximate the variances of the symbols within a block by their average $\alpha_k$, $k = 0, ..., K-1$, i.e. $\alpha_k = \frac{1}{N}\sum_{i=0}^{N-1} v_{x_{k,i}}^{apri}$, where $v_{x_{k,i}}^{apri}$ is the *a priori* variance of the symbol $x_{k,i}$. Thus, the forward message at $b_k$ in Fig. 2.3 can be given by

$$\overrightarrow{\mathbf{m}}_{b_k} = \mathbf{m}_k^{apri}, \quad \overrightarrow{\mathbf{V}}_{b_k} = \mathbf{V}_k^{apri} = \alpha_k \mathbf{I}. \tag{2.12}$$

Note that in Fig. 2.3, the reference of points $b_k$, $e_k$, etc, are only for notational convenience (in particular for notational consistency with Fig. 2.5); for instance, the mean vector and covariance matrix at point $b_k$ in fact are the mean vector and covariance matrix of the variable $\mathbf{x}_k$.

### S2. Backward message at $b_k$

According to rules (III.6) and (III.5) in [29], we can obtain the backward message at $b_k$ based on the observation and the noise characteristics. With $\overleftarrow{\mathbf{m}}_{e_k} = \mathbf{y}_k$ and $\overleftarrow{\mathbf{V}}_{e_k} = \overleftarrow{\mathbf{V}}_{\mathbf{n}_k} = \beta_k \mathbf{I}$, where $\beta_k = 2\sigma^2$ is the noise power, we have

$$\overleftarrow{\mathbf{V}}_{b_k}^{-1}\overleftarrow{\mathbf{m}}_{b_k} = \beta_k^{-1}\mathbf{F}_N^H\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k, \tag{2.13}$$

$$\overleftarrow{\mathbf{V}}_{b_k}^{-1} = \beta_k^{-1}\mathbf{F}_N^H\mathbf{H}_k^H\mathbf{H}_k\mathbf{F}_N. \tag{2.14}$$

### S3. Combination

Based on rules (II.3) and (II.1) in [29] and using (2.12)-(2.14), the marginal message at point $b_k$ can be determined by

$$\mathbf{m}_{b_k} = \mathbf{F}_N^H \left(\alpha_k^{-1}\mathbf{I} + \beta_k^{-1}\mathbf{H}_k^H\mathbf{H}_k\right)^{-1} \left(\alpha_k^{-1}\mathbf{F}_N\mathbf{m}_k^{apri} + \beta_k^{-1}\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k\right), \tag{2.15}$$

$$\mathbf{V}_{b_k} = \mathbf{F}_N^H \left(\alpha_k^{-1}\mathbf{I} + \beta_k^{-1}\mathbf{H}_k^H\mathbf{H}_k\right)^{-1} \mathbf{F}_N. \tag{2.16}$$

### S4. Extrinsic Information

Following the relations (14) and (15) in [25], the extrinsic information of each symbol $x_{k,i}$ can be calculated as

$$m_{x_{k,i}}^{ext} = v_{x_{k,i}}^{ext}\left(m_{x_{k,i}}^{post}/v_{x_{k,i}}^{post} - m_{x_{k,i}}^{apri}/v_{x_{k,i}}^{apri}\right), \tag{2.17}$$

$$v_{x_{k,i}}^{ext} = \left(1/v_{x_{k,i}}^{post} - 1/v_{x_{k,i}}^{apri}\right)^{-1} \tag{2.18}$$

where $m_{x_{k,i}}^{post}$ and $m_{x_{k,i}}^{apri}$ are the $i$th entry of the mean vector $\mathbf{m}_{b_k}$ and $\mathbf{m}_k^{apri}$, respectively, and $v_{x_{k,i}}^{post}$ and $v_{x_{k,i}}^{apri}$ are the $i$th diagonal entry of the covariance matrix $\mathbf{V}_{b_k}$ and $\mathbf{V}_k^{apri}$, respectively.

Note that $\mathbf{V}_{b_k}$ in (2.16) is a circulant matrix and only its diagonal elements are of our interest. Thus, the *a posteriori* variances of the symbols within a block are

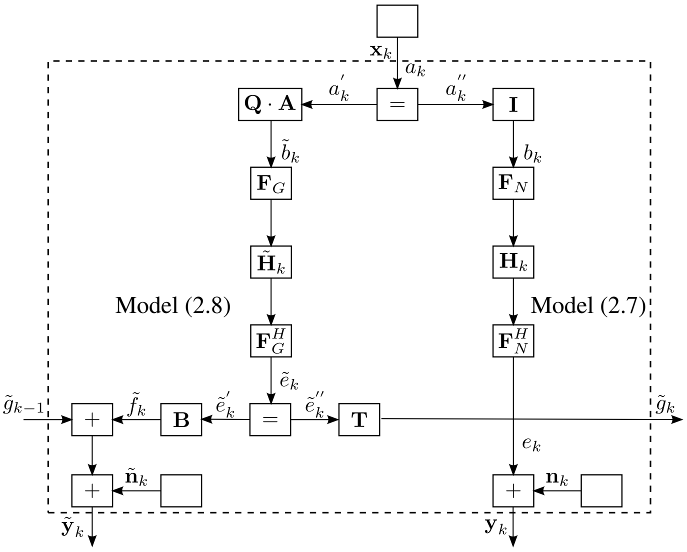


**Figure 2.5:** FFG of the $k$th block of the system model (2.7) and (2.8).

all identical by nature, i.e. for $i = 0, ..., N-1$,

$$v_{x_{k,i}}^{post} = \frac{1}{N}\sum_{j=0}^{N-1}\left[\frac{1}{\alpha_k} + \frac{1}{\beta_k}\left(\mathbf{H}_k\right)_{j,j}^{H}\left(\mathbf{H}_k\right)_{j,j}\right]^{-1}. \tag{2.19}$$

### 2.2.2 Proposed Equalizer

The FFG of the $k$th block of the system model (2.8) is shown in Fig. 2.5. Fig. 2.6 presents the block structure of the whole system for both model (2.7) and model (2.8). As we can see from Fig. 2.6, each block is connected with its two adjacent blocks, except for the first block and the last. For Block 0, we consider the incoming message from the left side as a degenerate deterministic Gaussian message, i.e. $\overrightarrow{\mathbf{m}}_{\tilde{g}_{-1}} = \mathbf{0}$ and $\overrightarrow{\mathbf{V}}_{\tilde{g}_{-1}} = \mathbf{0}$, and for Block $K-1$ the tail symbols are taken as the incoming message from the right side, i.e. $\overleftarrow{\mathbf{m}}_{\tilde{g}_{K-1}} = \tilde{\mathbf{y}}_K$ and $\overleftarrow{\mathbf{V}}_{\tilde{g}_{K-1}} = \mathbf{V}_{\tilde{\mathbf{n}}_k}$.

In the following, an equalization algorithm that exploits both model (2.7) and model (2.8) is proposed and it proceeds in five steps S1-S5. Note that S1-S4 only take

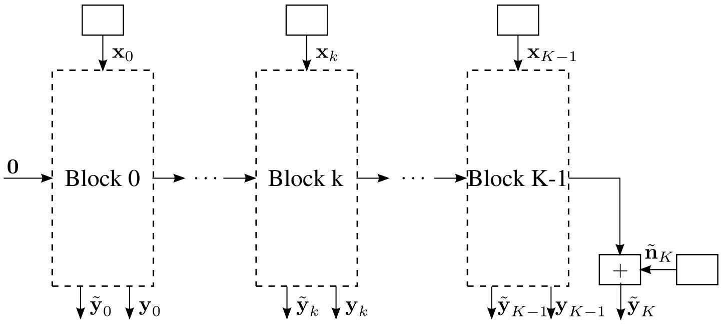


**Figure 2.6:** Block structure for the whole system described by (2.7) and (2.8).

the contribution of $\tilde{\mathbf{y}}_k$ from model (2.8) into consideration and only one extrinsic information for $\mathbf{x}_k$, $k = 0, ..., K-1$ is calculated; in S5, the extrinsic message obtained from model (2.7) in Section 2.2.1 is added in and combined with the extrinsic message obtained from (2.8) on a symbol level.

### S1. Initialization

In line with (2.12) in S1 in Section 2.2.1, the forward message at point $\tilde{b}_k$ is given by

$$\overrightarrow{\mathbf{m}}_{\tilde{b}_k} = (\mathbf{QA})\,\mathbf{m}_k^{apri} = (m_{x_{k,N-P}}^{apri}, ..., m_{x_{k,N-1}}^{apri}, 0, ..., 0)^T, \tag{2.20}$$

$$\overrightarrow{\mathbf{V}}_{\tilde{b}_k} = (\mathbf{QA})\,\mathbf{V}_k^{apri}\,(\mathbf{QA})^H = Diag(\alpha_k, ..., \alpha_k, 0, ..., 0). \tag{2.21}$$

### S2. Forward message at point $\tilde{e}_k$

According to rules (III.2) and (III.1) in [29], the forward message at $\tilde{e}_k$ is obtained as

$$\overrightarrow{\mathbf{m}}_{\tilde{e}_k} = \mathbf{F}_G^H \tilde{\mathbf{H}}_k \mathbf{F}_G \overrightarrow{\mathbf{m}}_{\tilde{b}_k}, \tag{2.22}$$

$$\overrightarrow{\mathbf{V}}_{\tilde{e}_k} = \mathbf{F}_G^H \tilde{\mathbf{H}}_k \mathbf{F}_G \overrightarrow{\mathbf{V}}_{\tilde{b}_k} \mathbf{F}_G^H \tilde{\mathbf{H}}_k^H \mathbf{F}_G. \tag{2.23}$$

### S3. Backward message at point $\tilde{e}_k$

Applying rules (III.2) and (III.1) in [29] again, the message passed from Block $k-1$, which is also the forward message at $\tilde{g}_{k-1}$, is

$$\overrightarrow{\mathbf{m}}_{\tilde{g}_{k-1}} = \mathbf{T}\overrightarrow{\mathbf{m}}_{\tilde{e}^{''}_{k-1}}, \tag{2.24}$$

$$\overrightarrow{\mathbf{V}}_{\tilde{g}_{k-1}} = \mathbf{T}\overrightarrow{\mathbf{V}}_{\tilde{e}^{''}_{k-1}}\mathbf{T}^H. \tag{2.25}$$

Then with rules (II.10) and (II.8) in [29] for the adder node, we have

$$\overleftarrow{\mathbf{m}}_{\tilde{f}_k} = \tilde{\mathbf{y}}_k - \overrightarrow{\mathbf{m}}_{\tilde{g}_{k-1}}, \tag{2.26}$$

$$\overleftarrow{\mathbf{V}}_{\tilde{f}_k} = \overrightarrow{\mathbf{V}}_{\tilde{g}_{k-1}} + \mathbf{V}_{\tilde{\mathbf{n}}_k} \tag{2.27}$$

where similar assumption is made as in S2 in Section 2.2.1, i.e. $\mathbf{V}_{\tilde{\mathbf{n}}_k} = \beta_k\mathbf{I}$ and $\beta_k = 2\sigma^2$. Next, rules (III.6) and (III.5) in [29] brings us to the backward message at point $\tilde{e}^{'}_k$ as

$$\overleftarrow{\mathbf{V}}^{-1}_{\tilde{e}^{'}_k}\overleftarrow{\mathbf{m}}_{\tilde{e}^{'}_k} = \mathbf{B}^H\overleftarrow{\mathbf{V}}^{-1}_{\tilde{f}_k}\overleftarrow{\mathbf{m}}_{\tilde{f}_k}, \tag{2.28}$$

$$\overleftarrow{\mathbf{V}}^{-1}_{\tilde{e}^{'}_k} = \mathbf{B}^H\overleftarrow{\mathbf{V}}^{-1}_{\tilde{f}_k}\mathbf{B}, \tag{2.29}$$

and by substituting (2.24)-(2.27) into (2.28) and (2.29), we get

$$\overleftarrow{\mathbf{V}}^{-1}_{\tilde{e}^{'}_k}\overleftarrow{\mathbf{m}}_{\tilde{e}^{'}_k} = \mathbf{B}^H\left(\mathbf{T}\overrightarrow{\mathbf{V}}_{\tilde{e}^{''}_{k-1}}\mathbf{T}^H + \beta_k\mathbf{I}\right)^{-1}\left(\tilde{\mathbf{y}}_k - \mathbf{T}\overrightarrow{\mathbf{m}}_{\tilde{e}^{''}_{k-1}}\right), \tag{2.30}$$

$$\overleftarrow{\mathbf{V}}^{-1}_{\tilde{e}^{'}_k} = \mathbf{B}^H\left(\mathbf{T}\overrightarrow{\mathbf{V}}_{\tilde{e}^{''}_{k+1}}\mathbf{T}^H + \beta_{k+1}\mathbf{I}\right)^{-1}\mathbf{B}. \tag{2.31}$$

A similar process as the derivation of (2.30) and (2.31) passes the messages from Block $k+1$ to Block $k$ and produces the backward message at $\tilde{e}^{''}_k$ as

$$\overleftarrow{\mathbf{V}}^{-1}_{\tilde{e}^{''}_k}\overleftarrow{\mathbf{m}}_{\tilde{e}^{''}_k} = \mathbf{T}^H\left(\mathbf{B}\overrightarrow{\mathbf{V}}_{\tilde{e}^{'}_{k+1}}\mathbf{B}^H + \beta_{k+1}\mathbf{I}\right)^{-1}\left(\tilde{\mathbf{y}}_{k+1} - \mathbf{B}\overrightarrow{\mathbf{m}}_{\tilde{e}^{'}_{k+1}}\right), \tag{2.32}$$

$$\overleftarrow{\mathbf{V}}^{-1}_{\tilde{e}^{''}_k} = \mathbf{T}^H\left(\mathbf{B}\overrightarrow{\mathbf{V}}_{\tilde{e}^{'}_{k+1}}\mathbf{B}^H + \beta_{k+1}\mathbf{I}\right)^{-1}\mathbf{T}. \tag{2.33}$$

Specially, when $k = 0$ and $k = K - 1$, we have

$$\overleftarrow{\mathbf{V}}_{\tilde{e}_0'}^{-1}\overleftarrow{\mathbf{m}}_{\tilde{e}_0'} = \beta_0^{-1}\mathbf{B}^H\tilde{\mathbf{y}}_0, \qquad \overleftarrow{\mathbf{V}}_{\tilde{e}_0'}^{-1} = \beta_0^{-1}\mathbf{B}^H\mathbf{B}, \tag{2.34}$$

$$\overleftarrow{\mathbf{V}}_{\tilde{e}_{K-1}''}^{-1}\overleftarrow{\mathbf{m}}_{\tilde{e}_{K-1}''} = \beta_K^{-1}\mathbf{T}^H\tilde{\mathbf{y}}_K, \qquad \overleftarrow{\mathbf{V}}_{\tilde{e}_{K-1}''}^{-1} = \beta_K^{-1}\mathbf{T}^H\mathbf{T}. \tag{2.35}$$

The backward message at $\tilde{e}^{'}$ in (2.30) and (2.31) and the message at $\tilde{e}^{''}$ in (2.32) and (2.33) are combined together at the equality node based on rules (II.3) and (II.1) in [29], yielding the backward message at point $\tilde{e}_k$ as

$$\overleftarrow{\mathbf{m}}_{\tilde{e}_k} = \overleftarrow{\mathbf{V}}_{\tilde{e}_k}\left(\overleftarrow{\mathbf{V}}_{\tilde{e}_k'}^{-1}\overleftarrow{\mathbf{m}}_{\tilde{e}_k'} + \overleftarrow{\mathbf{V}}_{\tilde{e}_k''}^{-1}\overleftarrow{\mathbf{m}}_{\tilde{e}_k''}\right), \tag{2.36}$$

$$\overleftarrow{\mathbf{V}}_{\tilde{e}_k} = \left(\overleftarrow{\mathbf{V}}_{\tilde{e}_k'}^{-1} + \overleftarrow{\mathbf{V}}_{\tilde{e}_k''}^{-1}\right)^{-1}. \tag{2.37}$$

**S4. Backward message at point $\tilde{b}_k$**

To reduce the complexity caused by matrix inversions, we approximate the two covariance matrices $\overrightarrow{\mathbf{V}}_{\tilde{e}_k}$ in (2.23) and $\overleftarrow{\mathbf{V}}_{\tilde{e}_k}$ in (2.37) as diagonal matrices, as follows.

$$\overrightarrow{\mathbf{V}}_{\tilde{e}_k} = \mu_k\mathbf{I}, \qquad \mu_k = \frac{1}{G}\sum_{j=0}^{G}\left(\tilde{\mathbf{H}}_k\right)_{j,j}\left(\tilde{\mathbf{H}}_k^H\right)_{j,j}, \tag{2.38}$$

and

$$\overleftarrow{\mathbf{V}}_{\tilde{e}_k} = \lambda_k\mathbf{I}, \tag{2.39}$$

$$\lambda_k = \frac{1}{G}\sum_{j=0}^{G}\left[(L-1)(\mu_{k-1}+\beta_k)+(P-L+1)\beta_k+(L-1)(\mu_{k+1}+\beta_{k+1})\right]. \tag{2.40}$$

But for the mean vectors, we still use the exact calculations, i.e. (2.22) for $\overrightarrow{\mathbf{m}}_{\tilde{e}_k}$ and a simpler form of (2.36) for $\overleftarrow{\mathbf{m}}_{\tilde{e}_k}$ as

$$\left(\overleftarrow{\mathbf{m}}_{\tilde{e}_k}\right)_i = \begin{cases} \left(\tilde{\mathbf{y}}_k\right)_i - \left(\overrightarrow{\mathbf{m}}_{\tilde{e}_{k-1}}\right)_i, & 0 \leqslant i \leqslant L-2 \\ \left(\tilde{\mathbf{y}}_k\right)_i, & L-1 \leqslant i \leqslant P-1 \\ \left(\tilde{\mathbf{y}}_{k+1}\right)_i - \left(\overrightarrow{\mathbf{m}}_{\tilde{e}_{k+1}}\right)_i, & P \leqslant i \leqslant G-1 \end{cases} \tag{2.41}$$

Thus, to obtain the extrinsic information at point $\tilde{b}_k$, we can follow the procedure in Section 2.2.1, i.e. (2.15), (2.16), (2.17) and (2.18), by replacing $(\overrightarrow{\mathbf{m}}_k^{apri}, \alpha_k)$ with $(\overrightarrow{\mathbf{m}}_{\tilde{b}_k}, \alpha_k')$, where $\alpha_k'$ is the average of the diagonal elements of the matrix in (2.21), $\alpha_k' = \frac{P}{G}\alpha_k$, and also replacing $(\mathbf{y}_k, \beta_k)$ with $(\overleftarrow{\mathbf{m}}_{\tilde{e}_k}, \lambda_k)$.

**S5. Combination**

Now we have obtained two extrinsic messages of $\mathbf{x}_k$, one from model (2.7) indicated by $^{ext1}$,

$$m_{x_{k,i}}^{ext1} = m_{b_{k,i}}^{ext1}, \quad v_{x_{k,i}}^{ext1} = v_{b_{k,i}}^{ext1}, \quad i = 0, ..., N-1, \tag{2.42}$$

and one from model (2.8) indicated by $^{ext2}$,

$$m_{x_{k,i}}^{ext2} = m_{\tilde{b}_{k,i-N+P}}^{ext2}, \quad v_{x_{k,i}}^{ext2} = v_{\tilde{b}_{k,i-N+P}}^{ext2}, \quad i = N-P, ..., N-1. \tag{2.43}$$

Applying rules (II.3) and (II.1) from [29] on a symbol level, we merge the two messages as

for $0 \leqslant i \leqslant N-P-1$,

$$m_{x_{k,i}}^{ext} = m_{x_{k,i}}^{ext1}, \quad v_{x_{k,i}}^{ext} = v_{x_{k,i}}^{ext1}, \tag{2.44}$$

for $N-P \leqslant i \leqslant N-1$,

$$m_{x_{k,i}}^{ext} = v_{x_{k,i}}^{ext} \left( m_{x_{k,i}}^{ext1} / v_{x_{k,i}}^{ext1} + m_{x_{k,i}}^{ext2} / v_{x_{k,i}}^{ext2} \right), \tag{2.45}$$

$$v_{x_{k,i}}^{ext} = \left( 1/v_{x_{k,i}}^{ext1} + 1/v_{x_{k,i}}^{ext2} \right)^{-1}. \tag{2.46}$$

### 2.2.3 Discussion

There are several scheduling methods in terms of the message passing in a graph. This chapter only demonstrates the simplest way, as shown in Section 2.2.1 and 2.2.2. In both sections, the extrinsic information coming from the soft decoder is taken as the *a priori* message for the equalizer, i.e. $\mathbf{m}_k^{apri}$ and $\mathbf{V}_k^{apri}$. However, one can easily replace the *a priori* message in one section by the *a posteriori* message obtained from the other section, e.g. replacing $\mathbf{m}_k^{apri}$ in (2.20) and $\mathbf{V}_k^{apri}$ in (2.21) by $\mathbf{m}_{b_k}$ from (2.15) and $\mathbf{V}_{b_k}$ from (2.16), respectively. In that case, a better *a priori* information of $\mathbf{x}_k$ is inputted to the equalizer, and a better extrinsic information may also be obtained at the output.

In regard to the complexity of the conventional equalization algorithm, the calculation of the *a posteriori* means $m_{x_{k,i}}^{post}$ is the most computationally intensive part. As we can see from (2.15), 2 DFT operations of length $N$ are required per block per iteration to get $\mathbf{m}_{b_k}$, since the calculation of $\mathbf{F}_N\mathbf{y}_k$ only needs to be done once for each block before the iteration starts. In implementation, the DFTs can be efficiently handled by FFTs, yielding a complexity of $\mathcal{O}(NlogN)$ per block per iteration. The calculation of the *a posteriori* variances $v_{x_{k,i}}^{post}$ in (2.19) and the abstraction of the extrinsic mean and variance in (2.17) and (2.18) are neglectable, since only scalar level computation is needed.

The extra complexity that the proposed algorithm brings about, mainly comes from the message passing for model (2.8). With the approximations we made in (2.38) and (2.40), the calculation of $\overleftarrow{\mathbf{V}}_{\tilde{e}_k}$ reduces to scalar level. From (2.41) and (2.22), we can see the calculation of $\overleftarrow{\mathbf{m}}_{\tilde{e}_k}$ results in 2 length-$G$ DFTs per block per iteration because the 4 DFTs in (2.41) are shared by two adjacent blocks. The repeating process mentioned in S4 in Section 2.2.2 generates 2 more length-$G$ DFTs as analyzed in the last paragraph. Thus the additional complexity that the proposed algorithm requires is $\mathcal{O}(GlogG)$ per block per iteration.

As a result, the proposed equalization algorithm makes use of models (2.7) and (2.8) and exploits all the observations at the receiver, but with proper approximations, it attains the same order of complexity as that of the conventional algorithm, i.e. $\mathcal{O}(NlogN)$ per block per iteration.

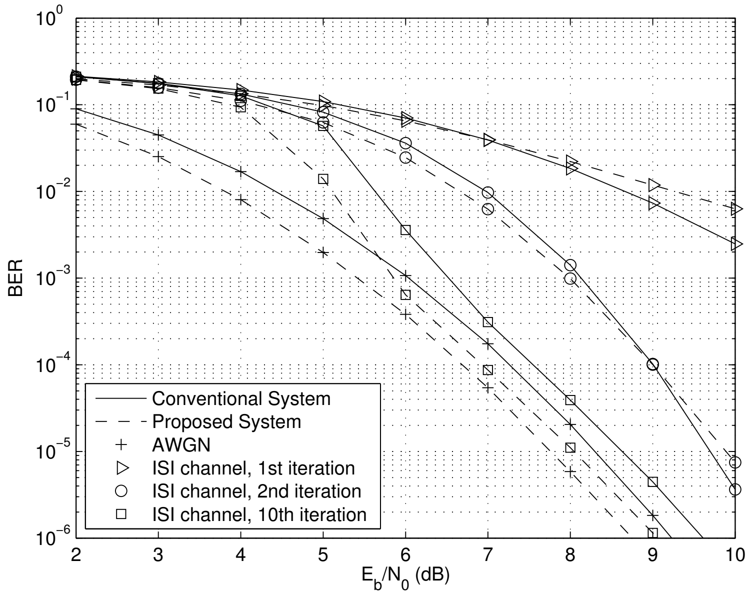


**Figure 2.7:** Performance comparison of the conventional system and the proposed system with 16QAM, Gray Mapping under AWGN/ISI channel.

## 2.3 Simulation Results

A rate-1/2 convolutional code $(23, 35)_8$, an S-random interleaver and the BCJR-based decoder are employed. Each frame is set to have 64 blocks and each block consists of 64 symbols, $N = 64$. With CP ratio 1/4, i.e. $P = 16$, each block is extended to have 80 symbols. The wide sense stationary uncorrelated scattering (WSSUS) model is adopted for the ISI channel. All the channel taps are independent of each other and the auto-correlation of each tap is the zeroth order Bessel function of the first kind. For each block, 17 coefficients are independently generated, $L = 17$. Perfect channel state information (CSI) is assumed to be available at the receiver. The channel energy is normalized to 1, i.e. $\mathrm{E}\left[||\mathbf{h}_k||^2\right] = 1$, and the uniform power delay profile applies. For each $E_b/N_0$ point, $10^5$ frames are simulated to get an average bit error rate (BER).

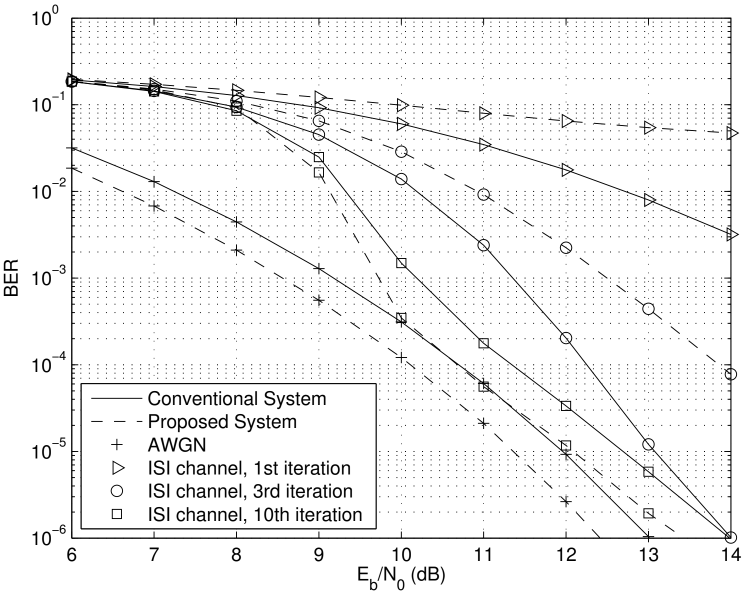


**Figure 2.8:** Performance comparison of the conventional system and the proposed system with 64QAM, Gray Mapping under AWGN/ISI channel.

Fig. 2.7 and Fig. 2.8 show the BER performance of the conventional system and the proposed system with 16QAM/64QAM modulation, Gray mapping over AWGN/ISI channels. As expected, the proposed equalization algorithm delivers a lower BER curve thanks to the extra information obtained from the CP. A gain of around 0.7dB is achieved with respect to the $E_b/N_0$ at $10^{-5}$, and it applies to both 16QAM and 64QAM modulations over both the AWGN channel and the ISI channel.

For both of the conventional system and the proposed system, the BER performance under the ISI channel improves as the number of iteration increases, and it eventually converges with that of the AWGN channel at high SNR. The only difference is that the proposed system ISI curves are bounded by lower AWGN curves. Note that the proposed system does not necessarily outperform the conventional one at all iterations. It becomes superior only after the second iteration for 16QAM and the fifth iteration for 64QAM. This is because, with the assumptions we made in Section 2.2.2, poor extrinsic information is passed out at the first several iterations from model (2.8). But as the iteration proceeds, better estimation is obtained and the proposed algorithm starts to surpass the conventional one.

## 2.4 Summary

This chapter considers the MMSE-based equalization algorithms in turbo FDE systems. By exploiting the information contained in the CP, the proposed equalization algorithm achieves a gain of around 0.7dB at 1/4 CP ratio for both 16QAM and 64QAM systems with Gray mapping over AWGN or ISI channel. In the meantime, the complexity is maintained at the same order as that of the conventional algorithm, i.e. $\mathcal{O}(NlogN)$ per block per iteration. Furthermore, such improvement can be anticipated for many other communication systems, such as the OFDM system, where cyclic prefix is also employed.

# Chapter 3

# Exploiting Cyclic Prefix in OFDM Systems With Known Channel

In OFDM systems, cyclic prefix insertion and removal enables the use of a set of computationally efficient single-tap equalizers at the receiver. Due to the extra transmission time and energy, the CP causes a loss in both spectrum efficiency and power efficiency. On the other hand, as a repetition of part of the data, the CP brings extra information and can be exploited for detection. Therefore, instead of discarding the CP observation as in the conventional OFDM system, we utilize all the received signals for equalization.

We consider the coded OFDM systems where turbo equalization is employed. We present the models of both the CP and the non-CP parts of the received signal in an FFG [29]. Then based on the computation rules of the FFG and the GMP technique [29], we develop an equalization algorithm, which exploits the prefix redundancy and enhances the system performance. With proper approximation, we propose a reduced-complexity algorithm without compromising the BER performance. The complexity of the reduced-complexity algorithm is $\mathcal{O}(2RNlogN + 4RGlogG + 2RG)$ per data block for $R$ iterations, where $N$ is the length of the data block and $G$ is equal to $P + L - 1$ with $P$ the length of the CP and $L$ the maximum delay spread of the channel. We summarize the contributions of this chapter as follows.

1. For exploiting the CP in OFDM systems, the equalizers proposed in the literature such as [55] and [56] are at least of $\mathcal{O}(N^3)$ complexity per data block per iteration, while our proposed reduced-complexity equalizer is only of $\mathcal{O}(NlogN)$ per data block per iteration, therefore making it desirable for use in practical applications.
2. Our proposed equalizer utilizes the full CP observation after interference cancellation and the tail of each block in the succeeding block (see the grey triangles in Fig. 3.2(b)) as well, while previous equalizers such as [50], [55] and [56] only make use of the CP or part of the CP for detection. As a result, simulation results show that our proposed equalizer achieves 1dB performance gain over the MMSE equalizers in [55] and [56] and 2dB gain over the MMSE equalizer in [50] for the system setup in Section V.
3. An SNR analysis is provided to verify the performance improvement of our proposed equalizer over the conventional equalizer. Such an analysis for exploiting the CP in OFDM systems, to the best of our knowledge, has not appeared in the literature before.
4. For AWGN channel, we have demonstrated that it is necessary to use an iterative receiver (i.e. turbo receiver) to take full advantage of the CP energy in OFDM systems. A bound of 0.97dB for 1/4 CP ratio is achieved within two iterations.
5. We have also investigated the sensitivity of the proposed equalizer to the quality of the available CSI. In the presence of channel estimation errors, our proposed equalizer is robust in terms of BER performance.

In the future, with the aid of interference cancellation among the signals from different transmit antennas, it is also possible to apply the proposed factor graph based approach and its associated approximation in multiple-input multiple-output (MIMO) OFDM systems[75].

## 3.1 System Model

Consider a coded OFDM system as depicted in Fig. 3.1. The information bit stream is encoded by a channel encoder, then the coded bit stream is interleaved and mapped onto an $M$-QAM symbol sequence $\mathbf{u}$, $\mathbf{u} = [u_0, u_1, ..., u_{Z-1}]^T$. This sequence is partitioned into $K$ blocks, each of length $N$, where we assume $Z = KN$. The $k$th block is denoted by $\mathbf{u}_k = [u_{kN}, u_{kN+1}, ..., u_{(k+1)N-1}]^T$, $k = 0, ..., K-1$. An $N$-point IFFT is applied on $\mathbf{u}_k$, producing $\mathbf{x}_k = \mathbf{F}_N^H \mathbf{u}_k$. By copying the last $P$ samples of $\mathbf{x}_k$ to its front, the CP is formed, which we denote by $\tilde{\mathbf{x}}_k = \mathbf{A}\mathbf{x}_k$, where $\mathbf{A} = \begin{bmatrix} \mathbf{0}_{P\times(N-P)} & \mathbf{I}_{P\times P} \end{bmatrix}$. For the CP to be effective, $P \geq L-1$ is required, where $L$ is the maximum delay spread of the multipath channel. The extended block $\mathbf{s}_k = [x_{(k+1)N-P}, ..., x_{(k+1)N-1}, x_{kN}, ..., x_{(k+1)N-1}]^T$, is of length $M$, $M = N + P$. The symbol sequence $\mathbf{s} = [s_0, s_1, ..., s_{J-1}]^T = [\mathbf{s}_0^T, \mathbf{s}_1^T, ..., \mathbf{s}_{K-1}^T]^T$, where $J = KM$, is transmitted over the channel, and at the receiver side, the sequence $\mathbf{r} = [r_0, r_1, ..., r_{V-1}]^T$, where $V = J + L - 1$, is observed, and

$$r_j = \sum_{l=0}^{L-1} h_{j,l} \cdot s_{j-l} + n_j, \quad j = 0, ..., V-1 \tag{3.1}$$

where $\{h_{j,l}, l = 0, ..., L-1\}$ is the CSI at time $j$, and $\{n_j\}$ is the AWGN noise with zero mean and variance $2\sigma^2$. In this chapter, we assume that the channel is quasi-static[24], which means that the channel remains the same within the duration of a block but it may vary from block to block.

As shown in Fig. 3.2, corresponding to $\mathbf{s}_k$, the counterpart block at the receiver $\mathbf{r}_k = [r_{kM}, r_{kM+1}, ..., r_{(k+1)M+L-2}]^T$, is always $L-1$ symbols longer than $\mathbf{s}_k$ because of

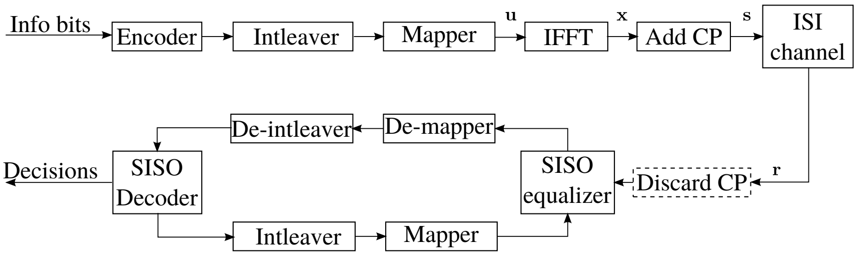


**Figure 3.1:** Block diagram of a coded OFDM system.

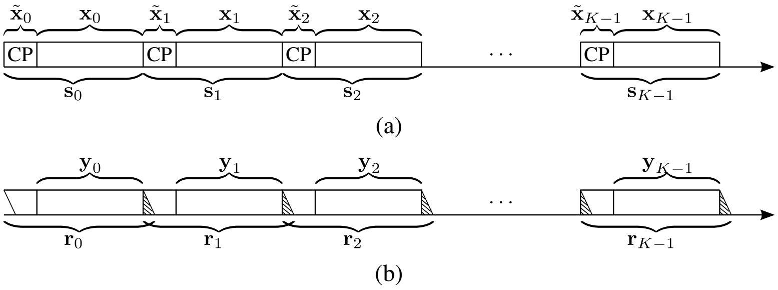


**Figure 3.2:** Data structure of a coded OFDM system. (a) Transmitted signal (b) Received signal

the multipath channel. In conventional OFDM systems, the CP is usually removed at the receiver, and only the part $\mathbf{y}_k = [r_{kM+P}, r_{kM+P-1}, ..., r_{(k+1)M-1}]^T$ is used for further processing. This part can be modeled as the circular convolution of the data block $\mathbf{x}_k$ and the channel $\mathbf{h}_k$ as follows.

$$\mathbf{y}_k = \mathbf{h}_k \circledast \mathbf{x}_k + \mathbf{n}_k, \quad k = 0, ..., K-1 \tag{3.2}$$

where "$\circledast$" 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 vector with mean $\mathbf{0}$ and covariance $2\sigma^2\mathbf{I}$.

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. It is of length $G = P + L - 1$ and denoted by

$\tilde{\mathbf{y}}_k = [r_{kM}, r_{kM+1}, ..., r_{kM+P-1}, r_{(k+1)M}, ..., r_{(k+1)M+L-2}]^T$. Note that $\tilde{\mathbf{y}}_k$ contains the interference from adjacent blocks $\mathbf{s}_{k-1}$ and $\mathbf{s}_{k+1}$, which we denote by $\mathbf{w}_{k-1}^{tail}$ and $\mathbf{w}_{k+1}^{head}$, respectively. Let $tail(\cdot)$ and $head(\cdot)$ be two truncation functions, which return the $L-1$ tail and the $L-1$ head symbols of the sequence in the parentheses, respectively. Then the interference terms $\mathbf{w}_{k-1}^{tail}$ and $\mathbf{w}_{k+1}^{head}$ can be given by

$$\mathbf{w}_{k-1}^{tail} =[tail(\mathbf{h}_{k-1} * \tilde{\mathbf{x}}_{k-1}), \mathbf{0}_{1\times P}]^T \tag{3.3}$$

$$\mathbf{w}_{k+1}^{head} =[\mathbf{0}_{1\times P}, head(\mathbf{h}_{k+1} * \tilde{\mathbf{x}}_{k+1})]^T \tag{3.4}$$

for $k = 1, ..., K-2$, where "$*$" denotes the linear convolution. For $k = 0$ and $k = K-1$, we have

$$\mathbf{w}_{-1}^{tail} =\mathbf{0}_{G\times 1}, \qquad \mathbf{w}_{K}^{head} = \mathbf{0}_{G\times 1}. \tag{3.5}$$

Modeled as the linear convolutions of the block $\tilde{\mathbf{x}}_k$ and the channel $\mathbf{h}_k$, this part of observation, $\tilde{\mathbf{y}}_k$, can be written as

$$\tilde{\mathbf{y}}_k = \mathbf{h}_k * \tilde{\mathbf{x}}_k + \mathbf{w}_{k-1}^{tail} + \mathbf{w}_{k+1}^{head} + \tilde{\mathbf{n}}_k, \qquad k = 0, ..., K-1 \tag{3.6}$$

where $\tilde{\mathbf{n}}_k$ denotes the AWGN vector of length $G$. For notational simplicity, we put the last three terms of (3.6) together as one error term

$$\mathbf{v}_k =\mathbf{w}_{k-1}^{tail} + \mathbf{w}_{k+1}^{head} + \tilde{\mathbf{n}}_k. \tag{3.7}$$

As a result, (3.6) becomes

$$\tilde{\mathbf{y}}_k = \mathbf{h}_k * \tilde{\mathbf{x}}_k + \mathbf{v}_k. \tag{3.8}$$

Since linear convolution after proper zero-padding can be transformed into circular convolution, and circular convolution in the time domain can be transformed into multiplication in the frequency domain by DFT operation, we can rewrite (3.2) and (3.8) as

$$\mathbf{y}_k = \mathbf{F}_N^H \mathbf{H}_k \mathbf{F}_N \mathbf{x}_k + \mathbf{n}_k, \tag{3.9}$$

$$\tilde{\mathbf{y}}_k = \mathbf{F}_G^H \tilde{\mathbf{H}}_k \mathbf{F}_G \mathbf{Q} \tilde{\mathbf{x}}_k + \mathbf{v}_k \tag{3.10}$$

where $\mathbf{Q} = \begin{bmatrix} \mathbf{I}_{P\times P} & \mathbf{0}_{P\times(L-1)} \end{bmatrix}^T$ is to adapt $\tilde{\mathbf{x}}_k$ for the DFT operation by adding $L-1$ zeros to its end and

$$\mathbf{H}_k = Diag\left(\sqrt{N}\mathbf{F}_N \hat{\mathbf{h}}_k\right), \quad \text{with} \quad \hat{\mathbf{h}}_k = \begin{bmatrix} \mathbf{h}_k^T & \mathbf{0}_{1\times(N-L)} \end{bmatrix}^T, \tag{3.11}$$

$$\tilde{\mathbf{H}}_k = Diag\left(\sqrt{G}\mathbf{F}_G \check{\mathbf{h}}_k\right), \quad \text{with} \quad \check{\mathbf{h}}_k = \begin{bmatrix} \mathbf{h}_k^T & \mathbf{0}_{1\times(G-L)} \end{bmatrix}^T. \tag{3.12}$$

Substituting $\mathbf{x}_k = \mathbf{F}_N^H \mathbf{u}_k$ and $\tilde{\mathbf{x}}_k = \mathbf{A}\mathbf{x}_k$ into (3.9) and (3.10), respectively, we have

$$\mathbf{y}_k = \mathbf{F}_N^H \mathbf{H}_k \mathbf{u}_k + \mathbf{n}_k, \tag{3.13}$$

$$\tilde{\mathbf{y}}_k = \mathbf{F}_G^H \tilde{\mathbf{H}}_k \mathbf{B} \mathbf{u}_k + \mathbf{v}_k \tag{3.14}$$

with $\mathbf{B} = \mathbf{F}_G \mathbf{Q} \mathbf{A} \mathbf{F}_N^H$. The term $\mathbf{v}_k$ can also be written in a similar form as

$$\mathbf{v}_k = \tilde{\mathbf{n}}_k + \mathbf{C}\mathbf{F}_G^H \tilde{\mathbf{H}}_{k-1} \mathbf{B} \mathbf{u}_{k-1} + \mathbf{C}^H \mathbf{F}_G^H \tilde{\mathbf{H}}_{k+1} \mathbf{B} \mathbf{u}_{k+1} \tag{3.15}$$

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

$$\mathbf{C} = \begin{bmatrix} \mathbf{0}_{(L-1)\times P} & \mathbf{I}_{(L-1)\times(L-1)} \\ \mathbf{0}_{P\times P} & \mathbf{0}_{P\times(L-1)} \end{bmatrix}. \tag{3.16}$$

## 3.2 Factor Graph Based Equalization Algorithms

In this section, we present the models (3.13) and (3.14) in a factor graph, and based on the computation rules of the graph, we develop an equalization algorithm that utilizes all the received signals.

### 3.2.1 FFG of Model (3.13) and Model (3.14)

According to model (3.13) and model (3.14), we can draw an FFG as illustrated in Fig. 3.3. The conventional equalizer only uses the $\mathbf{y}_k$ part of the observation, which corresponds to model (3.13) and the right branch of the graph, whereas the proposed equalizer utilizes both $\mathbf{y}_k$ and $\tilde{\mathbf{y}}_k$, which are related to models (3.13) and (3.14), which are represented by the right branch and the left branch of the graph, respectively.

In each iteration of the receiver, the soft decoder outputs extrinsic log-likelihood ratios (LLRs) about the coded bits, which are then interleaved and converted into probabilities about the data symbols. Characterized by the mean vector $\overrightarrow{\mathbf{m}}_{a_k}$ and the covariance matrix $\overrightarrow{\mathbf{V}}_{a_k}$, these probabilities are fed into the equalizer as the *a priori* information. Then based on the *a priori* information, the observation and the channel information, the equalizer produces extrinsic means $\mathbf{m}_k^{ext}$ and variances $\mathbf{V}_k^{ext}$ about the data symbols. These means and variances are passed onto the de-mapper where they are converted back to LLRs. Then these LLRs are de-interleaved and fed into the soft decoder serving as the *a priori* information.

At the first iteration, we set the *a priori* information of the equalizer to $\overrightarrow{\mathbf{m}}_{a_k} = \mathbf{0}$ and $\overrightarrow{\mathbf{V}}_{a_k} = \mathbf{I}$ because *a*) the $M$-QAM constellation points $\mathcal{A}_i$, $i = 0, ..., M-1$, are assumed to occur equally likely, and the mean of a symbol is $\frac{1}{M}\sum_{i=0}^{M-1}\mathcal{A}_i = 0$, the variance of a symbol is $\frac{1}{M}\sum_{i=0}^{M-1}|\mathcal{A}_i|^2 = E$, and the symbol energy $E$ is normalized to 1, i.e. $E = 1$, *b*) the equalizer is assumed to know about *a*) at the first iteration, but there is no information coming from the decoder. For other iterations, it is a common practice to model $\overrightarrow{\mathbf{V}}_{a_k}$ as a diagonal matrix, since symbols within a block can be treated as independent of each other due to the use of the interleaver.

For the conventional equalizer when applied in an iterative system, the *a priori*

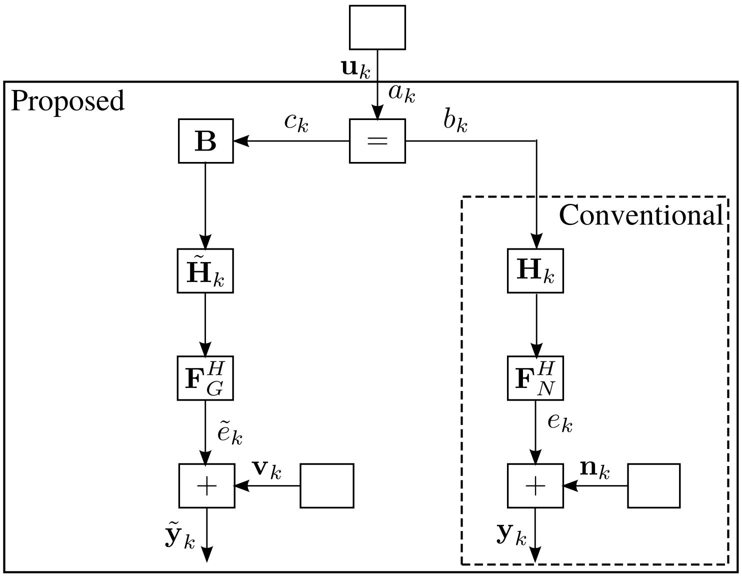


**Figure 3.3:** The $k$th block of the Forney-style factor graph for both conventional and proposed equalizers in an OFDM system.

information $(\overrightarrow{\mathbf{m}}_{a_k}, \overrightarrow{\mathbf{V}}_{a_k})$ is of no use in the calculation of the extrinsic information $(\mathbf{m}_k^{ext}, \mathbf{V}_k^{ext})$. All the sub-carriers are independent of each other in the frequency domain, and the *a priori* information of one symbol does not contribute to the extrinsic information of the others in the same block. However, for the proposed equalizer, the *a priori* information is very useful. Details are provided in the following two subsections, where the message passing process is described and points $a_k$, $b_k$, $c_k$, $e_k$ and $\tilde{e}_k$ in Fig. 3.3 are used to indicate the location of the message in discussion.

### 3.2.2 Message Passing for the Conventional Equalizer

1. As the noise is with zero mean and $2\sigma^2$ variance, following the rules (II.10) and (II.8) in [29], the backward message at point $e_k$ is given by

$$\overleftarrow{\mathbf{m}}_{e_k} = \mathbf{y}_k, \quad \overleftarrow{\mathbf{V}}_{e_k} = \beta \mathbf{I}_N \tag{3.17}$$

where $\beta = 2\sigma^2$. Applying the rules (III.6) and (III.5) in [29], we have the backward message at point $b_k$ as

$$\overleftarrow{\mathbf{W}}_{b_k}\overleftarrow{\mathbf{m}}_{b_k} = \beta^{-1}\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k, \tag{3.18}$$

$$\overleftarrow{\mathbf{W}}_{b_k} = \beta^{-1}\mathbf{H}_k^H\mathbf{H}_k \tag{3.19}$$

where $\overleftarrow{\mathbf{W}}_{b_k}\overleftarrow{\mathbf{m}}_{b_k}$ is the transformed mean vector, and $\overleftarrow{\mathbf{W}}_{b_k} = \overleftarrow{\mathbf{V}}_{b_k}^{-1}$ is the weight matrix. This message can also be written in the mean/covariance form as

$$\overleftarrow{\mathbf{m}}_{b_k} = \mathbf{H}_k^{-1}\mathbf{F}_N\mathbf{y}_k, \tag{3.20}$$

$$\overleftarrow{\mathbf{V}}_{b_k} = \Big[\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k\Big]^{-1}. \tag{3.21}$$

2. Combining the backward message at point $b_k$ with the *a priori* message of $\mathbf{u}_k$, we have the *a posteriori* message or marginal message of $\mathbf{u}_k$ as

$$\mathbf{W}_{\mathbf{u}_k}\mathbf{m}_{\mathbf{u}_k} = \overleftarrow{\mathbf{W}}_{b_k}\overleftarrow{\mathbf{m}}_{b_k} + \overrightarrow{\mathbf{V}}_{a_k}^{-1}\overrightarrow{\mathbf{m}}_{a_k}, \tag{3.22}$$

$$\mathbf{W}_{\mathbf{u}_k} = \overleftarrow{\mathbf{W}}_{b_k} + \overrightarrow{\mathbf{V}}_{a_k}^{-1}. \tag{3.23}$$

To extract the extrinsic information of each symbol in $\mathbf{u}_k$, we use the relations (14) and (15) in [25]. Since $\overleftarrow{\mathbf{V}}_{b_k}$ and $\overrightarrow{\mathbf{V}}_{a_k}$ are diagonal, the extrinsic variances and means of the data symbols in the $k$th block, $k = 0, ..., K-1$, can be written as

$$\mathbf{V}_k^{ext} = \Big[\mathbf{W}_{\mathbf{u}_k} - \overrightarrow{\mathbf{V}}_{a_k}^{-1}\Big]^{-1} = \overleftarrow{\mathbf{V}}_{b_k}, \tag{3.24}$$

$$\mathbf{m}_k^{ext} = \mathbf{V}_k^{ext}\Big[\mathbf{W}_{\mathbf{u}_k}\mathbf{m}_{\mathbf{u}_k} - \overrightarrow{\mathbf{V}}_{a_k}^{-1}\overrightarrow{\mathbf{m}}_{a_k}\Big] = \overleftarrow{\mathbf{m}}_{b_k}. \tag{3.25}$$

Note that from (3.24) and (3.25), the extrinsic information is exactly the same as the backward message at point $b_k$, and that there is no term associated with the *a priori* information in the expression of the extrinsic information, i.e. the *a priori* information does not contribute to the extrinsic information in the conventional equalizer case.

Once the extrinsic means and variances about the symbols are available, we can use (16) in [25] to calculate the LLRs about the coded bits. The equalizer presented here is the same as the conventional soft-output equalizer mentioned in [75].

### 3.2.3 Message Passing for the Proposed Equalizer

1. As shown in Fig. 3.3, the right branch of the graph for the proposed equalizer is exactly the same as the graph for the conventional equalizer. Therefore, the forward message at point $c_k$ is the same as the *a posteriori* message of $\mathbf{u}_k$ in the conventional equalizer. Using (3.22), (3.23) and (3.18), (3.19), we have

$$\overrightarrow{\mathbf{W}}_{c_k}\overrightarrow{\mathbf{m}}_{c_k} = \overrightarrow{\mathbf{V}}_{a_k}^{-1}\overrightarrow{\mathbf{m}}_{a_k} + \beta^{-1}\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k, \tag{3.26}$$

$$\overrightarrow{\mathbf{W}}_{c_k} = \overrightarrow{\mathbf{V}}_{a_k}^{-1} + \beta^{-1}\mathbf{H}_k^H\mathbf{H}_k, \tag{3.27}$$

or in the mean/covariance form as

$$\overrightarrow{\mathbf{m}}_{c_k} = \overrightarrow{\mathbf{V}}_{c_k}\Big(\overrightarrow{\mathbf{V}}_{a_k}^{-1}\overrightarrow{\mathbf{m}}_{a_k} + \beta^{-1}\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k\Big), \tag{3.28}$$

$$\overrightarrow{\mathbf{V}}_{c_k} = \Big(\overrightarrow{\mathbf{V}}_{a_k}^{-1} + \beta^{-1}\mathbf{H}_k^H\mathbf{H}_k\Big)^{-1}. \tag{3.29}$$

2. Using rules (II.10) and (II.8) in [29], we have the backward message at point $\tilde{e}_k$ as

$$\overleftarrow{\mathbf{m}}_{\tilde{e}_k} = \tilde{\mathbf{y}}_k - \overrightarrow{\mathbf{m}}_{\mathbf{v}_k}, \quad \overleftarrow{\mathbf{V}}_{\tilde{e}_k} = \overrightarrow{\mathbf{V}}_{\mathbf{v}_k}. \tag{3.30}$$

The mean vector of $\mathbf{v}_k$ in (3.15) is updated using $\overrightarrow{\mathbf{m}}_{c_{k-1}}$ and $\overrightarrow{\mathbf{m}}_{c_{k+1}}$, which are the latest mean vectors about $\mathbf{u}_{k-1}$ and $\mathbf{u}_{k+1}$,

$$\overrightarrow{\mathbf{m}}_{\mathbf{v}_k} = \mathbf{C}\mathbf{F}_G^H\boldsymbol{\xi}_{k-1} + \mathbf{C}^H\mathbf{F}_G^H\boldsymbol{\xi}_{k+1} \tag{3.31}$$

with $\boldsymbol{\xi}_k = \tilde{\mathbf{H}}_k \mathbf{B} \overrightarrow{\mathbf{m}}_{c_k}$. Using (3.15), we approximate the covariance matrix of $\mathbf{v}_k$ by

$$\overrightarrow{\mathbf{V}}_{\mathbf{v}_k} = \lambda_k \mathbf{I}_G \tag{3.32}$$

where

$$\lambda_k = 2\sigma^2 + \frac{1}{G} \sum_{l=0}^{L-1} \left[ \gamma_{k+1} (L-1-l) \left| h_{k+1}^l \right|^2 + \gamma_{k-1} l \left| h_{k-1}^l \right|^2 \right], \tag{3.33}$$

$$\gamma_k = \frac{1}{N} \sum_{i=0}^{N-1} \left( \overrightarrow{\mathbf{W}}_{c_k}^{-1} \right)_{i,i}. \tag{3.34}$$

Based on rules (III.6) and (III.5) in [29], the backward message at point $c_k$ is

$$\overleftarrow{\mathbf{W}}_{c_k} \overleftarrow{\mathbf{m}}_{c_k} = \lambda_k^{-1} \mathbf{B}^H \tilde{\mathbf{H}}_k^H \mathbf{F}_G \left( \tilde{\mathbf{y}}_k - \overrightarrow{\mathbf{m}}_{\mathbf{v}_k} \right), \tag{3.35}$$

$$\overleftarrow{\mathbf{W}}_{c_k} = \lambda_k^{-1} \boldsymbol{\Psi}_k \tag{3.36}$$

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

3. To obtain the *a posteriori* message at point $c_k$, we first have the covariance matrix as

$$\mathbf{V}_{c_k} = \left( \overrightarrow{\mathbf{W}}_{c_k} + \overleftarrow{\mathbf{W}}_{c_k} \right)^{-1} = \left( \overrightarrow{\mathbf{V}}_{a_k}^{-1} + \beta^{-1} \mathbf{H}_k^H \mathbf{H}_k + \lambda_k^{-1} \boldsymbol{\Psi}_k \right)^{-1}. \tag{3.37}$$

For the mean vector, by substituting rule (I.1) into rule (I.6) in [29], we get $\mathbf{m} = \overrightarrow{\mathbf{m}} + \mathbf{V}\big(\overleftarrow{\mathbf{W}}\overleftarrow{\mathbf{m}} - \overleftarrow{\mathbf{W}}\overrightarrow{\mathbf{m}}\big)$. Using (3.35) and (3.36), the *a posteriori* mean vector at point $c_k$ is given by

$$\begin{aligned} \mathbf{m}_{c_k} =& \overrightarrow{\mathbf{m}}_{c_k} + \mathbf{V}_{c_k} \big( \overleftarrow{\mathbf{W}}_{c_k} \overleftarrow{\mathbf{m}}_{c_k} - \overleftarrow{\mathbf{W}}_{c_k} \overrightarrow{\mathbf{m}}_{c_k} \big) \\ =& \overrightarrow{\mathbf{m}}_{c_k} + \lambda_k^{-1} \mathbf{V}_{c_k} \mathbf{B}^H \tilde{\mathbf{H}}_k^H \Big[ \mathbf{F}_G \big( \tilde{\mathbf{y}}_k - \overrightarrow{\mathbf{m}}_{\mathbf{v}_k} \big) - \tilde{\mathbf{H}}_k \mathbf{B} \overrightarrow{\mathbf{m}}_{c_k} \Big]. \end{aligned} \tag{3.38}$$

Then according to rules (II.5) and (II.6) in [29], we have the *a posteriori* message of $\mathbf{u}_k$ as

$$\mathbf{m}_{\mathbf{u}_k} = \mathbf{m}_{c_k}, \quad \mathbf{V}_{\mathbf{u}_k} = \mathbf{V}_{c_k}. \tag{3.39}$$

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 [25] as

$$\mathbf{V}_k^{ext} = \left\{ \left[ \left(\mathbf{V}_{\mathbf{u}_k}\right)_{diag} \right]^{-1} - \overrightarrow{\mathbf{V}}_{a_k}^{-1} \right\}^{-1}, \tag{3.40}$$

$$\mathbf{m}_k^{ext} = \mathbf{V}_k^{ext} \left\{ \left[ \left(\mathbf{V}_{\mathbf{u}_k}\right)_{diag} \right]^{-1} \mathbf{m}_{\mathbf{u}_k} - \overrightarrow{\mathbf{V}}_{a_k}^{-1} \overrightarrow{\mathbf{m}}_{a_k} \right\}. \tag{3.41}$$

As shown in (3.39) and (3.37), different from the conventional equalizer, the *a posteriori* covariance matrix $\mathbf{V}_{\mathbf{u}_k}$ for the proposed equalizer is not diagonal, and the *a posteriori* variances of the data symbols are given by the diagonal entries of $\mathbf{V}_{\mathbf{u}_k}$. Due to the existence of the non-zero off-diagonal entries in $\mathbf{V}_{\mathbf{u}_k}$, the *a posteriori* information of a symbol includes the *a priori* information of other symbols in the data block, and even after subtracting the *a priori* information of its own (see (3.40) and (3.41)), the *a priori* information of other symbols is still contained in the extrinsic information of the symbol.

We summarize the proposed algorithm as Parts A, B, and C1 in Table 3.1. Note that this equalizer features a higher latency than the conventional equalizer and some of the works in the literature such as [50], [55] and [58]. However, such a latency is intrinsic to a turbo receiver due to the use of the interleaver/de-interleaver. In a turbo equalization system, the whole sequence associated with a codeword must be processed before it can be passed on to the equalizer or the decoder.

Note also that Fig. 3.3 is cycle-free since it only depicts the $k$th block of the whole Forney-style factor graph. If we consider all the $k$ blocks, $k = 0, ..., K-1$, the dependence of $\mathbf{v}_k$ on $\mathbf{u}_{k-1}$ and $\mathbf{u}_{k+1}$ in model (3.15) results in cycles in the

**Table 3.1:** The proposed equalization algorithms with and without approximation

```
A. Initialization:
  1. The blocks $\mathbf{y}_k$, $\tilde{\mathbf{y}}_k$, $k = 0, ..., K-1$, are observed. The channel state information $\mathbf{h}_k$, $k = 0, ..., K-1$ and the noise variance $\beta = 2\sigma^2$ are either assumed to be known or made available by estimation.
  2. At each iteration, the a priori information $\big(\overrightarrow{\mathbf{m}}_{a_k}, \overrightarrow{\mathbf{V}}_{a_k}\big)$ is fed into the equalizer and the following steps are executed to calculate the extrinsic information $\big(\mathbf{m}_k^{ext}, \mathbf{V}_k^{ext}\big)$. Variables associated with indices $-1$ and $K$ are treated as zero.
B. Calculations of $\overrightarrow{\mathbf{W}}_{c_k}$, $\overrightarrow{\mathbf{m}}_{c_k}$, $\lambda_k$ and $\boldsymbol{\xi}_k$:
  1. for $k = 0, ..., K-1$
       (a) $\overrightarrow{\mathbf{W}}_{c_k} = \overrightarrow{\mathbf{V}}_{a_k}^{-1} + \beta^{-1}\mathbf{H}_k^H\mathbf{H}_k$
       (b) $\gamma_k = \frac{1}{N}\sum_{i=0}^{N-1}\big(\overrightarrow{\mathbf{W}}_{c_k}^{-1}\big)_{i,i}$
       (c) $\overrightarrow{\mathbf{m}}_{c_k} = \overrightarrow{\mathbf{W}}_{c_k}^{-1}\Big[\overrightarrow{\mathbf{V}}_{a_k}^{-1}\overrightarrow{\mathbf{m}}_{a_k} + \beta^{-1}\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k\Big]$
     end for
  2. for $k = 0, ..., K-1$
       (a) $\lambda_k = \beta + \frac{1}{G}\sum_{l=0}^{L-1}\Big[\gamma_{k+1}(L-1-l)\left|h_{k+1}^l\right|^2 + \gamma_{k-1}l\left|h_{k-1}^l\right|^2\Big]$
       (b) $\boldsymbol{\xi}_k = \tilde{\mathbf{H}}_k\mathbf{B}\overrightarrow{\mathbf{m}}_{c_k}$
     end for
C1. Exact calculations of $\mathbf{V}_k^{ext}$ and $\mathbf{m}_k^{ext}$:
     for $k = 0, ..., K-1$
       (a) $\mathbf{V}_{\mathbf{u}_k} = \Big[\overrightarrow{\mathbf{W}}_{c_k} + \lambda_k^{-1}\boldsymbol{\Psi}_k\Big]^{-1}$
       (b) $\mathbf{m}_{\mathbf{u}_k} = \overrightarrow{\mathbf{m}}_{c_k} + \lambda_k^{-1}\mathbf{V}_{\mathbf{u}_k}\mathbf{B}^H\tilde{\mathbf{H}}_k^H\Big[\mathbf{F}_G\big(\tilde{\mathbf{y}}_k - \mathbf{C}\mathbf{F}_G^H\boldsymbol{\xi}_{k-1} - \mathbf{C}^H\mathbf{F}_G^H\boldsymbol{\xi}_{k+1}\big) - \boldsymbol{\xi}_k\Big]$
       (c) $\mathbf{V}_k^{ext} = \Big\{\Big[\big(\mathbf{V}_{\mathbf{u}_k}\big)_{diag}\Big]^{-1} - \overrightarrow{\mathbf{V}}_{a_k}^{-1}\Big\}^{-1}$
       (d) $\mathbf{m}_k^{ext} = \mathbf{V}_k^{ext}\Big\{\Big[\big(\mathbf{V}_{\mathbf{u}_k}\big)_{diag}\Big]^{-1}\mathbf{m}_{\mathbf{u}_k} - \overrightarrow{\mathbf{V}}_{a_k}^{-1}\overrightarrow{\mathbf{m}}_{a_k}\Big\}$
     end for
C2. Approximated calculations of $\mathbf{V}_k^{ext}$ and $\mathbf{m}_k^{ext}$:
     for $k = 0, ..., K-1$
       (a) $\mathbf{V}_k^{ext} = \Big[\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\Big]^{-1}$
       (b) $\mathbf{m}_k^{ext} = \mathbf{V}_k^{ext}\Big\{\beta^{-1}\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\overrightarrow{\mathbf{m}}_{c_k} + \lambda_k^{-1}\mathbf{B}^H\tilde{\mathbf{H}}_k^H$
               $\cdot\Big[\mathbf{F}_G\big(\tilde{\mathbf{y}}_k - \mathbf{C}\mathbf{F}_G^H\boldsymbol{\xi}_{k-1} - \mathbf{C}^H\mathbf{F}_G^H\boldsymbol{\xi}_{k+1}\big) - \boldsymbol{\xi}_k\Big]\Big\}$
     end for
```

**Table 3.2:** Complexity of the proposed algorithm in Section 3.2.3 and the proposed reduced-complexity algorithm in Section 3.2.4

| The Proposed Algorithm | | The Proposed Reduced-complexity Algorithm | |
|---|---|---|---|
| $\mathbf{H}_k$, $\tilde{\mathbf{H}}_k$<br>$\mathbf{\Psi}_k$ | $\mathcal{O}(NlogN + GlogG)$<br>$\mathcal{O}(N^3)$ | $\mathbf{H}_k$, $\tilde{\mathbf{H}}_k$<br>$\mathbf{\Phi}_k$ | $\mathcal{O}(NlogN + GlogG)$<br>$\mathcal{O}(NlogN + GlogG)$ |
| $\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k$ | $\mathcal{O}(NlogN + N)$ | $\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k$ | $\mathcal{O}(NlogN + N)$ |
| $\boldsymbol{\xi}_k$<br>$\mathbf{V}_{\mathbf{u}_k}$<br>$\mathbf{m}_{\mathbf{u}_k}$ | $\mathcal{O}(RNlogN + RGlogG + RG)$<br>$\mathcal{O}(RN^3)$<br>$\mathcal{O}(RN^2 + RNlogN + 3RGlogG + RG)$ | $\boldsymbol{\xi}_k$<br>$\mathbf{m}_k^{ext}$ | $\mathcal{O}(RNlogN + RGlogG + RG)$<br>$\mathcal{O}(RNlogN + 3RGlogG + RG)$ |

graph. Thus, in order to use the GMP technique (which is a special case of the sum-product algorithm [27]), some scheduling schemes regarding the message passing must be employed. [27] suggests a serial scheduling scheme and a flooding scheduling scheme. The algorithms shown in Table 3.1 can be treated as a hybrid scheduling scheme, which passes messages within a block in a serial manner and passes messages between blocks in a flooding manner.

### 3.2.4 The Proposed Reduced-Complexity Algorithm

In general, for an $N$-point block, the complexity of the FFT operation is $\mathcal{O}(NlogN)$, while matrix multiplication is $\mathcal{O}(N^3)$, matrix-vector multiplication is $\mathcal{O}(N^2)$ and matrix inversion is $\mathcal{O}(N^3)$. Therefore, it is desirable to avoid the matrix multiplication, matrix-vector multiplication and matrix inversion and use FFT or other means of implementation instead. As seen from (3.36), in the proposed algorithm there is a full matrix $\mathbf{\Psi}_k$, and its calculation involves not only FFT operation but also matrix multiplication. Moreover, because of the existence of $\mathbf{\Psi}_k$, matrix inversion occurs in the calculation of $\mathbf{V}_{\mathbf{u}_k}$ (see C1.a in Table 3.1) too. Since the calculation of $\mathbf{m}_{\mathbf{u}_k}$ and $\mathbf{V}_{\mathbf{u}_k}$ needs to be done every block and in every iteration, a very high computational demand is incurred.

To reduce the complexity, we make the following approximation

$$\mathbf{V}_{\mathbf{u}_k} = \left[\overrightarrow{\mathbf{W}}_{c_k} + \lambda_k^{-1}\boldsymbol{\Phi}_k\right]^{-1} \tag{3.42}$$

where $\boldsymbol{\Phi}_k = \left(\boldsymbol{\Psi}_k\right)_{diag}$. Since $\overrightarrow{\mathbf{W}}_{c_k}$ is diagonal, $\mathbf{V}_{\mathbf{u}_k}$ becomes a diagonal matrix too. However, its diagonal entries are still different from the exact values (i.e. $\left[\left(\overrightarrow{\mathbf{W}}_{c_k} + \lambda_k^{-1}\boldsymbol{\Psi}_k\right)^{-1}\right]_{diag}$), due to the matrix inversion.

With the approximation of (3.42), the full matrix inversion originally involved in C1.a in Table 3.1 is turned into the computation of scalar reciprocals, and with $\mathbf{V}_{\mathbf{u}_k}$ approximated to be a diagonal matrix, the matrix-vector multiplication that previously exists in C1.b in Table 3.1 is also reduced to scalar multiplications, i.e.

$$\begin{aligned}\mathbf{m}_{\mathbf{u}_k} =& \overrightarrow{\mathbf{m}}_{c_k} + \lambda_k^{-1}\left[\overrightarrow{\mathbf{W}}_{c_k} + \lambda_k^{-1}\boldsymbol{\Phi}_k\right]^{-1}\mathbf{B}^H\tilde{\mathbf{H}}_k^H \\ &\cdot\left[\mathbf{F}_G\big(\tilde{\mathbf{y}}_k - \mathbf{C}\mathbf{F}_G^H\boldsymbol{\xi}_{k-1} - \mathbf{C}^H\mathbf{F}_G^H\boldsymbol{\xi}_{k+1}\big) - \boldsymbol{\xi}_k\right].\end{aligned} \tag{3.43}$$

Furthermore, the calculation of $\boldsymbol{\Phi}_k$ can be accomplished by a fast approach presented in Appendix A, where full matrix multiplication is avoided and only one $G$-point FFT and one $N$-point FFT are required.

Substituting (3.42) and (3.43) into C1.c and C1.d in Table 3.1, we can write the extrinsic variances and means of the data symbols as

$$\mathbf{V}_k^{ext} = \left[\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\right]^{-1} \tag{3.44}$$

$$\begin{aligned}\mathbf{m}_k^{ext} =& \mathbf{V}_k^{ext}\Big\{\beta^{-1}\mathbf{H}_k^H\mathbf{F}_N\mathbf{y}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\overrightarrow{\mathbf{m}}_{c_k} + \lambda_k^{-1}\mathbf{B}^H\tilde{\mathbf{H}}_k^H \\ &\cdot\left[\mathbf{F}_G\big(\tilde{\mathbf{y}}_k - \mathbf{C}\mathbf{F}_G^H\boldsymbol{\xi}_{k-1} - \mathbf{C}^H\mathbf{F}_G^H\boldsymbol{\xi}_{k+1}\big) - \boldsymbol{\xi}_k)\right]\Big\}.\end{aligned} \tag{3.45}$$

The proposed reduced-complexity equalization algorithm is summarized as Parts A, B and C2 in Table 3.1.

To evaluate the reduction of complexity due to the use of approximation (3.42), we count the complex multiplications required for processing one data block for $R$ iterations between the equalizer and the decoder. In Table 3.2, a breakdown list of the complexity is provided for the proposed algorithm with and without approximation. There are three types of computation that significantly contribute to the overall complexity.

1. $\mathbf{H}_k$, $\tilde{\mathbf{H}}_k$ and $\boldsymbol{\Psi}_k$ for the proposed algorithm without approximation, and $\mathbf{H}_k$, $\tilde{\mathbf{H}}_k$ and $\boldsymbol{\Phi}_k$ for the proposed reduced-complexity algorithm, need to be calculated once per CSI update;
2. $\mathbf{H}_k^H \mathbf{F}_N \mathbf{y}_k$ needs to be calculated once per received signal block for both with and without approximation;
3. $\boldsymbol{\xi}_k$, $\mathbf{V}_{\mathbf{u}_k}$ and $\mathbf{m}_{\mathbf{u}_k}$ for the proposed algorithm without approximation, and $\boldsymbol{\xi}_k$ and $\mathbf{m}_k^{ext}$ for the proposed reduced-complexity algorithm, need to be calculated once per iteration.

Without approximation (3.42), the complexity of the proposed algorithm is dominated by cubic terms, i.e. $\mathcal{O}(RN^3 + N^3)$ per data block for $R$ iterations, while with approximation, its overall complexity is dominated by the third type and only a few FFTs are needed, i.e. $\mathcal{O}(2RNlogN + 4RGlogG + 2RG)$ per data block for $R$ iterations.

## 3.3 SNR Performance Analysis

To enable the SNR analysis, we rewrite $\mathbf{m}_k^{ext}$ into the following form

$$\mathbf{m}_k^{ext} = \mathbf{u}_k + \boldsymbol{\zeta}_k \tag{3.46}$$

where $\mathbf{u}_k$ is the $k$th transmitted data block, and $\boldsymbol{\zeta}_k$ is the estimation error vector.

To distinguish from the mean and covariance of a Gaussian message in an FFG, we use $\mathrm{E}(\boldsymbol{\zeta}_k)$ and $\mathrm{Var}(\boldsymbol{\zeta}_k)$ to denote the mean vector and covariance matrix of $\boldsymbol{\zeta}_k$.

When $\mathrm{E}(\boldsymbol{\zeta}_k)$ is zero, the SNR of the $i$th sub-carrier in the $k$th block is given by

$$\mathrm{SNR}_{k,i} = \frac{P_{k,i}}{\mathrm{Var}(\boldsymbol{\zeta}_k)_{i,i}}, \quad i = 0, ..., N-1, \quad k - 0, ..., K-1 \tag{3.47}$$

where $P_{k,i}$ is the power of the data symbol and $\mathrm{Var}(\boldsymbol{\zeta}_k)_{i,i}$ is the variance of the estimate of the data symbol. Note that only the diagonal entries of $\mathrm{Var}(\boldsymbol{\zeta}_k)$ are of our interest.

### 3.3.1 SNR Analysis for the Proposed Reduced-Complexity Algorithm

To rewrite (3.45) in the form of (3.46), we *a*) substitute (3.13), (3.14) into (3.45), *b*) replace $\mathbf{v}_k$ with $\mathbf{v}_k = \overrightarrow{\mathbf{m}}_{\mathbf{v}_k} + \boldsymbol{\omega}_k$ where $\boldsymbol{\omega}_k \sim (\mathbf{0}, \overrightarrow{\mathbf{V}}_{\mathbf{v}_k})$, and *c*) replace $\overrightarrow{\mathbf{m}}_{a_k}$ with $\overrightarrow{\mathbf{m}}_{a_k} = \mathbf{u}_k + \boldsymbol{\epsilon}_k$ where $\boldsymbol{\epsilon}_k \sim (\mathbf{0}, \overrightarrow{\mathbf{V}}_{a_k})$. As a result, the estimation error $\boldsymbol{\zeta}_k$ is obtained as

$$\begin{aligned}\boldsymbol{\zeta}_k = & \Big[\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\Big]^{-1}\Big[\beta^{-1}\big(\mathbf{I} + \boldsymbol{\Theta}_k\big)\mathbf{H}_k^H\mathbf{F}_N\mathbf{n}_k \\ & + \boldsymbol{\Theta}_k\overrightarrow{\mathbf{V}}_{a_k}^{-1}\boldsymbol{\epsilon}_k + \lambda_k^{-1}\mathbf{B}^H\tilde{\mathbf{H}}_k^H\mathbf{F}_G\boldsymbol{\omega}_k\Big]\end{aligned} \tag{3.48}$$

where

$$\boldsymbol{\Theta}_k = \lambda_k^{-1}\big(\boldsymbol{\Phi}_k - \boldsymbol{\Psi}_k\big)\big(\overrightarrow{\mathbf{V}}_{a_k}^{-1} + \beta^{-1}\mathbf{H}_k^H\mathbf{H}_k\big)^{-1}. \tag{3.49}$$

Since $\mathrm{E}(\mathbf{n}_k) = \mathbf{0}_{N\times 1}$, $\mathrm{E}(\boldsymbol{\omega}_k) = \mathbf{0}_{G\times 1}$ and $\mathrm{E}(\boldsymbol{\epsilon}_k) = \mathbf{0}_{N\times 1}$, the error vector $\boldsymbol{\zeta}_k$ has zero means, i.e. $\mathrm{E}(\boldsymbol{\zeta}_k) = \mathbf{0}_{N\times 1}$. As shown in Appendix B, we have the diagonal entries of $\mathrm{Var}(\boldsymbol{\zeta}_k)$ for the proposed reduced-complexity algorithm as follows.

$$\begin{aligned}\mathrm{Var}(\boldsymbol{\zeta}_k)_{diag}^{prop} = & \big(\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\big)^{-1} + \lambda_k^{-2}\big(\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\big)^{-1} \\ & \cdot \Big[\big(\boldsymbol{\Phi}_k - \boldsymbol{\Psi}_k\big)\big(\overrightarrow{\mathbf{V}}_{a_k}^{-1} + \beta^{-1}\mathbf{H}_k^H\mathbf{H}_k\big)^{-1}\big(\boldsymbol{\Phi}_k - \boldsymbol{\Psi}_k\big)\Big]_{diag} \\ & \cdot \big(\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\big)^{-1}\end{aligned} \tag{3.50}$$

### 3.3.2 SNR Analysis for the Conventional Algorithm

To rewrite (3.25) in the form of (3.46), we substitute (3.13) into (3.25) and get

$$\mathbf{m}_k^{ext} = \mathbf{u}_k + \underbrace{\mathbf{H}_k^{-1}\mathbf{F}_N\mathbf{n}_k}_{\boldsymbol{\zeta}_k}. \tag{3.51}$$

Since $\mathrm{E}(\mathbf{n}_k) = \mathbf{0}_{N\times 1}$, the estimation error term $\boldsymbol{\zeta}_k$ has zero means, i.e. $\mathrm{E}(\boldsymbol{\zeta}_k) = \mathbf{0}_{N\times 1}$. The covariance matrix of $\boldsymbol{\zeta}_k$ is

$$\mathrm{Var}(\boldsymbol{\zeta}_k)^{conv} = \left[\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k\right]^{-1} \tag{3.52}$$

and it is diagonal.

### 3.3.3 SNR Improvement

The SNR improvement of the proposed reduced-complexity algorithm over the conventional algorithm can be evaluated for each iteration by substituting (3.50) and (3.52) into

$$\frac{\mathrm{SNR}_{k,i}^{prop}}{\mathrm{SNR}_{k,i}^{conv}} = \frac{P_{k,i}/\operatorname{Var}(\boldsymbol{\zeta}_k)_{i,i}^{prop}}{P_{k,i}/\operatorname{Var}(\boldsymbol{\zeta}_k)_{i,i}^{conv}} = \frac{\operatorname{Var}(\boldsymbol{\zeta}_k)_{i,i}^{conv}}{\operatorname{Var}(\boldsymbol{\zeta}_k)_{i,i}^{prop}}. \tag{3.53}$$

As with iterations the *a priori* covariance matrix $\overrightarrow{\mathbf{V}}_{a_k}$ gets very small in the high SNR regime, (3.50) becomes

$$\mathrm{Var}(\boldsymbol{\zeta}_k)_{diag}^{prop} = \left(\beta^{-1}\mathbf{H}_k^H\mathbf{H}_k + \lambda_k^{-1}\boldsymbol{\Phi}_k\right)^{-1}. \tag{3.54}$$

and the SNR gain for the $i$th sub-carrier in the $k$th block is given by

$$\frac{\mathrm{SNR}_{k,i}^{prop}}{\mathrm{SNR}_{k,i}^{conv}} = \frac{\mu_{k,i} + \tau_{k,i}}{\mu_{k,i}} = 1 + \frac{\tau_{k,i}}{\mu_{k,i}} \tag{3.55}$$

where $\mu_{k,i} = \beta^{-1}\left|(\mathbf{H}_k)_{i,i}\right|^2$, $\tau_{k,i} = \lambda_k^{-1}(\boldsymbol{\Phi}_k)_{i,i}$, for $i = 0, ..., N-1$, $k = 0, ..., K-1$.

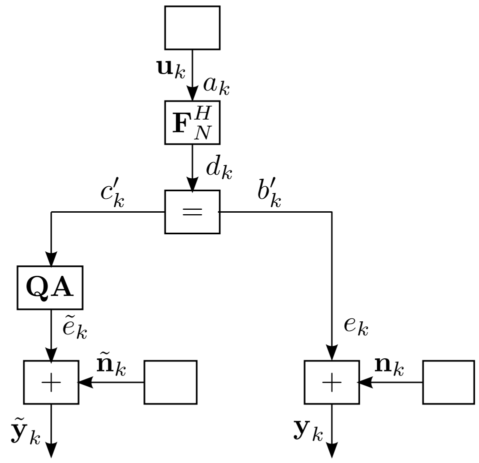


**Figure 3.4:** The $k$th block of the Forney-style factor graph for the proposed equalizer over AWGN channel.

### SNR Improvement over AWGN Channel

For the AWGN channel, we have $\mathbf{H}_k = \mathbf{I}_N$, $\tilde{\mathbf{H}}_k = \mathbf{I}_G$, and $\lambda_k = \beta = 2\sigma^2$, and the matrix $\mathbf{\Psi}_k$ is reduced to $\mathbf{F}_N\mathbf{A}^H\mathbf{A}\mathbf{F}_N^H$ with its diagonal entries all equal to the CP ratio $r_{CP} = P/N$. The proposed reduced-complexity algorithm improves the SNR of the $i$th sub-carrier in the $k$th block by

$$\frac{\mathrm{SNR}_{k,i}^{prop}}{\mathrm{SNR}_{k,i}^{conv}} = 1 + r_{CP} \tag{3.56}$$

for $i = 0, ..., N-1$ and $k = 0, ..., K-1$. For example, when $r_{CP} = 1/4$, we have $\mathrm{SNR}_{k,i}^{prop}/\mathrm{SNR}_{k,i}^{conv} = 1 + 1/4 = 0.97$dB.

Note that the proposed algorithm enhances the SNRs of all the sub-carriers equally by $10log_{10}(1+r_{CP})$ dB, and thus the overall system performance gain is also $10log_{10}(1+r_{CP})$ dB.

Note also that $10log_{10}(1+r_{CP})$ dB is the upper bound of the gain that the $r_{CP}$-ratio CP can offer over the AWGN channel, and iterations are essential for achieving such a bound. Although there is no ISI between data blocks for the AWGN channel, the use of the CP observation introduces mutual dependence between the symbols within a data block. From (3.13) and (3.14) with $\mathbf{H}_k = \mathbf{I}_N$ and $\tilde{\mathbf{H}}_k = \mathbf{I}_G$ over the AWGN channel, we can draw the factor graph for the proposed equalizer as Fig. 3.4. Since the covariance matrix $\overrightarrow{\mathbf{V}}_{a_k}$ is diagonal (see Section 3.2.1), the *a priori* message of $\mathbf{u}_k$ indicates that the symbols within a block are independent of each other. However, the backward message of $\mathbf{u}_k$ indicates some dependence between the symbols as its associated weight matrix is not diagonal. Details are as follows.

1. From (3.17), the backward message at point $b'_k$ has covariance matrix $\overleftarrow{\mathbf{V}}_{b'_k} = \beta\mathbf{I}_N$.
2. Using (3.30) with $\overrightarrow{\mathbf{m}}_{\mathbf{v}_k} = \mathbf{0}$ and rule (III.5) in [29], the backward message at point $c'_k$ has weight matrix $\overleftarrow{\mathbf{W}}_{c'_k} = \beta^{-1}\mathbf{A}^H\mathbf{A}$, and it is a diagonal matrix but not a scaled identity matrix.
3. Using rule (II.1) in [29], the backward message at point $d_k$ has weight matrix $\overleftarrow{\mathbf{W}}_{d_k} = \beta^{-1}\left(\mathbf{A}^H\mathbf{A} + \mathbf{I}_N\right)$, and it is also a diagonal matrix but not a scaled identity matrix.
4. Using rule (III.5) in [29], the backward message at point $a_k$ has weight matrix $\overleftarrow{\mathbf{W}}_{a_k} = \beta^{-1}\left(\mathbf{F}_N\mathbf{A}^H\mathbf{A}\mathbf{F}_N^H + \mathbf{I}_N\right)$, and it is a full matrix (specifically, a Toeplitz matrix).

The non-zero off-diagonal entries of $\overleftarrow{\mathbf{W}}_{a_k}$ demonstrate that there is mutual dependence between the symbols within a data block. As a result, the extrinsic information that the equalizer produces is relevant to the *a priori* information fed back from the decoder, and that leads to performance improvement of the turbo equalization system during the iteration process.

If the proposed reduced-complexity algorithm is employed (i.e. with approximation (3.42)), the extrinsic covariance matrix $\mathbf{V}_k^{ext}$ does not contain any term related to the *a priori* message of $\mathbf{u}_k$ anymore (see C2.a in Table 3.1), but the extrinsic

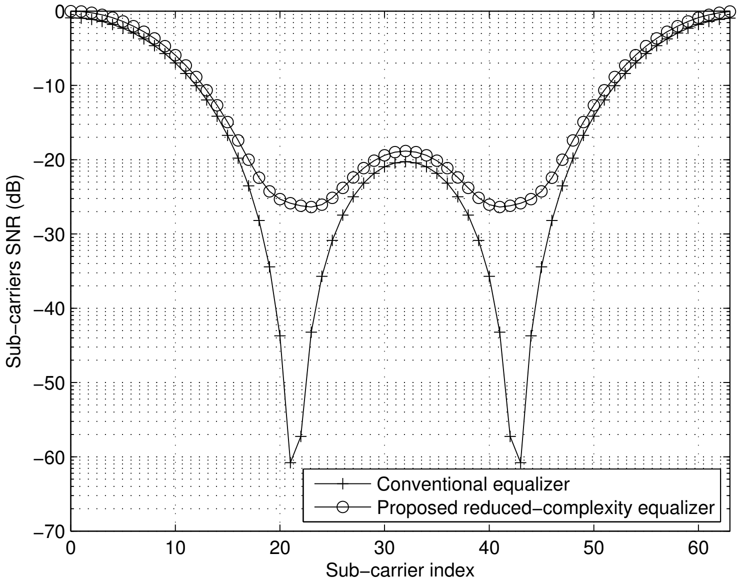


**Figure 3.5:** Sub-carrier SNR improvement over Proakis C channel. Sub-carrier SNRs are normalized by the maximum sub-carrier SNR of both equalizers.

mean vector $\mathbf{m}_k^{ext}$ still does (see C2.b in Table 3.1). As will be shown in Fig. 3.6, with $r_{CP} = 1/4$, the proposed reduced-complexity equalizer achieves only 0.7dB gain over the conventional equalizer after the first iteration, and two iterations are needed to reach the bound of gain 0.97dB.

Note that if the IFFT node (i.e. the $\mathbf{F}_N^H$ node) is removed, Fig. 3.4 becomes the factor graph of the SC-FDE system [69] (where the CP is also being exploited). In that case, there is no performance improvement with iteration for the AWGN channel since the weight matrix $\overleftarrow{\mathbf{W}}_{a_k} = \beta^{-1}\left(\mathbf{A}^H\mathbf{A} + \mathbf{I}_N\right)$ of the backward message of $\mathbf{u}_k$ is a diagonal matrix and no mutual dependence is introduced.

### SNR Improvement over ISI Channels

For ISI channels, we use Proakis C channel [1] as an example to show how the SNR is improved by the proposed algorithm. This channel has 5 taps in the time domain, $\mathbf{h} = [0.227, 0.46, 0.688, 0.46, 0.227]^T$, and in the frequency domain, it has two deep fades as can be seen from Fig. 3.5. In conventional systems, it is hard to recover data from sub-carriers in these fades. However, by adding another branch in the FFG (the left branch in Fig. 3.3), the proposed algorithm can obtain extra information from the CP observation, which enhances the SNRs of the sub-carriers in the fades by as much as 30dB (see Fig. 3.5). Other sub-carriers' SNRs are also improved at the same time but only marginally. The unevenness of the SNR gains over different sub-carriers makes it hard to predict the overall system performance gain under ISI channels. However, as an example we will simulate a convolutional coded OFDM system over the Proakis C channel and show the numerical results in the next section.

## 3.4 Simulation Results

If not stated otherwise, the simulation settings are as follows. A rate-1/2 convolutional code $(23, 35)$, an S-random interleaver, a 4QAM modulator with Gray mapping and the BCJR-based decoder [17] are employed. Each data sequence consists of 64 blocks, i.e. $K = 64$, and each data block has 64 symbols before CP insertion, i.e. $N = 64$. With 1/4 ratio, the CP length is 16, i.e. $P = 16$. For each $E_b/N_0$ point, at least $10^7$ sequences are simulated to get an average BER.

There are three types of channels simulated, the AWGN channel, the Proakis C channel, and a random 17-tap channel. For the former two, the channel is fixed and every block experiences the same channel. For the third type, 17 complex coefficients are generated independently as the channel taps for each block, i.e. $L = 17$, and these coefficients vary from block to block. For all the three types, the energy of the channel is normalized to 1, i.e. $\mathrm{E}\left[\frac{1}{L}\sum_{l=0}^{L-1}\left|h_k^l\right|^2\right] = 1$, for $k = 0, ..., K-1$. Moreover, uniform power delay profile applies for the random 17-tap channel.

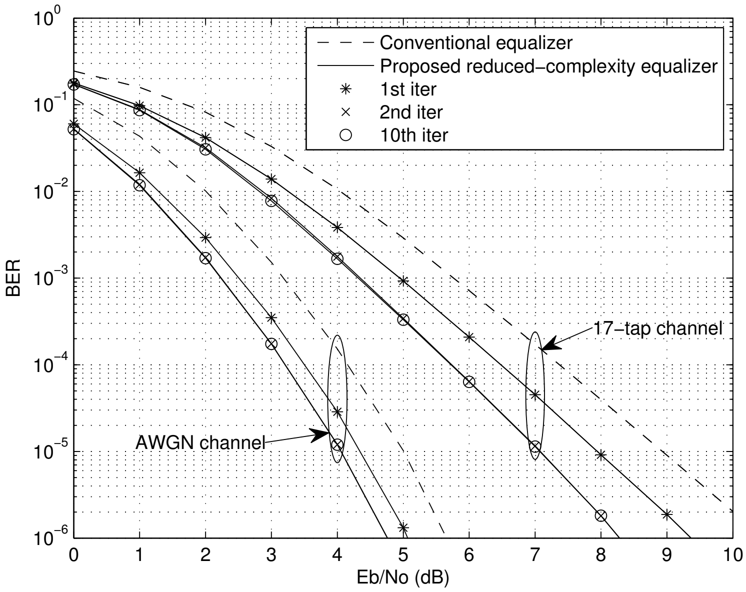


**Figure 3.6:** BER performance of the conventional equalizer and the proposed reduced-complexity equalizer over the AWGN channel and the random 17-tap channel.

### 3.4.1 Performance with Perfect CSI

In Fig. 3.6, the BER performance of the proposed reduced-complexity algorithm is compared with that of the conventional algorithm (i.e. without using the CP) over the AWGN channel and the random 17-tap channel. As the performance of the conventional algorithm does not improve with iterations, only the BER of its first iteration is plotted. For the proposed algorithm, we can see that the BER decreases dramatically within the first two iterations, but after that the improvement is very small. Compared with the conventional algorithm at the BER of $10^{-5}$, the proposed algorithm achieves 0.97dB gain over the AWGN channel and 2dB gain over the random 17-tap channel after convergence. Note that as discussed in Section IV.C.1, 0.97dB is the upper bound of the SNR gain that the 1/4-ratio CP can offer over the AWGN channel.

Fig. 3.7 shows the frame error rate (FER) of both algorithms. At the BER of

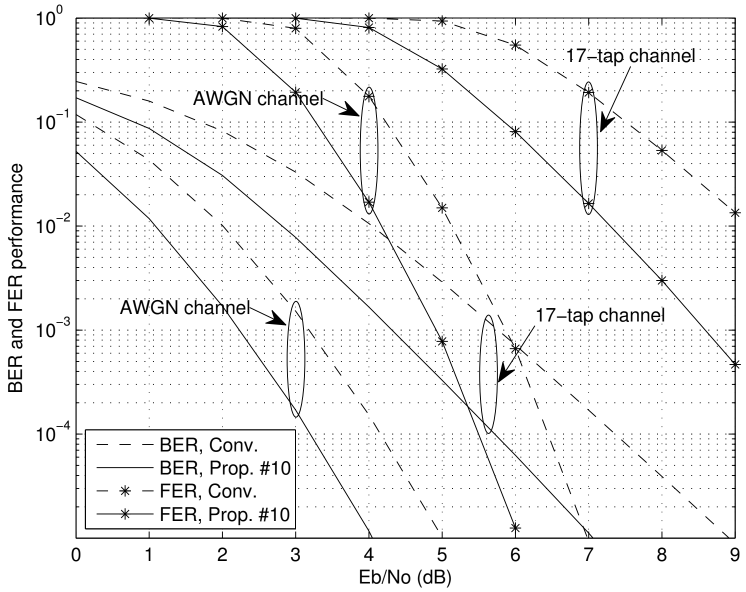


**Figure 3.7:** BER and FER performance of the conventional algorithm (conv.) and the proposed reduced-complexity algorithm with 10 iterations (prop. #10) over the AWGN and 17-tap channels.

$10^{-5}$, the FER of the proposed algorithm is around $10^{-2}$, and compared with the conventional algorithm, the proposed algorithm still obtains a gain of 0.97dB over the AWGN channel and 2dB over the 17-tap channel.

Fig. 3.8 shows the improvement that the proposed reduced-complexity algorithm provides over the Proakis C channel. This channel has two deep fades in the frequency domain as shown in Fig. 3.5. By exploiting the extra information in the CP, SNRs of the sub-carriers in the deep fades are enormously enhanced, and after two iterations the proposed algorithm obtains 6dB gain at the BER of $10^{-5}$ compared with the conventional algorithm.

Fig. 3.9 compares the proposed reduced-complexity algorithm and the conventional algorithm when 64QAM with Gray mapping, and the rate-1/2 convolutional code (133,171) and block interleaver specified in IEEE 802.11a [76] are employed.

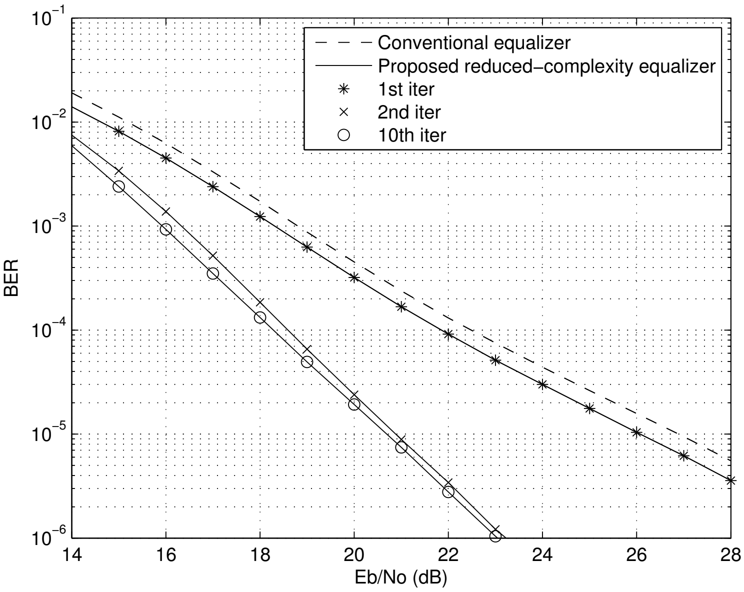


**Figure 3.8:** BER performance of the conventional equalizer and the proposed reduced-complexity equalizer over the Proakis C channel.

Compared with Fig. 3.6, a similar gain is achieved by the proposed algorithm in this case, i.e. 0.97dB gain over the AWGN channel and 2dB gain over the random 17-tap channel at the BER of $10^{-5}$.

### 3.4.2 Performance with Channel Estimation Error

This set of simulations are to investigate the sensitivity of the proposed algorithm to the quality of the available CSI. Three scenarios, with severe, moderate and mild channel estimation error for the random 17-tap channel, are simulated, and their performance is compared with that of the perfect CSI scenario in Fig. 3.10.

As in [77], the estimate for the $l$th tap of the channel is modeled as $\hat{h}^l = h^l + n^l$, $l = 0, ..., L-1$, where $h^l$ denotes the true channel tap coefficient, $\hat{h}^l$ the estimate and $n^l$ the estimation error. The error $n^l$ is assumed to be complex Gaussian with zero-mean and variance $\sigma_l^2$, and statistically independent for different $l$'s. Since uniform power delay profile is applied in the 17-tap channel, the power of the error for

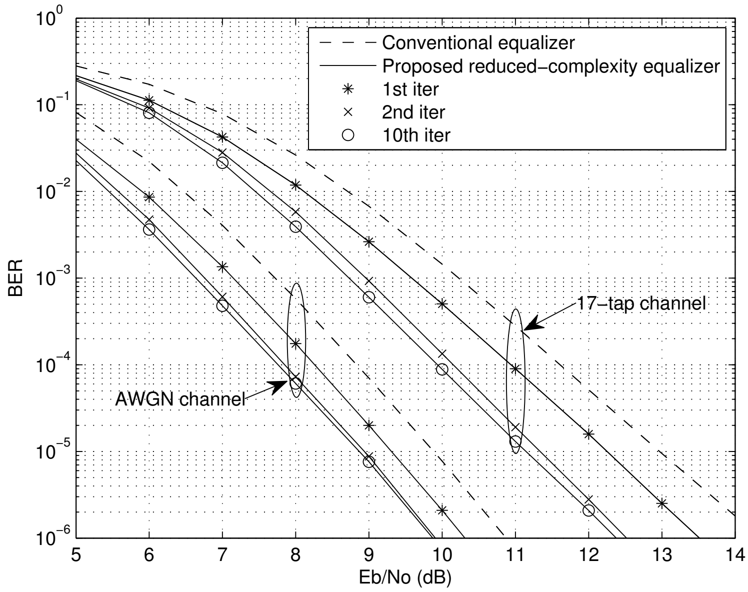


**Figure 3.9:** BER performance of the conventional equalizer and the proposed reduced-complexity equalizer over the AWGN channel and the random 17-tap channel, with 64QAM and Gray mapping, (133,171) convolutional code and block interleaver [76].

different taps is assumed to be equal, i.e. $\sigma_l^2 = \sigma^2$, $l = 0, ..., L-1$. The mean-square error (MSE) of the channel estimation is defined as $\sum_{l=0}^{L-1} |\hat{h}^l - h^l|^2$, and therefore in our case it is equal to $L\sigma^2$. Three different MSE values, $10^{-1}$, $10^{-2}$ and $10^{-3}$, are simulated, respectively, as the severe, moderate and mild channel estimation error scenarios.

As shown in Fig. 3.10, at the BER of $10^{-5}$, there is a performance loss of around 4.5dB for the severe channel estimation error scenario. When the MSE reduces to $10^{-2}$, the loss is reduced to within 0.5dB, and for the mild channel estimation error scenario, we can hardly see any performance loss.

Note that the assumed imperfect channel knowledge is partial, since the number and position of the impulse response samples are perfectly known and the uncertainty only lies on the values of its samples. Now we investigate the case where the

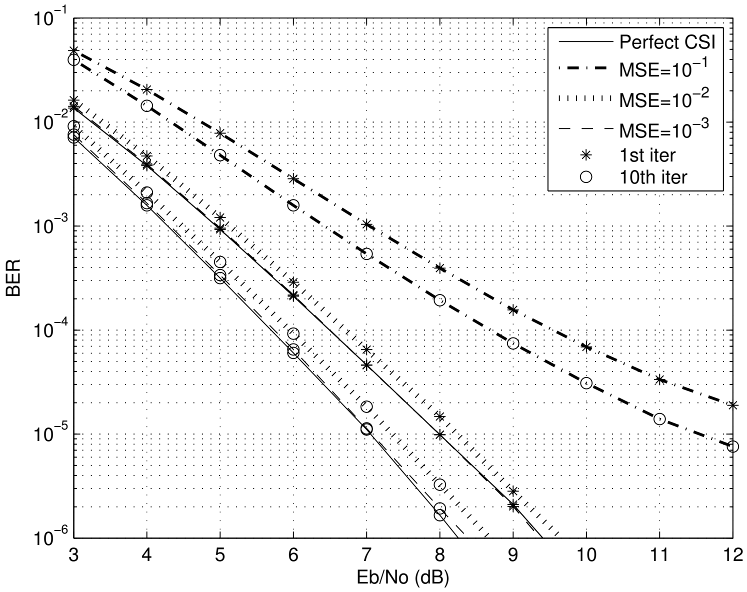


**Figure 3.10:** BER performance of the proposed reduced-complexity algorithm with different qualities of CSI over the random 17-tap channel.

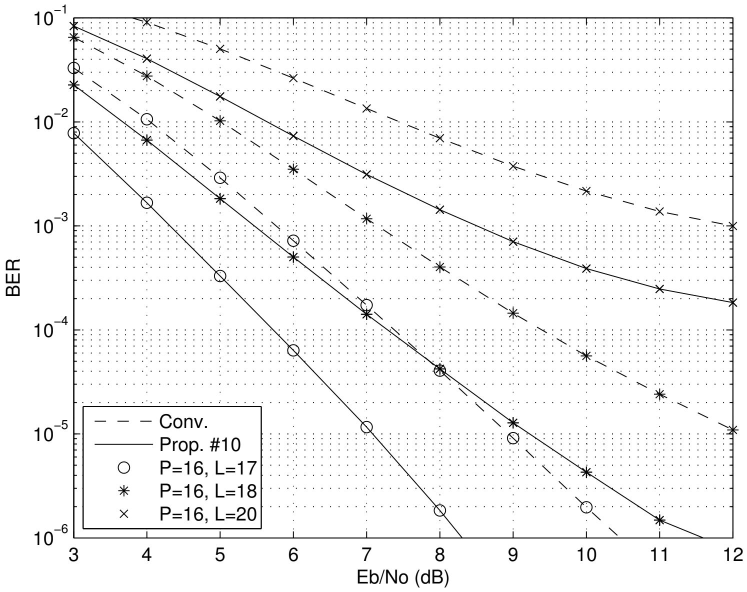


**Figure 3.11:** BER performance of the conventional algorithm (conv.) and the proposed reduced-complexity algorithm with 6 iterations (prop. #6) over ISI channels with $L = 17$, $18, 20$ and uniform power delay profile.

channel delay spread $L$ exceeds the CP length $P$. In the simulation, the CP length is $P = 16$ and the channel delay spread is $L = 17, 18, 20$. We assume that the first $P+1$ taps are known at the receiver, but the last $L-P-1$ taps that fall out of the CP duration are treated as zero by the receiver, i.e. they are treated as interferences that are not cancelled out.

Fig. 3.11 shows the BER performance of the proposed reduced-complexity algorithm over the ISI channels with uniform power delay profile (PDF), and Fig. 3.12 and Fig. 3.13 are for the exponentially decaying PDFs, $q_l = e^{-0.1l}$ and $q_l = e^{-0.3l}$, respectively, where $q_l$ is the power of the $l$-th tap of the ISI channel, $l = 0, ..., L-1$. The energy of the channel is normalized to 1 in all three figures. We can see that the proposed algorithm is more robust to this kind of channel estimation error when the channel has exponentially decaying PDF than when it has uniform PDF.

### 3.4.3 Performance Comparison with [50], [55] and [56]

In this set of simulations, we apply the methods in [50], [55] and [56] in turbo equalization systems and compare their performance with our proposed reduced-complexity algorithm.

The work [50] uses the 'clear' part of the CP observation, and under the settings of this chapter, i.e. $P = 16$ and $L = 17$, there is no 'clear' symbols left to be exploited, therefore, the scheme in [50] is reduced to the conventional scheme. As shown in Fig. 3.14, over the random 17-tap channel, the MMSE solution from [50] performs the same as the conventional system (see Fig. 3.6), and its least square (LS) solution leads to an error floor due to the error propagation.

The works [55] and [56] make use of the first $P$ symbols of the vector $\tilde{\mathbf{y}}_k$ after interference cancellation, while our proposed algorithm also utilizes the last $L-1$ symbols of $\tilde{\mathbf{y}}_k$ after interference cancellation, therefore exploiting more information about the CP part of data. Fig. 3.14 compares the performance of both the LS

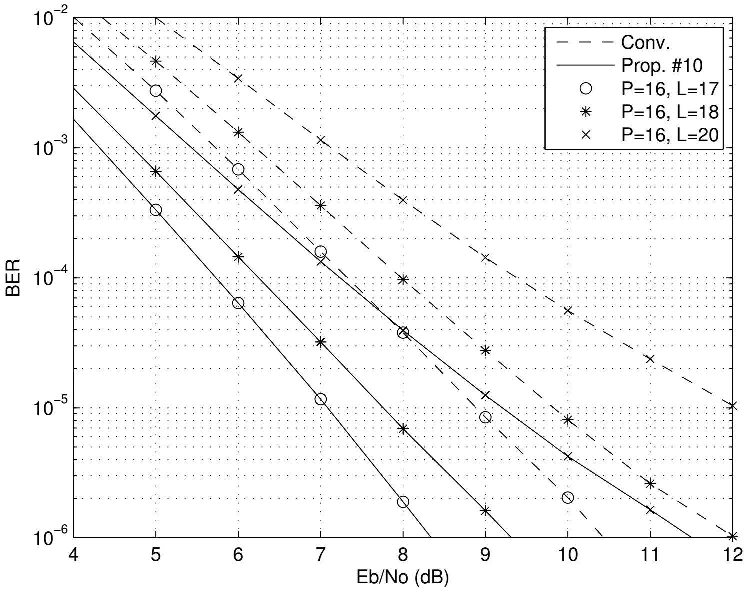


**Figure 3.12:** BER performance of the conventional algorithm (conv.) and the proposed reduced-complexity algorithm with 10 iterations (prop. #10) over ISI channels with $L = 17, 18, 20$ and power delay profile $q_l = e^{-0.1l}$, $l = 0, ..., L-1$.

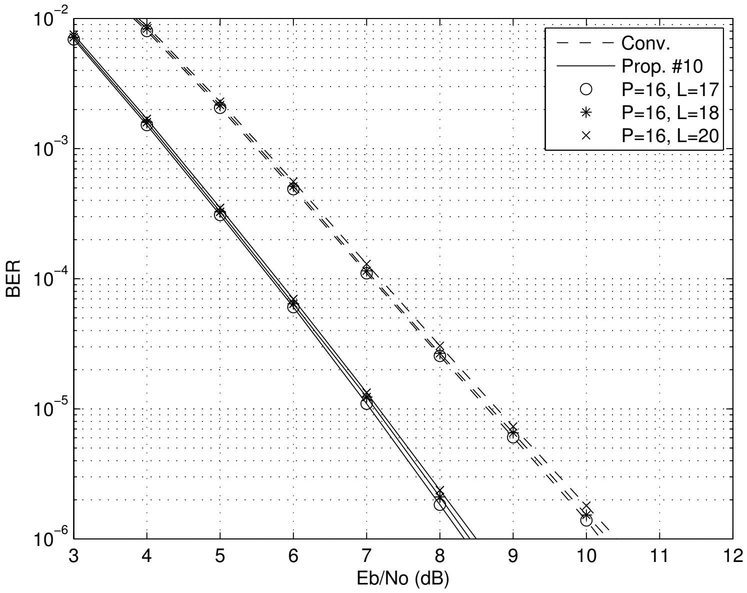


**Figure 3.13:** BER performance of the conventional algorithm (conv.) and the proposed reduced-complexity algorithm with 10 iterations (prop. #10) over ISI channels with $L = 17, 18, 20$ and power delay profile $q_l = e^{-0.3l}$, $l = 0, ..., L-1$.

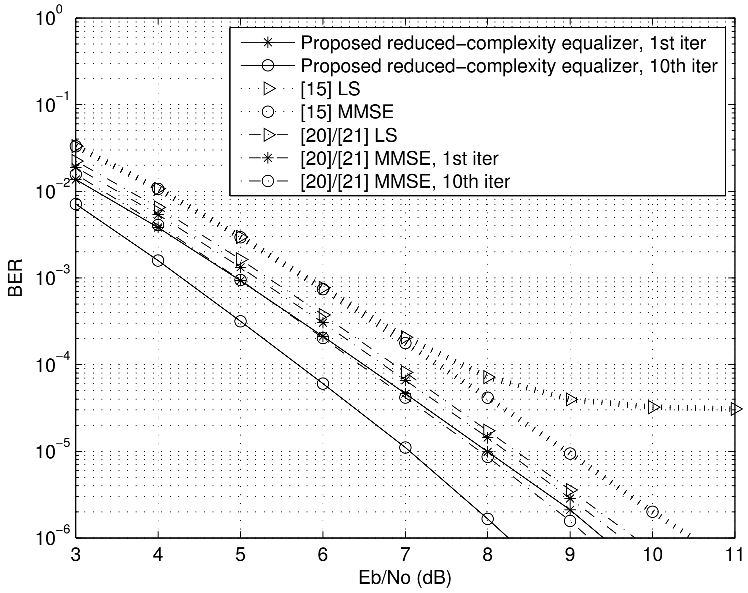


**Figure 3.14:** Performance comparison between the proposed reduced-complexity algorithm and some previous works over the random 17-tap channel.

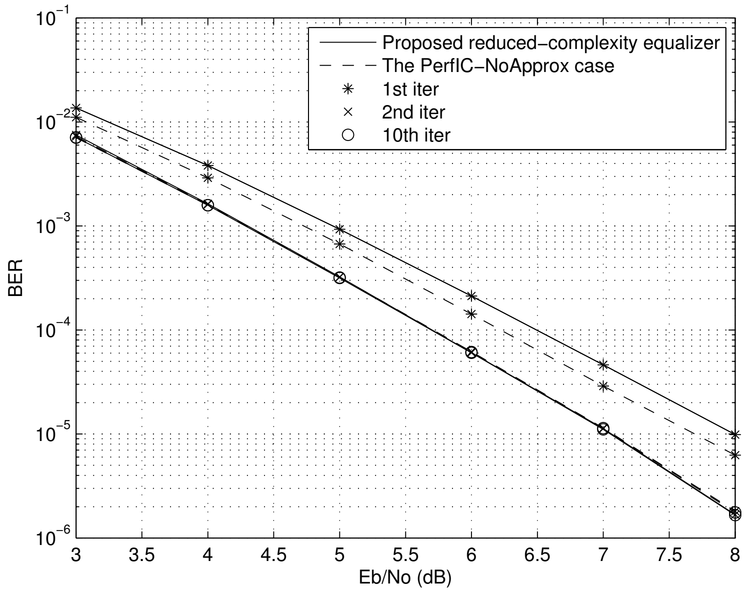


**Figure 3.15:** Performance comparison between the proposed reduced-complexity algorithm and the *PerfIC-NoApprox* case over the random 17-tap channel.

and MMSE methods in [55] and [56] with that of our proposed reduced-complexity algorithm, where perfect interference cancellation is carried out using the true data for [55] and [56], but not for our proposed algorithm. As we can see, over the random 17-tap channel, the LS method in [55] and [56] has similar performance as the first iteration of their MMSE method. With iterations, the MMSE method in [55] and [56] improves, but its convergence performance can only reach that of the first iteration of our proposed algorithm, and there is a gap of around 1dB between their convergence performance and ours at the BER of $10^{-5}$.

Note that the performance improvement of the proposed algorithm is achieved at the cost of higher latency, which is inherent in a turbo receiver as discussed in Section 3.2.3.

### 3.4.4 Performance Comparison with an Ideal Case

This set of simulations are to evaluate the performance loss due to the use of approximations (3.32) and (3.42). To this end, we compare the proposed reduced-complexity algorithm with an ideal case, the *PerfIC-NoApprox* case, where the following two hypotheses are employed.

1. Perfect interference cancellation is performed, i.e. instead of using (3.31) and (3.32), $\overrightarrow{\mathbf{m}}_{\mathbf{v}_k}$ and $\overrightarrow{\mathbf{V}}_{\mathbf{v}_k}$ are calculated as follows.

$$\overrightarrow{\mathbf{m}}_{\mathbf{v}_k} = \mathbf{C}\mathbf{F}_G^H\tilde{\mathbf{H}}_k\mathbf{B}\mathbf{u}_{k-1} + \mathbf{C}^H\mathbf{F}_G^H\tilde{\mathbf{H}}_k\mathbf{B}\mathbf{u}_{k+1}, \tag{3.57}$$

$$\overrightarrow{\mathbf{V}}_{\mathbf{v}_k} = 2\sigma^2\mathbf{I}_G, \tag{3.58}$$

   where the true data $\mathbf{u}_k$ is assumed to be perfectly known. As a result, each block is interference free from its adjacent blocks; there is no cycle in the graph; and there is no approximation to $\overrightarrow{\mathbf{V}}_{\mathbf{v}_k}$.

2. The approximation of (3.42) is disabled, i.e. the exact algorithm (i.e. Parts A, B and C1 in Table 3.1) is executed.

**Table 3.3:** THE CONVERGED VALUES $\bar{v}_a^c$ AND THE NOISE VARIANCES $\beta$.

| $E_b/N_0$ | 0dB | 1dB | 2dB | 3dB | 4dB | 5dB | 6dB |
|---|---|---|---|---|---|---|---|
| $\bar{v}_a^c$ | 0.2692 | 0.1084 | 0.0610 | 0.0516 | 0.0502 | 0.0500 | 0.0500 |
| $\beta$ | 1.0010 | 0.7951 | 0.6316 | 0.5017 | 0.3985 | 0.3165 | 0.2514 |

Fig. 3.15 shows that the difference between the *PerfIC-NoApprox* case and the proposed reduced-complexity algorithm only exists in the first iteration, and that the convergence performance of these two cases is the same.

### 3.4.5 Performance with Different Initialization of $\overrightarrow{\mathbf{V}}_{a_k}$

In Section 3.2.1, we have made the initialization of $\overrightarrow{\mathbf{V}}_{a_k} = \mathbf{I}$ at the first iteration. This set of simulations are to investigate the impact of different initialization of $\overrightarrow{\mathbf{V}}_{a_k}$ on the performance of the proposed reduced-complexity algorithm.

We define the average *a priori* variance of a data symbol by $\bar{v}_a = \frac{1}{KN}\sum_{k=0}^{K-1}\sum_{i=0}^{N-1} v_{a_{k,i}}$ where $v_{a_{k,i}}$ is the $i$-th diagonal entry of $\overrightarrow{\mathbf{V}}_{a_k}$, $N$ is the number of data symbols in a block and $K$ is the number of blocks in a data sequence. For a particular $E_b/N_0$ and a particular iteration, $\bar{v}_a$ is averaged over all the simulated sequences.

Fig. 3.16 shows that with $\overrightarrow{\mathbf{V}}_{a_k} = \mathbf{I}$ at the first iteration, $\bar{v}_a$ converges at the second iteration with lower value for higher $E_b/N_0$. The converged values $\bar{v}_a^c$ are listed in Table 3.3 together with the noise variances $\beta$. Fig. 3.17 compares the BER performance of the proposed reduced-complexity algorithm over the AWGN channel with different initializations of $\overrightarrow{\mathbf{V}}_{a_k}$, i.e. $\overrightarrow{\mathbf{V}}_{a_k} = \mathbf{I}$, $\overrightarrow{\mathbf{V}}_{a_k} = \bar{v}_a^c\mathbf{I}$, $\overrightarrow{\mathbf{V}}_{a_k} = \beta\mathbf{I}$, and $\overrightarrow{\mathbf{V}}_{a_k} = 100\mathbf{I}$. We can see that except for the case with $\overrightarrow{\mathbf{V}}_{a_k} = \bar{v}_a^c\mathbf{I}$, there is slight difference in the BER performance at the first iteration, and at the second iteration, four cases converge to the same BER curve.

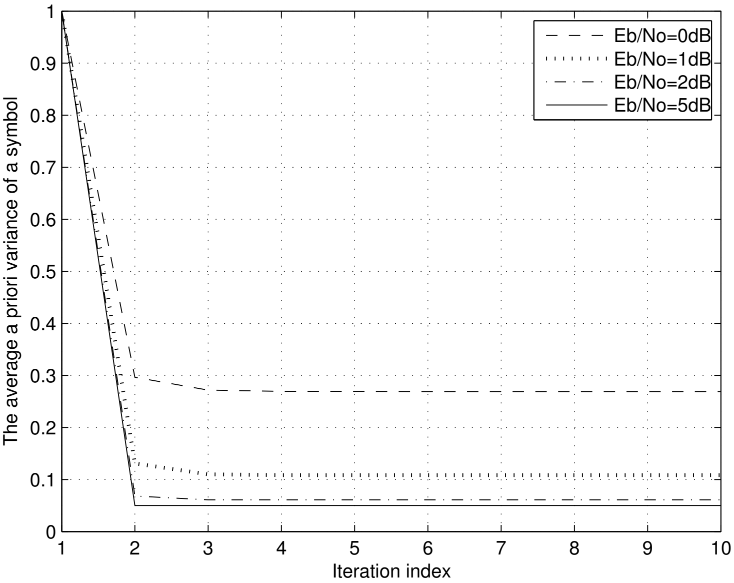


**Figure 3.16:** The average *a priori* variance of a data symbol $\bar{v}_a$ for the proposed reduced-complexity algorithm over the AWGN channel with $\overrightarrow{\mathbf{V}}_{a_k} = \mathbf{I}$ at the first iteration.

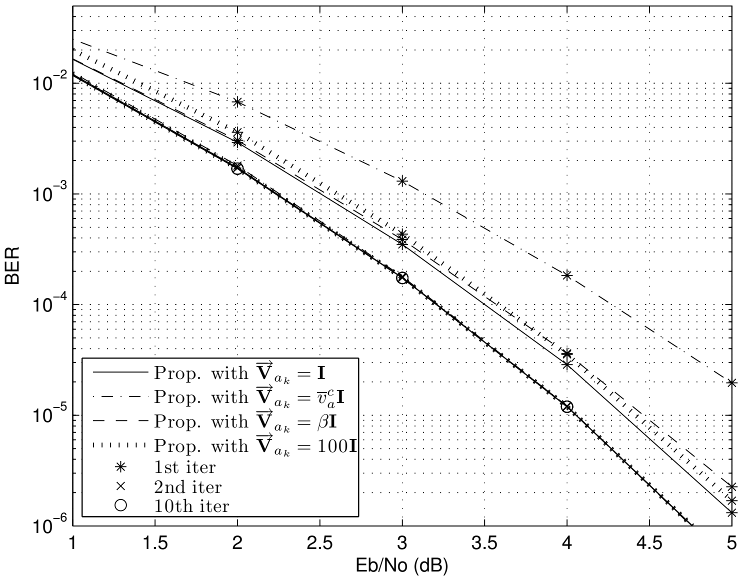


**Figure 3.17:** BER performance of the proposed reduced-complexity algorithm over the AWGN channel with different initializations of $\overrightarrow{\mathbf{V}}_{a_k}$.

## 3.5 Summary

We have proposed a factor graph approach to exploiting the CP for equalization in OFDM systems. The models of both the CP part and non-CP part of the received signal are presented in an FFG, and an equalization algorithm is developed based on the computation rules of the FFG and the GMP technique. Proper approximation is made to reduce the overall complexity and SNR analysis is carried out to justify the improvement. Simulation results show that significant gains are achieved by using the proposed algorithm and the turbo equalization system converges within two iterations.

# Chapter 4

# Exploiting Cyclic Prefix in OFDM Systems With Unknown Channel

The quality of the channel estimation has great impact on the OFDM system performance. To this end, pilots can be inserted into the data sequence to estimate the channel at the expense of overall system throughput. Moreover, as in [78–80], the soft data information fed back from the SISO decoder can be used to facilitate the channel estimation based on the turbo principle [8]. On the other hand, when the channel is perfectly known at the receiver, iterations between decoder and equalizer (i.e. turbo processing) is not needed for conventional OFDM systems, since different sub-carriers are independent and thereby turbo processing can not achieve any performance gain. Recently, we have shown in Chapter 3 that when CP is exploited for equalization, the iterations between decoder and equalizer are necessary for taking full advantage of the CP, since the use of CP in equalization makes different sub-carriers in OFDM dependent. However, channel estimation is not considered in Chapter 3.

In this chapter, we consider joint channel estimation, data detection and decoding in coded OFDM systems. Detection and decoding are performed iteratively between an equalizer and a SISO decoder based on the turbo principle. Inside the equalizer, the EM algorithm is employed via message passing to exploit the CP observation for joint detection and channel estimation. First, models for the CP observation, non-CP observation and the time correlation of the time-varying channel are each presented in an FFG, and message passing process is described for each graph. Rather than only using the means of channel estimates as in [34], we modify the system noise models to accommodate both the means and variances of channel estimates. Then when the three FFGs are combined, a larger EM algorithm is formed, and different uses of the CP and scheduling schemes for message passing between graphs are discussed. A scheduling scheme is proposed as an effective trade-off between the system performance and complexity. Simulation results show that the proposed approach effectively achieves channel estimation and data detection with a BER performance that is close to the case where the channel is perfectly known. The modification in system noise models improves the BER performance at early iterations, and the proposed scheduling scheme achieves good BER performance with lower complexity.

In comparison to previous works, the contributions of this chapter are listed as follows.

1. [64] and [65] do not take the coding information into account whereas this chapter considers joint decoding, detection and channel estimation; [62] does not make use of the time correlation of the time-varying channel whereas this chapter does.
2. [62] and [63] use CP in the channel estimator but not in the equalizer (i.e. the equalizer is still the one-tap equalizer) whereas this chapter compares four different uses of the CP, i.e. using CP for data detection/channel estimation or neither or both, and shows that using CP for data detection is more preferable

in terms of system BER performance and complexity.

3. [62–65] requires cubic complexity for processing one block in each inner iteration since full matrix inversion is involved, whereas this chapter proposes an effective trade-off scheduling scheme between graphs, which only requires $\mathcal{O}(NlogN)$ per data block per outer iteration (see Section 4.3.2).
4. [34, 81] and [72] only consider the means of channel estimates in the system noise models, whereas this chapter also accommodates the variances of channel estimates and improves the system performance at early iterations.

## 4.1 System Model

Consider a turbo OFDM system as depicted in Fig. 4.1. The information bits are encoded, interleaved and mapped onto a symbol sequence $\mathbf{u} = [u_0, ..., u_{Z-1}]^T$, where $Z$ is the length of the sequence. Assume that $Z = KN$ and each sequence consists of $K$ blocks, each of length $N$. The $k$th block, $k = 0, ..., K-1$, is denoted by $\mathbf{u}_k = [u_{kN}, ..., u_{(k+1)N-1}]^T$. An $N$-point IFFT is applied on $\mathbf{u}_k$, producing $\mathbf{x}_k = \mathbf{F}_N^H \mathbf{u}_k$, and the CP is formed by duplicating the last $P$ samples $\tilde{\mathbf{x}}_k$ in front of $\mathbf{x}_k$, where $\tilde{\mathbf{x}}_k = \mathbf{A}\mathbf{x}_k$, and $\mathbf{A} = \left[ \begin{array}{cc} \mathbf{0}_{P\times(N-P)} & \mathbf{I}_{P\times P} \end{array} \right]$. For the CP to be effective, we have $P \geq L-1$ where $L$ is the maximum delay spread of the multipath channel. After prefixing, the $k$th block is extended to $\mathbf{s}_k = [\tilde{\mathbf{x}}_k^T, \mathbf{x}_k^T]^T = [x_{(k+1)N-P}, ..., x_{(k+1)N-1}, x_{kN}, ..., x_{(k+1)N-1}]^T$ of length $M$, where $M = N + P$. The sequence $\mathbf{s} = [\mathbf{s}_0^T, ..., \mathbf{s}_{K-1}^T]^T = [s_0, ..., s_{J-1}]^T$, where $J = KM$, is then transmitted over the channel, and $\mathbf{r} = [r_0, r_1, ..., r_{V-1}]^T$ is observed at the receiver, where $V = J + L - 1$, and

$$r_j = \sum_{l=0}^{L-1} h_{j,l} \cdot s_{j-l} + n_j, \quad j = 0, ..., V-1 \tag{4.1}$$

where $\{h_{j,l},\ l = 0, ..., L-1\}$ is the channel impulse response (CIR) at time $j$, and $\{n_j\}$ is the AWGN noise with mean zero and variance $2\sigma^2$.

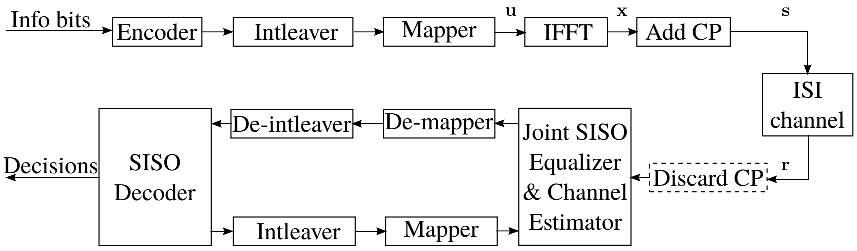


**Figure 4.1:** Block diagram of a coded OFDM system.

We adopt the wide sense stationary uncorrelated scattering (WSSUS) model [82] for the multipath time-varying channel, where different taps of the channel are assumed to be independent of each other. The power delay profile is given by $\boldsymbol{\eta} = \left[\eta_0, ..., \eta_{L-1}\right]^T$, where $\eta_l$ is the power of the $l$th tap, $l = 0, ..., L-1$. To characterize the time variation of the channel, we assume that the channel remains the same within a block but may vary in the next block [83], and we use the following first-order auto-regressive (AR) model to approximate the channel,

$$\mathbf{h}_k = \mathbf{R}\mathbf{h}_{k-1} + \boldsymbol{\delta}_{k-1} \tag{4.2}$$

where $\mathbf{h}_k = \left[h_k^0, ..., h_k^{L-1}\right]^T$ is the channel that $\mathbf{s}_k$ undergoes, $\mathbf{R} = \rho\mathbf{I}_{L\times L}$ is the correlation function of the channel, with $\rho = J_0(2\pi f_D T_s)$ and $f_D$ the maximum Doppler frequency, $T_s$ the time duration of a block and $J_0(\cdot)$ the zeroth order Bessel function of the first kind, and $\boldsymbol{\delta}_{k-1}$ is Gaussian noise with mean $\mathbf{0}$ and covariance $(1-\rho^2)Diag(\boldsymbol{\eta})$.

The counterpart of the block $\mathbf{s}_k$ at the receiver is $\mathbf{r}_k = [r_{kM}, ..., r_{(k+1)M+L-2}]^T$ of length $M+L-1$. The block $\mathbf{r}_k$ can be partitioned into two parts, $\mathbf{y}_k = [r_{kM+P}, ..., r_{(k+1)M-1}]^T$ of length $N$, and $\tilde{\mathbf{y}}_k = [r_{kM}, ..., r_{kM+P-1}, r_{(k+1)M}, ..., r_{(k+1)M+L-2}]^T$ of length $G$, where $G = P+L-1$. In conventional OFDM systems, the CP observation is usually discarded at the receiver, and only the part $\mathbf{y}_k$ is processed. We model this part as

$$\mathbf{y}_k = \mathbf{x}_k \circledast \mathbf{h}_k + \mathbf{n}_k, \quad k = 0, ..., K-1 \tag{4.3}$$

where "$\circledast$" denotes the circular convolution and $\mathbf{n}_k$ is the AWGN vector with mean $\mathbf{0}$ and covariance matrix $2\sigma^2\mathbf{I}$.

The other part $\tilde{\mathbf{y}}_k$ consists of the CP observation of the $k$th block and the first $L-1$ received symbols of the $(k+1)$th block. It also contains the interference from the previous block $\mathbf{s}_{k-1}$ and the next block $\mathbf{s}_{k+1}$, which we denote by $\mathbf{w}_{k-1}^{tail}$ and $\mathbf{w}_{k+1}^{head}$, respectively. The part $\tilde{\mathbf{y}}_k$ is modeled as

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{x}}_k * \mathbf{h}_k + \mathbf{w}_{k-1}^{tail} + \mathbf{w}_{k+1}^{head} + \tilde{\mathbf{n}}_k, \quad k = 0, ..., K-1 \tag{4.4}$$

where "$*$" denotes the linear convolution, $\tilde{\mathbf{n}}_k$ denotes the AWGN vector of length $G$, and for $k=0$, $\mathbf{w}_{-1}^{tail} = \mathbf{0}$ and for $k = K-1$, $\mathbf{w}_{K}^{head} = \mathbf{0}$. For notational simplicity, we put the last three terms of (4.4) together as

$$\mathbf{v}_k = \mathbf{w}_{k-1}^{tail} + \mathbf{w}_{k+1}^{head} + \tilde{\mathbf{n}}_k, \tag{4.5}$$

as a result, (4.4) becomes

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{x}}_k * \mathbf{h}_k + \mathbf{v}_k. \tag{4.6}$$

Transforming (4.3) and (4.6) from time domain to frequency domain and substituting $\mathbf{x}_k = \mathbf{F}_N^H \mathbf{u}_k$ and $\tilde{\mathbf{x}}_k = \mathbf{A}\mathbf{x}_k$ in, we obtain

$$\mathbf{y}_k = \mathbf{F}_N^H \Big[\mathbf{u}_k \odot \big(\sqrt{N}\mathbf{F}_N \hat{\mathbf{S}} \mathbf{h}_k\big)\Big] + \mathbf{n}_k, \tag{4.7}$$

$$\tilde{\mathbf{y}}_k = \mathbf{F}_G^H \Big[\big(\mathbf{B}\mathbf{u}_k\big) \odot \big(\sqrt{G}\mathbf{F}_G \check{\mathbf{S}} \mathbf{h}_k\big)\Big] + \mathbf{v}_k \tag{4.8}$$

where "$\odot$" is point-wise multiplication, $\mathbf{B} = \mathbf{F}_G \mathbf{Q} \mathbf{A} \mathbf{F}_N^H$ with $\mathbf{Q} = \begin{bmatrix} \mathbf{I}_{P\times P} & \mathbf{0}_{P\times(L-1)} \end{bmatrix}^T$, and $\hat{\mathbf{S}} = \begin{bmatrix} \mathbf{I}_{L\times L} & \mathbf{0}_{L\times(N-L)} \end{bmatrix}^T$, $\check{\mathbf{S}} = \begin{bmatrix} \mathbf{I}_{L\times L} & \mathbf{0}_{L\times(G-L)} \end{bmatrix}^T$ are to adapt $\mathbf{h}_k$ for DFT operation. To write $\mathbf{v}_k$ in a similar form, we have

$$\mathbf{v}_k = \tilde{\mathbf{n}}_k + \mathbf{C}\mathbf{F}_G^H\Big[\big(\mathbf{B}\mathbf{u}_{k-1}\big) \odot \big(\sqrt{G}\mathbf{F}_G\check{\mathbf{S}}\mathbf{h}_{k-1}\big)\Big] \\ + \mathbf{C}^H\mathbf{F}_G^H\Big[\big(\mathbf{B}\mathbf{u}_{k+1}\big) \odot \big(\sqrt{G}\mathbf{F}_G\check{\mathbf{S}}\mathbf{h}_{k+1}\big)\Big] \tag{4.9}$$

where matrix $\mathbf{C}$ and its Hermitian transpose $\mathbf{C}^H$ are used to obtain the interference terms $\mathbf{w}_{k-1}^{tail}$ and $\mathbf{w}_{k+1}^{head}$, respectively, and

$$\mathbf{C} = \begin{bmatrix} \mathbf{0}_{(L-1)\times P} & \mathbf{I}_{(L-1)\times(L-1)} \\ \mathbf{0}_{P\times P} & \mathbf{0}_{P\times(L-1)} \end{bmatrix}. \tag{4.10}$$

Let $\mathbf{d}_k = \mathbf{u}_k \odot (\sqrt{N}\mathbf{F}_N\hat{\mathbf{S}}\mathbf{h}_k)$ and $\tilde{\mathbf{d}}_k = (\mathbf{B}\mathbf{u}_k) \odot (\sqrt{G}\mathbf{F}_G\check{\mathbf{S}}\mathbf{h}_k)$. Assuming that the lengths of $\mathbf{d}_k$ and $\tilde{\mathbf{d}}_k$ are integral multiples of $L$, i.e. $N = TL$ and $G = SL$, we can divide $\mathbf{d}_k$ and $\tilde{\mathbf{d}}_k$ into $T$ and $S$ groups, respectively, to obtain multiple independent estimates of the channel and then combine them together. The $z$th group of $\mathbf{d}_k$ is given by

$$\mathbf{d}_k^z = \mathbf{b}_k^z \odot \mathbf{c}_k^z, \quad z = 0, ..., T-1 \tag{4.11}$$

where

$$\mathbf{d}_k^z = \big(\mathbf{d}_k\big)_{\{z,z+T,...,z+(L-1)T\}}, \tag{4.12}$$

$$\mathbf{b}_k^z = \big(\mathbf{u}_k\big)_{\{z,z+T,...,z+(L-1)T\}}, \tag{4.13}$$

$$\mathbf{c}_k^z = \big(\sqrt{N}\mathbf{F}_N\hat{\mathbf{S}}\mathbf{h}_k\big)_{\{z,z+T,...,z+(L-1)T\}}, \tag{4.14}$$

and the $z$th group of $\tilde{\mathbf{d}}_k$ is given by

$$\tilde{\mathbf{d}}_k^z = \tilde{\mathbf{b}}_k^z \odot \tilde{\mathbf{c}}_k^z, \quad z = 0, ..., S-1 \tag{4.15}$$

where

$$\tilde{\mathbf{d}}_k^z = \big(\tilde{\mathbf{d}}_k\big)_{\{z,z+S,...,z+(L-1)S\}}, \tag{4.16}$$

$$\tilde{\mathbf{b}}_k^z = \big(\mathbf{B}\mathbf{u}_k\big)_{\{z,z+S,...,z+(L-1)S\}}, \tag{4.17}$$

$$\tilde{\mathbf{c}}_k^z = \big(\sqrt{G}\mathbf{F}_G\check{\mathbf{S}}\mathbf{h}_k\big)_{\{z,z+S,...,z+(L-1)S\}}. \tag{4.18}$$

As in [83], we rewrite $\mathbf{c}_k^z$ in (4.14) and $\tilde{\mathbf{c}}_k^z$ in (4.18) respectively as

$$\mathbf{c}_k^z = \sqrt{L}\mathbf{F}_L Diag\big(1, e^{-j2\pi z/N}, ..., e^{-j2\pi(L-1)z/N}\big)\mathbf{h}_k, \quad z = 0, ..., T-1, \tag{4.19}$$

$$\tilde{\mathbf{c}}_k^z = \sqrt{L}\mathbf{F}_L Diag\big(1, e^{-j2\pi z/G}, ..., e^{-j2\pi(L-1)z/G}\big)\mathbf{h}_k, \quad z = 0, ..., S-1. \tag{4.20}$$

Moreover, we have the relations between $\mathbf{c}_k^z$ ($\tilde{\mathbf{c}}_k^z$) and $\mathbf{c}_k^{z-1}$ ($\tilde{\mathbf{c}}_k^{z-1}$) as

$$\mathbf{c}_k^{z-1} = \mathbf{D}\mathbf{c}_k^z, \quad z = 1, ..., T-1, \tag{4.21}$$

$$\tilde{\mathbf{c}}_k^{z-1} = \mathbf{E}\tilde{\mathbf{c}}_k^z, \quad z = 1, ..., S-1, \tag{4.22}$$

where

$$\mathbf{D} = \mathbf{F}_L Diag\big(1, e^{j2\pi/N}, ..., e^{j2\pi(L-1)/N}\big)\mathbf{F}_L^H, \tag{4.23}$$

$$\mathbf{E} = \mathbf{F}_L Diag\big(1, e^{j2\pi/G}, ..., e^{j2\pi(L-1)/G}\big)\mathbf{F}_L^H, \tag{4.24}$$

and specially when $z = 0$, we have

$$\mathbf{c}_k^0 = \tilde{\mathbf{c}}_k^0 = \sqrt{L}\mathbf{F}_L\mathbf{h}_k. \tag{4.25}$$

## 4.2 EM Algorithm and Message Passing

The EM algorithm [84] provides a way of solving the maximum *a posteriori* (MAP) estimation problem with tractable complexity. The MAP channel estimation problem in this chapter is formulated as

$$\left\{\hat{\mathbf{h}}_0, ..., \hat{\mathbf{h}}_{K-1}\right\} = \arg\max_{\mathbf{h}_0,...,\mathbf{h}_{K-1}} \log p(\mathbf{h}_0, ..., \mathbf{h}_{K-1}|\mathbf{y}_k, \tilde{\mathbf{y}}_k, k = 0, ..., K-1). \tag{4.26}$$

Assuming that there is no correlation between adjacent blocks after inter-block interference cancellation, (4.26) can be decoupled into $K$ independent maximization problems, i.e. for $k = 0, ..., K-1$,

$$\begin{aligned}\hat{\mathbf{h}}_k &= \arg\max_{\mathbf{h}_k} \log p(\mathbf{h}_k|\mathbf{y}_k, \tilde{\mathbf{y}}_k)\\ &= \arg\max_{\mathbf{h}_k} \log \sum_{\mathbf{u}_k} p(\mathbf{h}_k|\mathbf{y}_k, \tilde{\mathbf{y}}_k, \mathbf{u}_k) p(\mathbf{u}_k).\end{aligned} \tag{4.27}$$

Direct calculation of (4.27) requires enumeration over all the possible values of hidden variable $\mathbf{u}_k$ and a prohibitively high computational complexity is induced. The EM algorithm, however, can solve (4.27) asymptotically by iterating the following two steps [84]:

$$\begin{aligned}\text{E-step: } &\text{compute } Q(\mathbf{h}_k|\hat{\mathbf{h}}_k^{p-1})\\ &= E_{\mathbf{u}_k}\left\{ \log p(\mathbf{y}_k, \tilde{\mathbf{y}}_k|\mathbf{h}_k, \mathbf{u}_k) \Big| \mathbf{y}_k, \tilde{\mathbf{y}}_k, \hat{\mathbf{h}}_k^{p-1}\right\}\end{aligned} \tag{4.28}$$

$$\begin{aligned}\text{M-step: } &\text{solve } \hat{\mathbf{h}}_k^{p}\\ &= \arg\max_{\mathbf{h}_k} \left\{ Q(\mathbf{h}_k|\hat{\mathbf{h}}_k^{p-1}) + \log p(\mathbf{h}_k)\right\}\end{aligned} \tag{4.29}$$

where $\hat{\mathbf{h}}_k^{p-1}$ is the channel estimate of the $(p-1)$th EM iteration, and $E_B\{A|C\}$ denotes the expectation of function $A$ with respect to variable $B$ given condition $C$, and $p(\mathbf{h}_k)$ represents the *a priori* probability of $\mathbf{h}_k$ obtained from other blocks. One of the properties of the EM algorithm is that the log-likelihood function in (4.27) always increases with iteration, i.e. $\log p(\hat{\mathbf{h}}_k^{p}|\mathbf{y}_k, \tilde{\mathbf{y}}_k) \geq \log p(\hat{\mathbf{h}}_k^{p-1}|\mathbf{y}_k, \tilde{\mathbf{y}}_k)$, and hence, convergence is guaranteed (although not necessarily to the global maxima) [84].

In [34], the EM algorithm has been viewed as Gaussian message passing in factor graphs for linear Gaussian models with unknown coefficients. In this chapter, we adopt the notations in [34] for a Forney-style factor graph (FFG) and employ the computation rules in both [34] and [29]. These rules are for real vectors but can be extended for complex vectors [81].

### 4.2.1 Factor Graph for the $\mathbf{y}_k$ Part of Observation

According to models (4.7), (4.11), (4.12), (4.13), (4.14), (4.21) and (4.25), we can draw a factor graph for the $\mathbf{y}_k$ part of observation as depicted in Fig. 4.2 (Subgraph 1).

To prepare for the message passing EM algorithm, we first use rules (II.10) and (II.8) in [29] to obtain the backward message of $\mathbf{e}_k$ as

$$\overleftarrow{\mathbf{m}}_{\mathbf{e}_k} = \mathbf{y}_k, \quad \overleftarrow{\mathbf{V}}_{\mathbf{e}_k} = \beta \mathbf{I}_N \tag{4.30}$$

where $\beta = 2\sigma^2$ is the variance of the AWGN noise. Then using rules (III.6) and (III.5) in [29], we get

$$\overleftarrow{\mathbf{m}}_{\mathbf{d}_k} = \mathbf{F}_N \mathbf{y}_k, \quad \overleftarrow{\mathbf{V}}_{\mathbf{d}_k} = \beta \mathbf{I}_N. \tag{4.31}$$

Reversing through the Aggregation node and going through the Partition node, we obtain the backward message of $\mathbf{d}_k^z$ and the forward message of $\mathbf{b}_k^z$, $z = 0, ..., T-1$, respectively, as

$$\overleftarrow{\mathbf{m}}_{\mathbf{d}_k^z} = \left(\overleftarrow{\mathbf{m}}_{\mathbf{d}_k}\right)_{\{z, z+T, ..., z+(L-1)T\}}, \tag{4.32}$$

$$\overleftarrow{\mathbf{V}}_{\mathbf{d}_k^z} = \left(\overleftarrow{\mathbf{V}}_{\mathbf{d}_k}\right)_{\{z, z+T, ..., z+(L-1)T\}}, \tag{4.33}$$

and

$$\overrightarrow{\mathbf{m}}_{\mathbf{b}_k^z} = \left(\overrightarrow{\mathbf{m}}_{\mathbf{b}_k}\right)_{\{z, z+T, ..., z+(L-1)T\}}, \tag{4.34}$$

$$\overrightarrow{\mathbf{V}}_{\mathbf{b}_k^z} = \left(\overrightarrow{\mathbf{V}}_{\mathbf{b}_k}\right)_{\{z, z+T, ..., z+(L-1)T\}}. \tag{4.35}$$

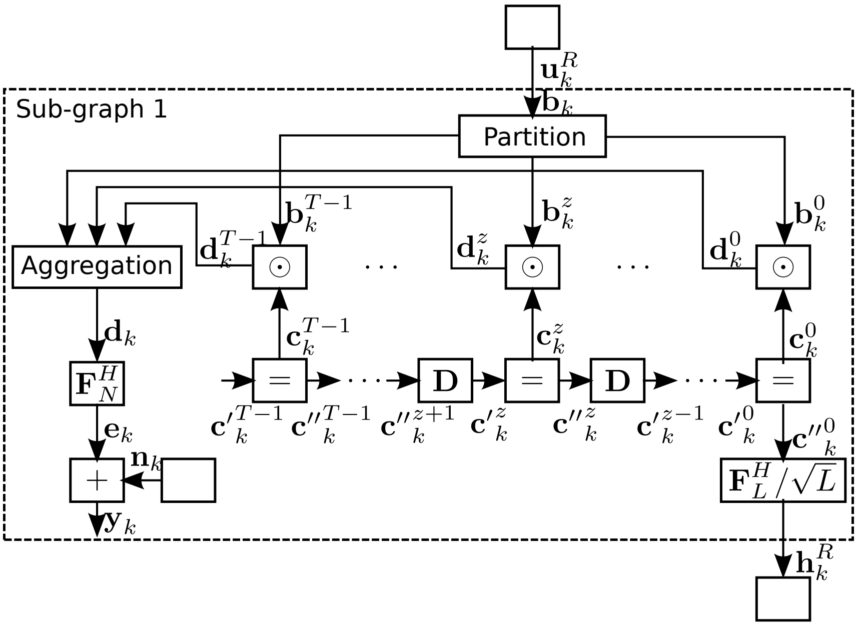


**Figure 4.2:** The factor graph for the $\mathbf{y}_k$ part of observation, where the Partition node separates $\mathbf{b}_k$ into sub-vectors $\mathbf{b}_k^z$, $z = 0, ..., T-1$, and the Aggregation node groups $\mathbf{d}_k^z$, $z = 0, ..., T-1$, into one vector $\mathbf{d}_k$.

Using rules (II.1)-(II.4) in [34], the EM algorithm for Sub-graph 1 can be summarized as in Table 4.1, where the E-step calculates the *a posteriori* message of $\mathbf{b}_k$ using the mean and covariance of $\mathbf{d}_k$ and channel estimate $\left[\mathbf{h}_k^R\right]^{new}$, and the M-step calculates the forward message of $\mathbf{h}_k^R$. Note that Sub-graph 1 represents the relation between the data and the channel based on the non-CP observation $\mathbf{y}_k$, and the E-step of Sub-graph 1 (Table 4.1.A) achieves data detection based on $\mathbf{y}_k$ and the M-step of Sub-graph 1 (Table 4.1.B) achieves channel estimation based on $\mathbf{y}_k$.

### 4.2.2 Factor Graph for the $\tilde{\mathbf{y}}_k$ Part of Observation

According to models (4.8), (4.15), (4.16), (4.17), (4.18), (4.22) and (4.25), we can draw a factor graph for the $\tilde{\mathbf{y}}_k$ part of observation as shown in Fig. 4.3 (Sub-graph 2).

**Table 4.1:** Message passing for the EM algorithm in Sub-graph 1

```
Inputs:   $\overrightarrow{\mathbf{m}}_{\mathbf{b}_k},\overrightarrow{\mathbf{V}}_{\mathbf{b}_k};\ \overleftarrow{\mathbf{m}}_{\mathbf{d}_k},\overleftarrow{\mathbf{V}}_{\mathbf{d}_k};\ \left[\mathbf{h}_k^R\right]^{new}$
Outputs:  $\mathbf{m}_{\mathbf{b}_k},\mathbf{V}_{\mathbf{b}_k};\ \overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R},\overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R}$

A. E-step of Sub-graph 1:
  1. initialize $\mathbf{c}_k^z$ using channel estimate $\left[\mathbf{h}_k^R\right]^{new}$, $z = 0, ..., T-1$
       $\mathbf{c}_k^0 = \sqrt{L}\mathbf{F}_L\left[\mathbf{h}_k^R\right]^{new}$
       $\mathbf{c}_k^z = \mathbf{D}^H\mathbf{c}_k^{z-1}, \quad z = 1, ..., T-1$
  2. calculate the a posteriori message of $\mathbf{b}_k^z$
       $\mathbf{V}_{\mathbf{b}_k^z} = \left[\overrightarrow{\mathbf{V}}_{\mathbf{b}_k^z}^{-1} + \beta^{-1}Diag\left(\mathbf{c}_k^{z*} \odot \mathbf{c}_k^z\right)\right]^{-1}$
       $\mathbf{m}_{\mathbf{b}_k^z} = \mathbf{V}_{\mathbf{b}_k^z}\left[\overrightarrow{\mathbf{V}}_{\mathbf{b}_k^z}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{b}_k^z} + \beta^{-1}\left(\mathbf{c}_k^{z*} \odot \overleftarrow{\mathbf{m}}_{\mathbf{d}_k^z}\right)\right]$
B. M-step of Sub-graph 1:
  1. calculate the backward message of $\mathbf{c}_k^z$ as
       $\overleftarrow{\mathbf{V}}_{\mathbf{c}_k^z} = \beta\left[\left(\mathbf{V}_{\mathbf{b}_k^z}\right)_{diag} + Diag\left(\mathbf{m}_{\mathbf{b}_k^z}^* \odot \mathbf{m}_{\mathbf{b}_k^z}\right)\right]^{-1}$
       $\overleftarrow{\mathbf{m}}_{\mathbf{c}_k^z} = \beta^{-1}\overleftarrow{\mathbf{V}}_{\mathbf{c}_k^z}\left(\mathbf{m}_{\mathbf{b}_k^z}^* \odot \overleftarrow{\mathbf{m}}_{\mathbf{d}_k^z}\right)$
  2. set the forward message of $\mathbf{c''}_k^{T-1}$ as
       $\overrightarrow{\mathbf{m}}_{\mathbf{c''}_k^{T-1}} = \overleftarrow{\mathbf{m}}_{\mathbf{c}_k^{T-1}}, \quad \overrightarrow{\mathbf{V}}_{\mathbf{c''}_k^{T-1}} = \overleftarrow{\mathbf{V}}_{\mathbf{c}_k^{T-1}}$
  3. for $z = T-1, ..., 1$ do
       a) $\overrightarrow{\mathbf{V}}_{\mathbf{c'}_k^{z-1}} = \mathbf{D}\overrightarrow{\mathbf{V}}_{\mathbf{c''}_k^z}\mathbf{D}^H, \quad \overrightarrow{\mathbf{m}}_{\mathbf{c'}_k^{z-1}} = \mathbf{D}\overrightarrow{\mathbf{m}}_{\mathbf{c''}_k^z}$
       b) $\overrightarrow{\mathbf{V}}_{\mathbf{c''}_k^{z-1}} = \left[\overrightarrow{\mathbf{V}}_{\mathbf{c'}_k^{z-1}}^{-1} + \overleftarrow{\mathbf{V}}_{\mathbf{c}_k^{z-1}}^{-1}\right]^{-1}$
          $\overrightarrow{\mathbf{m}}_{\mathbf{c''}_k^{z-1}} = \overrightarrow{\mathbf{V}}_{\mathbf{c''}_k^{z-1}}\left[\overrightarrow{\mathbf{V}}_{\mathbf{c'}_k^{z-1}}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{c'}_k^{z-1}} + \overleftarrow{\mathbf{V}}_{\mathbf{c}_k^{z-1}}^{-1}\overleftarrow{\mathbf{m}}_{\mathbf{c}_k^{z-1}}\right]$
     end for
  4. calculate the forward message of $\mathbf{h}_k^R$ as
       $\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R} = \frac{1}{\sqrt{L}}\mathbf{F}_L^H\overrightarrow{\mathbf{m}}_{\mathbf{c''}_k^0}, \ \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R} = \frac{1}{L}\mathbf{F}_L^H\overrightarrow{\mathbf{V}}_{\mathbf{c''}_k^0}\mathbf{F}_L$
```

From (4.9), we have the mean vector for $\mathbf{v}_k$ in Sub-graph 2 as

$$\overrightarrow{\mathbf{m}}_{\mathbf{v}_k} = \mathbf{C}\mathbf{F}_G^H \boldsymbol{\xi}_{k-1} + \mathbf{C}^H \mathbf{F}_G^H \boldsymbol{\xi}_{k+1} \tag{4.36}$$

where

$$\boldsymbol{\xi}_k = \big(\mathbf{B}\mathbf{m}_{\mathbf{u}_k}\big) \odot \big(\sqrt{G}\mathbf{F}_G \check{\mathbf{S}} \mathbf{m}_{\mathbf{h}_k}\big), \tag{4.37}$$

and $\mathbf{m}_{\mathbf{u}_k}$ and $\mathbf{m}_{\mathbf{h}_k}$ are the *a priori* means of $\mathbf{u}_k$ and $\mathbf{h}_k$. The covariance matrix of $\mathbf{v}_k$ is given by

$$\overrightarrow{\mathbf{V}}_{\mathbf{v}_k} = \lambda_k \mathbf{I}_G \tag{4.38}$$

where

$$\lambda_k = \beta + \frac{1}{G}\sum\nolimits_{l=0}^{L-1} \left[\gamma_{k+1}(L-1-l)\left|h_{k+1}^l\right|^2 + \gamma_{k-1} l \left|h_{k-1}^l\right|^2\right], \tag{4.39}$$

and

$$\gamma_k = \frac{1}{N}\sum\nolimits_{i=0}^{N-1} \big(\mathbf{V}_{\mathbf{u}_k}\big)_{i,i}, \qquad h_k^l = \big(\mathbf{m}_{\mathbf{h}_k}\big)_l, \tag{4.40}$$

and $\mathbf{V}_{\mathbf{u}_k}$ is the *a priori* covariance of $\mathbf{u}_k$, and $\mathbf{m}_{\mathbf{h}_k}$ is the *a priori* mean of $\mathbf{h}_k$. Then similar to (4.30)-(4.35), to prepare for the message passing in Sub-graph 2, we have the backward message of $\tilde{\mathbf{d}}_k$ as

$$\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k} = \mathbf{F}_G\big(\tilde{\mathbf{y}}_k - \overrightarrow{\mathbf{m}}_{\mathbf{v}_k}\big), \qquad \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{d}}_k} = \overrightarrow{\mathbf{V}}_{\mathbf{v}_k}, \tag{4.41}$$

and the $z$th group, $z = 0, ..., S-1$, of the backward message of $\mathbf{d}_k$ and the forward message of $\mathbf{b}_k$ is respectively given by

$$\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k^z} = \big(\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k}\big)_{\{z,z+S,\dots,z+(L-1)S\}}, \tag{4.42}$$

$$\overleftarrow{\mathbf{V}}_{\tilde{\mathbf{d}}_k^z} = \big(\overleftarrow{\mathbf{V}}_{\tilde{\mathbf{d}}_k}\big)_{\{z,z+S,\dots,z+(L-1)S\}}, \tag{4.43}$$

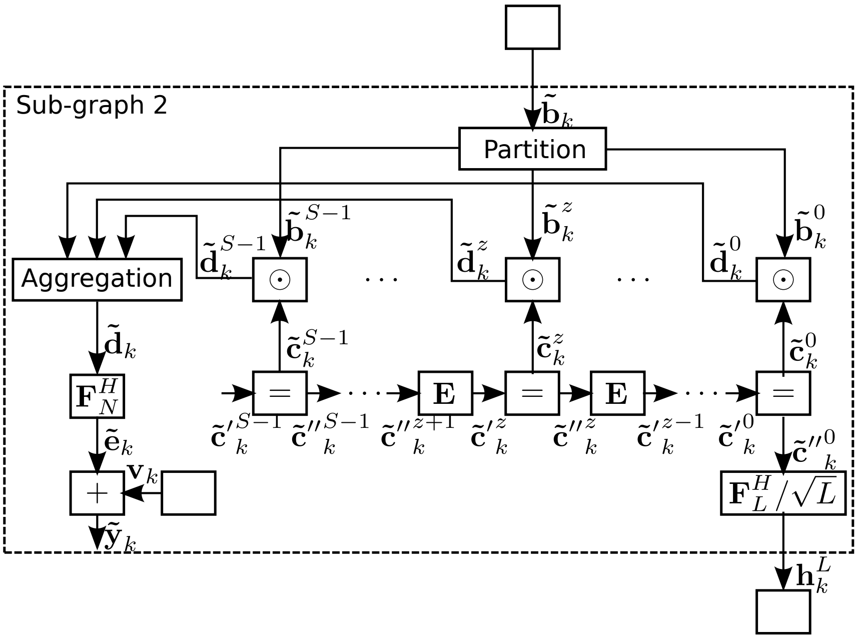

Sub-graph 2
$\tilde{\mathbf{b}}_k$
Partition
$\tilde{\mathbf{b}}_k^{S-1}$
$\tilde{\mathbf{b}}_k^{z}$
$\tilde{\mathbf{b}}_k^{0}$
Aggregation
$\tilde{\mathbf{d}}_k^{S-1}$
$\tilde{\mathbf{d}}_k^{z}$
$\tilde{\mathbf{d}}_k^{0}$
$\odot$
$\cdots$
$\tilde{\mathbf{d}}_k$
$\tilde{\mathbf{c}}_k^{S-1}$
$\tilde{\mathbf{c}}_k^{z}$
$\tilde{\mathbf{c}}_k^{0}$
$\mathbf{F}_N^H$
$=$
$\mathbf{E}$
$\tilde{\mathbf{c}}'^{S-1}_k$
$\tilde{\mathbf{c}}''^{S-1}_k$
$\tilde{\mathbf{c}}''^{z+1}_k$
$\tilde{\mathbf{c}}'^{z}_k$
$\tilde{\mathbf{c}}''^{z}_k$
$\tilde{\mathbf{c}}'^{z-1}_k$
$\tilde{\mathbf{c}}'^{0}_k$
$\tilde{\mathbf{c}}''^{0}_k$
$\tilde{\mathbf{e}}_k$
$\mathbf{v}_k$
$+$
$\mathbf{F}_L^H/\sqrt{L}$
$\tilde{\mathbf{y}}_k$
$\mathbf{h}_k^L$

**Figure 4.3:** The factor graph for the $\tilde{\mathbf{y}}_k$ part of observation, where the Partition node separates $\tilde{\mathbf{b}}_k$ into sub-vectors $\tilde{\mathbf{b}}_k^z$, $z = 0, ..., S-1$, and the Aggregation node groups $\tilde{\mathbf{d}}_k^z$, $z = 0, ..., S-1$, into one vector $\tilde{\mathbf{d}}_k$.

and

$$\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k^z} = \left(\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}\right)_{\{z,z+S,...,z+(L-1)S\}}, \tag{4.44}$$

$$\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k^z} = \left(\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)_{\{z,z+S,...,z+(L-1)S\}}. \tag{4.45}$$

The message passing EM algorithm for Sub-graph 2 is listed as in Table 4.2, where the E-step calculates the *a posteriori* message of $\tilde{\mathbf{b}}_k$ using the mean and covariance of $\tilde{\mathbf{d}}_k$ and channel estimate $\left[\mathbf{h}_k^L\right]^{new}$, and the M-step calculates the forward message of $\mathbf{h}_k^L$. Note that Sub-graph 2 represents the relation between the data and the channel based on the CP observation $\tilde{\mathbf{y}}_k$, and the E-step (Table 4.2.A) achieves data detection based on $\tilde{\mathbf{y}}_k$ and the M-step (Table 4.2.B) achieves channel estimation based on $\tilde{\mathbf{y}}_k$.

**Table 4.2:** Message passing for the EM algorithm in Sub-graph 2

**Inputs:** $\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k};\ \overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k}, \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{d}}_k};\ \left[\mathbf{h}_k^L\right]^{new}$

**Outputs:** $\mathbf{m}_{\tilde{\mathbf{b}}_k}, \mathbf{V}_{\tilde{\mathbf{b}}_k};\ \overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}$

**A. E-step of Sub-graph 2:**

1. initialize $\tilde{\mathbf{c}}_k^z$ using channel estimate $\left[\mathbf{h}_k^L\right]^{new}$, $z = 0, ..., S-1$

$$\tilde{\mathbf{c}}_k^0 = \sqrt{L}\mathbf{F}_L\left[\mathbf{h}_k^L\right]^{new}$$
$$\tilde{\mathbf{c}}_k^z = \mathbf{E}^H\tilde{\mathbf{c}}_k^{z-1}, \quad z = 1, ..., S-1$$

2. calculate the *a posteriori* message of $\tilde{\mathbf{b}}_k^z$

$$\mathbf{V}_{\tilde{\mathbf{b}}_k^z} = \left[\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k^z}^{-1} + \lambda_k^{-1} Diag\left(\tilde{\mathbf{c}}_k^{z*} \odot \tilde{\mathbf{c}}_k^z\right)\right]^{-1}$$
$$\mathbf{m}_{\tilde{\mathbf{b}}_k^z} = \mathbf{V}_{\tilde{\mathbf{b}}_k^z}\left[\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k^z}^{-1}\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k^z} + \lambda_k^{-1}\left(\tilde{\mathbf{c}}_k^{z*} \odot \overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k^z}\right)\right]$$

**B. M-step of Sub-graph 2:**

1. calculate the backward message of $\tilde{\mathbf{c}}_k^z$ as

$$\overleftarrow{\mathbf{V}}_{\tilde{\mathbf{c}}_k^z} = \lambda_k\left[\left(\mathbf{V}_{\tilde{\mathbf{b}}_k^z}\right)_{diag} + Diag\left(\mathbf{m}_{\tilde{\mathbf{b}}_k^z}^* \odot \mathbf{m}_{\tilde{\mathbf{b}}_k^z}\right)\right]^{-1}$$
$$\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{c}}_k^z} = \lambda_k^{-1}\overleftarrow{\mathbf{V}}_{\tilde{\mathbf{c}}_k^z}\left(\mathbf{m}_{\tilde{\mathbf{b}}_k^z}^* \odot \overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k^z}\right)$$

2. set the forward message of $\tilde{\mathbf{c}}''^{S-1}_k$ as

$$\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{c}}''^{S-1}_k} = \overleftarrow{\mathbf{m}}_{\tilde{\mathbf{c}}_k^{S-1}}, \quad \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}''^{S-1}_k} = \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{c}}_k^{S-1}}$$

3. for $z = S-1, ..., 1$ do

   a) $\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}'^{z-1}_k} = \mathbf{E}\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}''^{z}_k}\mathbf{E}^H, \quad \overrightarrow{\mathbf{m}}_{\tilde{\mathbf{c}}'^{z-1}_k} = \mathbf{E}\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{c}}''^{z}_k}$

   b) $\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}''^{z-1}_k} = \left[\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}'^{z-1}_k}^{-1} + \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{c}}^{z-1}_k}^{-1}\right]^{-1}$

   $\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{c}}''^{z-1}_k} = \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}''^{z-1}_k}\left[\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}'^{z-1}_k}^{-1}\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{c}}'^{z-1}_k} + \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{c}}^{z-1}_k}^{-1}\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{c}}^{z-1}_k}\right]$

   end for

4. calculate the forward message of $\mathbf{h}_k^L$ as

$$\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L} = \frac{1}{\sqrt{L}}\mathbf{F}_L^H\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{c}}''^{0}_k}, \ \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L} = \frac{1}{L}\mathbf{F}_L^H\overrightarrow{\mathbf{V}}_{\tilde{\mathbf{c}}''^{0}_k}\mathbf{F}_L$$

### 4.2.3 Exploiting the Variances of Channel Estimates

In Sub-graph 1 (Table 4.1.A) and Sub-graph 2 (Table 4.2.A), channel estimates $\left[\mathbf{h}_k^R\right]^{new}$ and $\left[\mathbf{h}_k^L\right]^{new}$ are used respectively to calculate the *a posteriori* messages of $\mathbf{b}_k$ and $\tilde{\mathbf{b}}_k$, instead of the exact channel itself. These estimates show partial knowledge of the channel, and can be unreliable at early iterations, as will be illustrated by the simulation results in Fig. 4.8 and Fig. 4.9. Therefore, rather than only using the means of the channel estimates like in Table 4.1, Table 4.2 and [34], we also include the variances of the channel estimates into the system models. To achieve this, (4.7) and (4.8) are rewritten as

$$\mathbf{y}_k = \mathbf{F}_N^H\Big[\mathbf{u}_k \odot \big(\sqrt{N}\mathbf{F}_N\hat{\mathbf{S}}\hat{\mathbf{h}}_k\big)\Big] + \underbrace{\mathbf{n}_k + \mathbf{F}_N^H\Big[\mathbf{u}_k \odot \big[\sqrt{N}\mathbf{F}_N\hat{\mathbf{S}}\big(\mathbf{h}_k - \hat{\mathbf{h}}_k\big)\big]\Big]}_{\mathbf{n}'_k}, \tag{4.46}$$

$$\tilde{\mathbf{y}}_k = \mathbf{F}_G^H\Big[\big(\mathbf{B}\mathbf{u}_k\big) \odot \big(\sqrt{G}\mathbf{F}_G\check{\mathbf{S}}\hat{\mathbf{h}}_k\big)\Big] + \underbrace{\mathbf{v}_k + \mathbf{F}_G^H\Big[\big(\mathbf{B}\mathbf{u}_k\big) \odot \big[\sqrt{G}\mathbf{F}_G\check{\mathbf{S}}\big(\mathbf{h}_k - \hat{\mathbf{h}}_k\big)\big]\Big]}_{\mathbf{v}'_k} \tag{4.47}$$

where $\mathbf{n}'_k$ and $\mathbf{v}'_k$ are regarded as the augmented noises that absorb both the channel noise and the uncertainty in the channel estimate ($\mathbf{v}'_k$ also absorbs the inter-block interference cancellation noise, i.e. the first and second terms in (4.5)).

To calculate the mean vectors and covariance matrices of $\mathbf{n}'_k$ and $\mathbf{v}'_k$, we first write their second terms respectively as

$$\mathbf{g}_k = \sqrt{N}\mathbf{F}_N^H Diag\big(\mathbf{u}_k\big)\mathbf{F}_N\hat{\mathbf{S}}\big(\mathbf{h}_k - \hat{\mathbf{h}}_k\big), \tag{4.48}$$

$$\mathbf{g}'_k = \sqrt{G}\mathbf{F}_G^H Diag\big(\mathbf{B}\mathbf{u}_k\big)\mathbf{F}_G\check{\mathbf{S}}\big(\mathbf{h}_k - \hat{\mathbf{h}}_k\big). \tag{4.49}$$

Then we decompose $\mathbf{h}_k$ into

$$\mathbf{h}_k = \hat{\mathbf{h}}_k + \boldsymbol{\epsilon}_k \tag{4.50}$$

where $\hat{\mathbf{h}}_k$ is deterministic with $\hat{\mathbf{h}}_k = \mathbf{m}_{\hat{\mathbf{h}}_k}$ and $\mathbf{m}_{\hat{\mathbf{h}}_k}$ is the mean vector of channel estimate $\hat{\mathbf{h}}_k$, and $\boldsymbol{\epsilon}_k$ is random with $\boldsymbol{\epsilon}_k \sim \mathcal{N}\big(\mathbf{0}, \mathbf{V}_{\hat{\mathbf{h}}_k}\big)$ and $\mathbf{V}_{\hat{\mathbf{h}}_k}$ is the covariance matrix of channel estimate $\hat{\mathbf{h}}_k$. It is obvious that the means of $\mathbf{g}_k$ and $\mathbf{g}'_k$ are null, i.e. $\mathbf{m}_{\mathbf{g}_k} = \mathbf{0}$, $\mathbf{m}_{\mathbf{g}'_k} = \mathbf{0}$. Then assuming that each of the modulation constellation points occurs equally likely and the energy of each symbol is normalized to 1, we have

$$\mathbf{V}_{\mathbf{u}_k} = \mathbf{I}_N. \tag{4.51}$$

It follows from the independence of the data and the channel that we obtain a simple expression for each of the covariance matrices of $\mathbf{g}_k$ and $\mathbf{g}'_k$. Details of the derivation can be found in Appendix C and the expressions are respectively

$$\overrightarrow{\mathbf{V}}_{\mathbf{g}_k} = \operatorname{tr}\big(\mathbf{V}_{\hat{\mathbf{h}}_k}\big)\mathbf{I}_N, \tag{4.52}$$

$$\overrightarrow{\mathbf{V}}_{\mathbf{g}'_k} \approx \frac{P}{G}\operatorname{tr}\big(\mathbf{V}_{\hat{\mathbf{h}}_k}\big)\mathbf{I}_G. \tag{4.53}$$

Hence, the mean vectors and covariance matrices of $\mathbf{n}'_k$ and $\mathbf{v}'_k$ are given by

$$\overrightarrow{\mathbf{m}}_{\mathbf{n}'_k} = \overrightarrow{\mathbf{m}}_{\mathbf{n}_k} = \mathbf{0}, \qquad \overrightarrow{\mathbf{V}}_{\mathbf{n}'_k} = \beta\mathbf{I}_N + \overrightarrow{\mathbf{V}}_{\mathbf{g}_k} = \beta'\mathbf{I}_N, \tag{4.54}$$

$$\overrightarrow{\mathbf{m}}_{\mathbf{v}'_k} = \overrightarrow{\mathbf{m}}_{\mathbf{v}_k}, \qquad \overrightarrow{\mathbf{V}}_{\mathbf{v}'_k} = \lambda_k\mathbf{I}_G + \overrightarrow{\mathbf{V}}_{\mathbf{g}'_k} \approx \lambda'_k\mathbf{I}_G \tag{4.55}$$

where $\beta' = \beta + \operatorname{tr}\big(\mathbf{V}_{\hat{\mathbf{h}}_k}\big)$ and $\lambda'_k = \lambda_k + \frac{P}{G}\operatorname{tr}\big(\mathbf{V}_{\hat{\mathbf{h}}_k}\big)$.

Note that if the channel is perfectly known at the receiver, the second terms of $\mathbf{n}'_k$ and $\mathbf{v}'_k$ are null and $\mathbf{n}'_k = \mathbf{n}_k$, $\mathbf{v}'_k = \mathbf{v}_k$. However, if the channel is unknown, neglecting the two terms results in unmodeled system driving noise and statistical modeling errors. Such errors may cause degradation in the system performance and even possibly non-convergence of the estimator [85]. To include the 'missing' system noise, which is not accounted for in Section 4.2.A and Section 4.2.B, one only needs to replace $\beta$ by $\beta'$ in Table 4.1 and $\lambda_k$ by $\lambda'_k$ in Table 4.2 without making other

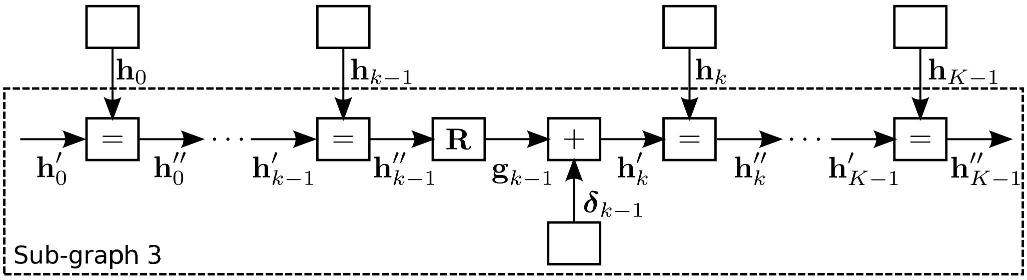


**Figure 4.4:** The factor graph for the AR model.

changes, since the mean vectors remain the same as shown in (4.54) and (4.55).

Note also that the difference between $\beta$ and $\beta'$, and $\lambda_k$ and $\lambda'_k$ is more prominent for high $E_b/N_0$ regime than for low $E_b/N_0$ regime, which is why the estimator becomes very sensitive as $E_b/N_0$ increases, as will be shown in Section 4.4.2.

### 4.2.4 Factor Graph for the AR Model and the Main Graph

Given the observations $\mathbf{y}_k$ and $\tilde{\mathbf{y}}_k$, Sub-graph 1 and Sub-graph 2 show constraints of the data and the channel, and the two sub-graphs can be combined together in order to integrate two copies of information and enhance the data detection and the channel estimation. Likewise, other relations such as the AR model (4.2), which shows the variation of the channel with respect to time, can also be used. Note that the AR model (4.2) only shows the constraint of the channel (not of the data) and thus can only be used to facilitate the channel estimation.

Fig. 4.4 (Sub-graph 3) depicts the factor graph for model (4.2) and Table 4.3 describes the message passing for Sub-graph 3. Note that the EM algorithm is not used in Sub-graph 3 since there is no multiplier node with unknown coefficients. Table 4.3 takes the *a priori* message of $\mathbf{h}_k$ as its input, and outputs the *a posteriori* message of $\mathbf{h}_k$ for $k = 0, ..., K-1$.

Fig. 4.5 demonstrates the combination of Sub-graph 1, Sub-graph 2 and Sub-graph 3 in one graph (the Main graph). The Main graph corresponds to a larger EM algorithm, with the E-steps of Sub-graph 1 and Sub-graph 2 being the E-step

**Table 4.3:** Message passing for the AR model in Sub-graph 3

```
Inputs: $\overrightarrow{\mathbf{m}}_{\mathbf{h}_k},\overrightarrow{\mathbf{V}}_{\mathbf{h}_k}$        Outputs: $\mathbf{m}_{\mathbf{h}_k},\mathbf{V}_{\mathbf{h}_k}$

A. Forward recursion: calculate $(\overrightarrow{\mathbf{m}}_{\mathbf{h}'_k},\overrightarrow{\mathbf{V}}_{\mathbf{h}'_k})$, $k = 0,...,K-1$
  1) set the forward message of $\mathbf{h}'_0$ as
        $\overrightarrow{\mathbf{m}}_{\mathbf{h}'_0} = \mathbf{0}, \quad \overrightarrow{\mathbf{V}}_{\mathbf{h}'_0} = \boldsymbol{\infty}$
  2) for $k = 0,...,K-2$ do
       a) $\overrightarrow{\mathbf{V}}_{\mathbf{h}''_k} = \left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k}^{-1} + \overrightarrow{\mathbf{V}}_{\mathbf{h}'_k}^{-1}\right]^{-1}$
          $\overrightarrow{\mathbf{m}}_{\mathbf{h}''_k} = \overrightarrow{\mathbf{V}}_{\mathbf{h}''_k}\left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}_k} + \overrightarrow{\mathbf{V}}_{\mathbf{h}'_k}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}'_k}\right]$
       b) $\overrightarrow{\mathbf{V}}_{\mathbf{g}_k} = \mathbf{R}\overrightarrow{\mathbf{V}}_{\mathbf{h}''_k}\mathbf{R}^H, \qquad \overrightarrow{\mathbf{m}}_{\mathbf{g}_k} = \mathbf{R}\overrightarrow{\mathbf{m}}_{\mathbf{h}''_k}$
       c) $\overrightarrow{\mathbf{V}}_{\mathbf{h}'_{k+1}} = \overrightarrow{\mathbf{V}}_{\mathbf{g}_k} + \overrightarrow{\mathbf{V}}_{\boldsymbol{\delta}_k}, \qquad \overrightarrow{\mathbf{m}}_{\mathbf{h}'_{k+1}} = \overrightarrow{\mathbf{m}}_{\mathbf{g}_k}$
     end for

B. Backward recursion: calculate $(\overleftarrow{\mathbf{m}}_{\mathbf{h}''_k},\overleftarrow{\mathbf{V}}_{\mathbf{h}''_k})$, $k = 0,...,K-1$
  1) set the backward message of $\mathbf{h}''_{K-1}$ as
        $\overrightarrow{\mathbf{m}}_{\mathbf{h}''_{K-1}} = \mathbf{0}, \quad \overrightarrow{\mathbf{V}}_{\mathbf{h}''_{K-1}} = \boldsymbol{\infty}$
  2) for $k = K-1,...,1$ do
       a) $\overleftarrow{\mathbf{V}}_{\mathbf{h}'_k} = \left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k}^{-1} + \overleftarrow{\mathbf{V}}_{\mathbf{h}''_k}^{-1}\right]^{-1}$
          $\overleftarrow{\mathbf{m}}_{\mathbf{h}'_k} = \overleftarrow{\mathbf{V}}_{\mathbf{h}'_k}\left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}_k} + \overleftarrow{\mathbf{V}}_{\mathbf{h}''_k}^{-1}\overleftarrow{\mathbf{m}}_{\mathbf{h}''_k}\right]$
       b) $\overleftarrow{\mathbf{V}}_{\mathbf{g}_{k-1}} = \overleftarrow{\mathbf{V}}_{\mathbf{h}'_k} + \overrightarrow{\mathbf{V}}_{\boldsymbol{\delta}_{k-1}}, \qquad \overleftarrow{\mathbf{m}}_{\mathbf{g}_{k-1}} = \overleftarrow{\mathbf{m}}_{\mathbf{h}'_k}$
       c) $\overleftarrow{\mathbf{V}}_{\mathbf{h}''_{k-1}} = \left[\mathbf{R}^H\overleftarrow{\mathbf{V}}_{\mathbf{g}_{k-1}}^{-1}\mathbf{R}\right]^{-1}$
          $\overleftarrow{\mathbf{m}}_{\mathbf{h}''_{k-1}} = \overleftarrow{\mathbf{V}}_{\mathbf{h}''_{k-1}}\left[\mathbf{R}^H\overleftarrow{\mathbf{V}}_{\mathbf{g}_{k-1}}^{-1}\overleftarrow{\mathbf{m}}_{\mathbf{g}_{k-1}}\right]$
     end for

C. Combination: calculate $(\mathbf{m}_{\mathbf{h}_k},\mathbf{V}_{\mathbf{h}_k})$, $k = 0,...,K-1$
     for $k = 0,...,K-1$ do
          $\mathbf{V}_{\mathbf{h}_k} = \left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k}^{-1} + \overrightarrow{\mathbf{V}}_{\mathbf{h}'_k}^{-1} + \overleftarrow{\mathbf{V}}_{\mathbf{h}''_k}^{-1}\right]^{-1}$
          $\mathbf{m}_{\mathbf{h}_k} = \mathbf{V}_{\mathbf{h}_k}\left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}_k} + \overrightarrow{\mathbf{V}}_{\mathbf{h}'_k}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}'_k} + \overleftarrow{\mathbf{V}}_{\mathbf{h}''_k}^{-1}\overleftarrow{\mathbf{m}}_{\mathbf{h}''_k}\right]^{-1}$
     end for
```

of the whole EM algorithm (which detects data) and the M-steps of Sub-graph 1 and Sub-graph 2 and the message passing for Sub-graph 3 being the M-step of the whole EM algorithm (which estimates the channel). Between the three sub-graphs, the message passing process can vary significantly according to different scheduling schemes, which we will discuss in detail in the next section.

## 4.3 Message Scheduling and Discussion

In this section, two scheduling schemes {R-AR-L} and {R-L-AR} are given as an example to show the possible message passings between the three sub-graphs in Fig. 4.5. More scheduling schemes are listed in Table 4.5 and will be compared in Section 4.4. For the name of each scheme, {R}, {L} and {AR} respectively stand for the right branch of the Main graph (Sub-graph 1), the left branch (Sub-graph 2) and the AR-model branch (Sub-graph 3).

### 4.3.1 Two Scheduling Examples

Assume that there are $I_{out}$ outer iterations between the equalizer and the decoder in Fig. 4.1. For the $q$th outer iteration, $q = 1, ..., I_{out}$, the equalizer takes $\left(\mathbf{m}_k^{apri,q}, \mathbf{V}_k^{apri,q}\right)$ as its input and produces $\left(\mathbf{m}_k^{ext,q}, \mathbf{V}_k^{ext,q}\right)$ as its output. Inside the equalizer, $I_{in}$ inner iterations are performed and for the $p$th inner iteration, $p = 1, ..., I_{in}$, the following steps take place for the schemes {R-AR-L} and {R-L-AR} respectively.

**The scheduling scheme {R-AR-L}:**

S1. Table 4.1. The input $\left(\overleftarrow{\mathbf{m}}_{\mathbf{d}_k}, \overleftarrow{\mathbf{V}}_{\mathbf{d}_k}\right)$ is obtained from (4.31); $\left(\overrightarrow{\mathbf{m}}_{\mathbf{b}_k}, \overrightarrow{\mathbf{V}}_{\mathbf{b}_k}\right)$ and $\left[\mathbf{h}_k^R\right]^{new}$ are initialized by

$$\overrightarrow{\mathbf{m}}_{\mathbf{b}_k} = \mathbf{m}_k^{apri,q},\ \overrightarrow{\mathbf{V}}_{\mathbf{b}_k} = \mathbf{V}_k^{apri,q},\ \left[\mathbf{h}_k^R\right]^{new} = \hat{\mathbf{h}}_k^{p-1,q} \tag{4.56}$$

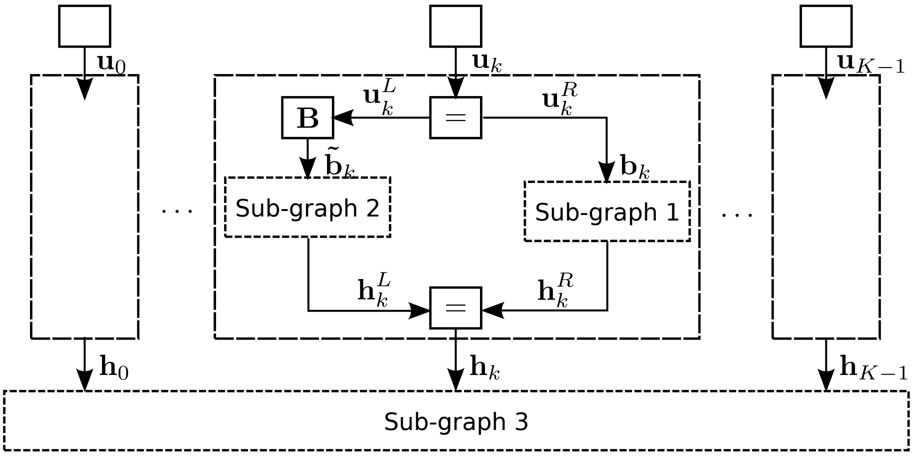


**Figure 4.5:** The Main graph that combines Sub-graph 1, 2 and 3.

where $\mathbf{m}_k^{apri,q}$ and $\mathbf{V}_k^{apri,q}$ are the *a priori* mean vector and covariance matrix of $\mathbf{u}_k$ passed from the decoder in the $q$th outer iteration, and $\hat{\mathbf{h}}_k^{p-1,q}$ is the channel estimate of the $(p-1)$th inner iteration in the $q$th outer iteration. Note that for the first outer iteration, $\mathbf{m}_k^{apri,1} = \mathbf{0}$, $\mathbf{V}_k^{apri,1} = \mathbf{I}$, and $\hat{\mathbf{h}}_k^{0,1} = \mathbf{0}$. The output $\left(\mathbf{m}_{\mathbf{b}_k}, \mathbf{V}_{\mathbf{b}_k}\right)$ is used in S3 and S5, and $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R}\right)$ is used in S2.

S2. Table 4.3. The input $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k}\right)$ is initialized by

$$\overrightarrow{\mathbf{m}}_{\mathbf{h}_k} = \overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R}, \quad \overrightarrow{\mathbf{V}}_{\mathbf{h}_k} = \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R} \tag{4.57}$$

where $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R}\right)$ is the channel estimate from S1. The output $\left(\mathbf{m}_{\mathbf{h}_k}, \mathbf{V}_{\mathbf{h}_k}\right)$ is used in S3 and S4.

S3. Table 4.2. The input $\left(\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)$ is obtained by

$$\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k} = \mathbf{B}\mathbf{m}_k^{apri,q}, \quad \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k} = \mathbf{B}\mathbf{V}_k^{apri,q}\mathbf{B}^H \tag{4.58}$$

and $\left(\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k}, \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{d}}_k}\right)$ is obtained from (4.36)-(4.41), where $\left(\mathbf{m}_{\mathbf{u}_k}, \mathbf{V}_{\mathbf{u}_k}\right)$ in (4.37) (4.40) is initialized by $\left(\mathbf{m}_{\mathbf{b}_k}, \mathbf{V}_{\mathbf{b}_k}\right)$ from S1 and $\mathbf{m}_{\mathbf{h}_k}$ in (4.37) (4.40) by $\mathbf{m}_{\mathbf{h}_k}$ from S2. The input $\left[\mathbf{h}_k^L\right]^{new}$ is initialized by $\mathbf{m}_{\mathbf{h}_k}$ from S2 as well. The output $\left(\mathbf{m}_{\tilde{\mathbf{b}}_k}, \mathbf{V}_{\tilde{\mathbf{b}}_k}\right)$ is used in S5, and $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}\right)$ is used in S4.

S4. The channel estimate from S2, $\left(\mathbf{m}_{\mathbf{h}_k}, \mathbf{V}_{\mathbf{h}_k}\right)$, and the channel estimate from S3, $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}\right)$, are combined as

$$\mathbf{V}_{\hat{\mathbf{h}}_k^p} = \left[\mathbf{V}_{\mathbf{h}_k}^{-1} + \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}^{-1}\right]^{-1}, \tag{4.59}$$

$$\mathbf{m}_{\hat{\mathbf{h}}_k^p} = \mathbf{V}_{\hat{\mathbf{h}}_k^p}^{-1}\left[\mathbf{V}_{\mathbf{h}_k}^{-1}\mathbf{m}_{\mathbf{h}_k} + \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}\right] \tag{4.60}$$

which yields the channel estimate of the $p$th inner iteration, i.e. $\hat{\mathbf{h}}_k^{p,q} = \mathbf{m}_{\hat{\mathbf{h}}_k^p}$.

S5. After $I_{in}$ inner iterations (i.e. running S1-S4 for $I_{in}$ times), the equalizer calculates the extrinsic means and variances of the data symbols $\left(\mathbf{m}_k^{ext,q}, \mathbf{V}_k^{ext,q}\right)$, using $\left(\mathbf{m}_{\mathbf{b}_k}, \mathbf{V}_{\mathbf{b}_k}\right)$ from S1 and $\left(\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)$ from S3 (see [71])[1]. These extrinsic means and variances can be converted into log-likelihood ratios (LLRs) of the coded bits as in [25], which are then fed back to the decoder.

**The scheduling scheme {R-L-AR}:**

S1. Table 4.1. Same inputs as S1 in {R-AR-L}. The output $\left(\mathbf{m}_{\mathbf{b}_k}, \mathbf{V}_{\mathbf{b}_k}\right)$ is used in S2 and S4, and $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R}\right)$ is used in S2 and S3.

S2. Table 4.2. The input $\left(\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)$ is obtained the same way as in (4.58), and $\left(\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{d}}_k}, \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{d}}_k}\right)$ is obtained from (4.36)-(4.41), where $\left(\mathbf{m}_{\mathbf{u}_k}, \mathbf{V}_{\mathbf{u}_k}\right)$ in (4.37) (4.40) is initialized by $\left(\mathbf{m}_{\mathbf{b}_k}, \mathbf{V}_{\mathbf{b}_k}\right)$ from S1 and $\mathbf{m}_{\mathbf{h}_k}$ in (4.37) (4.40) by $\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R}$ from S1. The input $\left[\mathbf{h}_k^L\right]^{new}$ is also initialized by $\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R}$ from S1. The output $\left(\mathbf{m}_{\tilde{\mathbf{b}}_k}, \mathbf{V}_{\tilde{\mathbf{b}}_k}\right)$ is used in S4, and $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}\right)$ is used in S3.

S3. Table 4.3. The input $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k}\right)$ is initialized by

$$\overrightarrow{\mathbf{V}}_{\mathbf{h}_k} = \left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R}^{-1} + \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}^{-1}\right]^{-1}, \tag{4.61}$$

$$\overrightarrow{\mathbf{m}}_{\mathbf{h}_k} = \overrightarrow{\mathbf{V}}_{\mathbf{h}_k}^{-1}\left[\overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R} + \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}^{-1}\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}\right] \tag{4.62}$$

where $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^R}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^R}\right)$ is the channel estimate from S1 and $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}\right)$ is the

[1] In S3, Table 4.2 only outputs $\left(\mathbf{m}_{\tilde{\mathbf{b}}_k}, \mathbf{V}_{\tilde{\mathbf{b}}_k}\right)$ but $\left(\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)$ can be obtained by $\overleftarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k} = \left[\mathbf{V}_{\tilde{\mathbf{b}}_k}^{-1} - \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}^{-1}\right]^{-1}$ and $\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k} = \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\left[\mathbf{V}_{\tilde{\mathbf{b}}_k}^{-1}\mathbf{m}_{\tilde{\mathbf{b}}_k} - \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}^{-1}\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}\right]$, where $\left(\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)$ is from (4.58).

channel estimate from S2. The output of this step is the channel estimate of the $p$th inner iteration, i.e. $\hat{\mathbf{h}}_k^{p,q} = \mathbf{m}_{\mathbf{h}_k}$.

S4. Same as S5 in {R-AR-L}, except that $\left(\overleftarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overleftarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)$ is from S2 instead of S3.

### 4.3.2 Discussion

Regarding how to use Sub-graph 2, here we discuss two variations, {L$_{\mathrm{C}}$} and {L$_{\mathrm{D}}$}. {L$_{\mathrm{C}}$} only uses Sub-graph 2 for channel estimation while {L$_{\mathrm{D}}$} only uses Sub-graph 2 for data detection. The difference between {L} and {L$_{\mathrm{C}}$} is that in {L$_{\mathrm{C}}$}, the output of Table 4.2, $\left(\overrightarrow{\mathbf{m}}_{\tilde{\mathbf{b}}_k}, \overrightarrow{\mathbf{V}}_{\tilde{\mathbf{b}}_k}\right)$, is not combined for the calculation of $\left(\mathbf{m}_k^{ext,q}, \mathbf{V}_k^{ext,q}\right)$ as in S5, {R-AR-L}, while the difference between {L} and {L$_{\mathrm{D}}$} is that in {L$_{\mathrm{D}}$}, the M-step of Table 4.2 is omitted and its supposed output, $\left(\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L}, \overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}\right)$, is set to $\overrightarrow{\mathbf{m}}_{\mathbf{h}_k^L} = \mathbf{0}$ and $\overrightarrow{\mathbf{V}}_{\mathbf{h}_k^L}^{-1} = \mathbf{0}$, i.e. no channel estimate is provided by Sub-graph 2.

Table 4.4 lists the complexity required by Sub-graph 1 and Sub-graph 2 to process one data block (only the complex multiplications are counted). Sub-graph 3 ({AR}) is not listed because it only needs scalar computations which are negligible. As we can see, there are two categories of calculation in Table 4.4.

1. One needs to be done every inner iteration. Both {L$_{\mathrm{C}}$} and {L} use CP for channel estimation and the *a priori* covariance matrix of $\tilde{\mathbf{b}}_k$ is not diagonal (see (4.58)), therefore, $L$-length matrix inversion and matrix-vector multiplication (with the complexity of $L^3$ and $L^2$ respectively) are involved in the E-steps of Table 4.1 and Table 4.2. In this category, the remaining complexity in {L$_{\mathrm{C}}$} and {L} and the overall complexity of {L$_{\mathrm{D}}$} and {R} are dominated by the $L$-length FFTs.
2. The other needs to be done every outer iteration. When Sub-graph 2 is employed for data detection, i.e. {L$_{\mathrm{D}}$} and {L}, combination of two copies of data estimates and calculation of the extrinsic information as in S5, {R-AR-L} require complexity of $2NlogN + 4GlogG + 2G$ [71]. When Sub-graph 2 is em-

**Table 4.4:** Complexity per data block required by the three sub-graphs

| | per inner iteration | per outer iteration |
|---|---|---|
| {R} | $NlogL + 4(T-1)LlogL$<br>$+2LlogL$ | — |
| {L$_\text{C}$} | $GlogL + GL^2 + GL$<br>$+4(S-1)LlogL + 2LlogL$ | $2NlogN + 2GlogG$<br>$+GL^2 + GL$ |
| {L$_\text{D}$} | $GlogL$ | $2NlogN + 4GlogG + 2G$ |
| {L} | $GlogL + GL^2 + GL$<br>$+4(S-1)LlogL + 2LlogL$ | $4NlogN + 6GlogG$<br>$+GL^2 + GL + 2G$ |

ployed for channel estimation, i.e. {L$_\text{C}$} and {L}, calculation of (4.58) renders complexity of $2NlogN + 2GlogG + GL^2 + GL$.

Since {L$_\text{D}$} requires lower complexity than {L$_\text{C}$} or {L} and in terms of system performance, {L$_\text{D}$} is more advantageous than {L$_\text{C}$} and has similar performance to {L} as will be shown in Section 4.4.3, we propose the scheduling scheme {{R-AR}*$I_{in}$-L$_\text{D}$}\*$I_{out}$ as an effective trade-off between the system performance and complexity. The overall complexity of {{R-AR}*$I_{in}$-L$_\text{D}$}\*$I_{out}$ is dominated by the second category of calculation and is $\mathcal{O}(NlogN)$ per data block per outer iteration, much lower than that of the equalizer in [63] with $(N+P)^3 * I_{in}$ per data block per outer iteration.

The use of interference cancellation in {L} (or {L$_\text{C}$}, {L$_\text{D}$}) and the use of AR model {AR} may cause an increase in latency compared to the conventional equalizer with perfect channel information. However, such a latency is acceptable in turbo receiver since the use of interleaver/de-interleaver requires the whole sequence associated with a codeword to be processed before being passed to the decoder or back to the equalizer.

**Table 4.5:** Different simulation settings

| Case | Channel status | Modified noise models[1] | CP used for channel estimation[2] | CP used for data detection[3] | Scheduling scheme[4] |
|---|---|---|---|---|---|
| a | Known | – | – | N | {R} *1 *10 |
| b | | – | – | Y | {R-L$_\text{D}$} *1 *10 |
| c | Unknown | N | N | N | {R-AR} *3 *10 |
| d | | Y | N | N | {R-AR} *3 *10 |
| e | | N | N | Y | {R-AR-L$_\text{D}$}*3 *10 |
| f | | Y | N | Y | {R-AR-L$_\text{D}$}*3 *10 |
| g | | Y | Y | N | {R-AR-L$_\text{C}$}*3 *10 |
| h | | Y | Y | Y | {R-AR-L} *3 *10 |
| i | | Y | Y | Y | {R-AR-L} *5 *10 |
| j | | Y | Y | Y | {R-AR-L} *1 *30 |
| k | | Y | Y | Y | {R-L-AR} *3 *10 |
| l | | Y | Y | Y | {{R-AR}*3- L$_\text{D}$}*10 |

[1] 'Y' – modified noise models (4.54) and (4.55) are in use; 'N' – otherwise.

[2] 'Y' – CP is used for channel estimation; 'N' – otherwise.

[3] 'Y' – CP is used for data detection; 'N' – otherwise.

[4] The format is {scheduling scheme between three sub-graphs}*$I_{in}$*$I_{out}$. For example, {R-AR-L}*3*10 means 3 inner iterations over {R}, {AR} and {L} and in this particular order {R-AR-L}, and 10 outer iterations between the equalizer and the decoder.

## 4.4 Simulation Results

A rate-1/2 convolutional code $(23, 35)_8$, an S-random interleaver, a 4QAM modulator with Gray mapping and the BCJR-based decoder are employed. Each symbol sequence consists of 64 data blocks, i.e. $K = 64$, and each data block has 64 symbols before CP insertion, i.e. $N = 64$. With 1/4 ratio, the CP length is 16, i.e. $P = 16$. For each $E_b/N_0$ point, $10^7$ sequences are simulated to get an average BER (bit error rate). The MSE (mean squared error) of channel estimation is obtained by averaging $\sum_{l=0}^{L-1} |\hat{h}^l - h^l|^2$ over all the $10^7$ sequences, where $h^l$ denotes the true coefficient of the $l$th tap and $\hat{h}^l$ denotes the estimate of the $l$th tap.

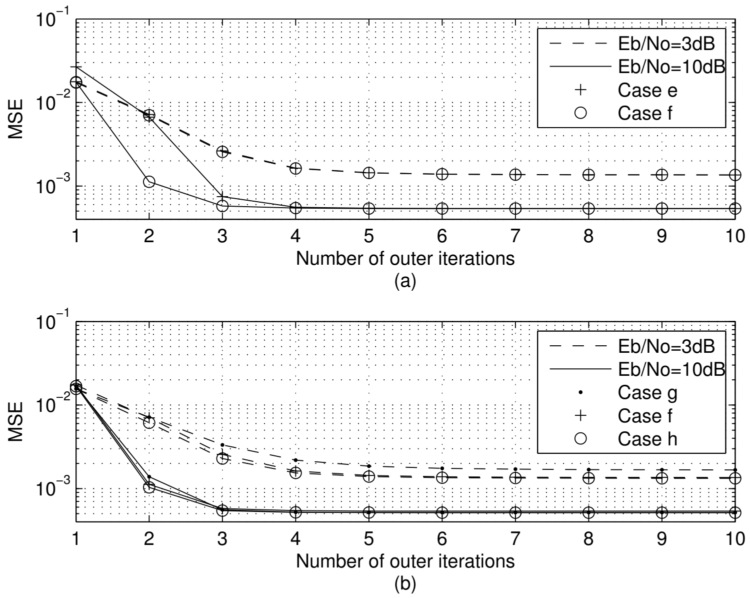
The time-varying multipath channel has 16 taps, i.e. $L = 16$, and the power delay profile is given by $\eta_l = e^{-0.1l}$, $l = 0, ..., L-1$. The energy of the channel is normalized to 1, i.e. $\mathrm{E}\left[\frac{1}{L}\sum_{l=0}^{L-1}|h_k^l|^2\right] = 1$, $k = 0, ..., K-1$. The carrier frequency is set to $2GHz$, the symbol duration is $0.25\mu s$ and the mobile velocity is $550km/hr$. Therefore, the duration of a data block is $T_s = 20\mu s$, the maximum Doppler frequency is $f_D = 926Hz$ and the value of $\rho$ in the matrix $\mathbf{R}$ in (4.2) is $J_0(2\pi f_D T_s) = 0.9966$.

The 25th root length-139 Chu sequence [86] is selected to be the training sequence, as defined in the long term evolution (LTE) specifications [42]. To avoid spectral loss, the Chu sequence is periodically extended and superimposed onto the data symbol sequence [87]. The power ratio of the training sequence to the data sequence is 0.1, i.e. the power overhead is 0.45dB.

Simulations are carried out with 12 different settings as listed in Table 4.5, which can be categorized into three sets. Set 1 includes Cases $a - f$, and compares the proposed joint data detection and channel estimation cases with the known channel cases and shows the effect of remodeling the system noise. Set 2 includes Cases $d$, $f$, $g$, $h$, and investigates the different uses of the CP, i.e. using it for channel estimation/data detection only or for neither/both. Set 3 includes Cases $h$, $i$, $j$, $k$, $l$, and compares the different scheduling schemes of the equalizer, where the performance of Case $h$ serves as a benchmark.

### 4.4.1 MSE Performance

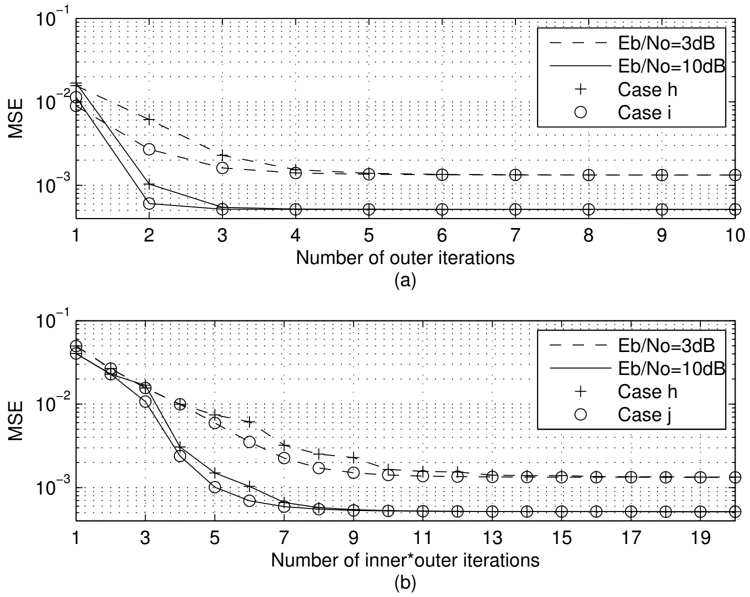
Fig. 4.6(a) compares Case $e$ with Case $f$ and shows that channel estimation is more sensitive to the modeling error at high $E_b/N_0$ than at low $E_b/N_0$, which is consistent with the analysis in the last paragraph of Section 4.2.3. To be specific, for the first two iterations, the MSE performance of $E_b/N_0 = 10$dB is worse than



(a)

(b)

**Figure 4.6:** The effect of (a) modelling error and (b) different uses of CP on the MSE performance of channel estimation.



(a)

(b)

**Figure 4.7:** The effect of (a) increasing the number of inner iterations per outer iteration and (b) exchanging the inner iterations for outer iterations on the MSE performance of channel estimation.

that of $E_b/N_0 = 3$dB when $\mathbf{n}_k = 2\sigma^2 \mathbf{I}_{N\times 1}$ and (4.38) are used for noise models (Case $e$). Fig. 4.6(b) shows that different uses of CP only have slight impact on the MSE performance of channel estimation at low $E_b/N_0$ regime. Fig. 4.7 compares two scheduling schemes, Cases $i$ and $j$, with the benchmark, Case $h$, and shows that both increasing the number of inner iterations and exchanging inner iterations for outer iterations can boost the speed of convergence of channel estimation.

### 4.4.2 BER Performance of Set 1

When the CP is not in use (i.e. the conventional system), Fig. 4.8 compares the BER performance with and without the modified noise models (4.54) and (4.55) employed when the channel is unknown (Cases $d$ and $c$) with the BER performance when the channel is perfectly known (Case $a$). Fig. 4.9 compares the corresponding cases (Cases $f$, $e$, $b$) when the CP is used for data detection. Note that Cases $a$ and $b$ are the same as the cases considered in [71].

After convergence, Cases $c$ and $d$ are both within 0.5dB from Case $a$ at the BER of $10^{-5}$ and so are for Cases $e$, $f$ and $b$, which means that the proposed equalization algorithm achieves effective channel estimation. Also, Case $a$ reaches the BER of $10^{-5}$ at 7.3dB and Case $b$ at 9.1dB, i.e. there is a gain of 1.8dB achieved by exploiting CP for data detection. For Cases $c$ and $e$, the BER curves tend to rise as $E_b/N_0$ increases at the first two outer iterations, while Cases $d$ and $f$ do not have this behavior, which is consistent with the analysis in the last paragraph of Section 4.2.3. After convergence, Cases $c$ and $d$ (or $e$ and $f$) have the same BER performance because both cases can produce reliable channel estimates at later iterations regardless of the noise models.

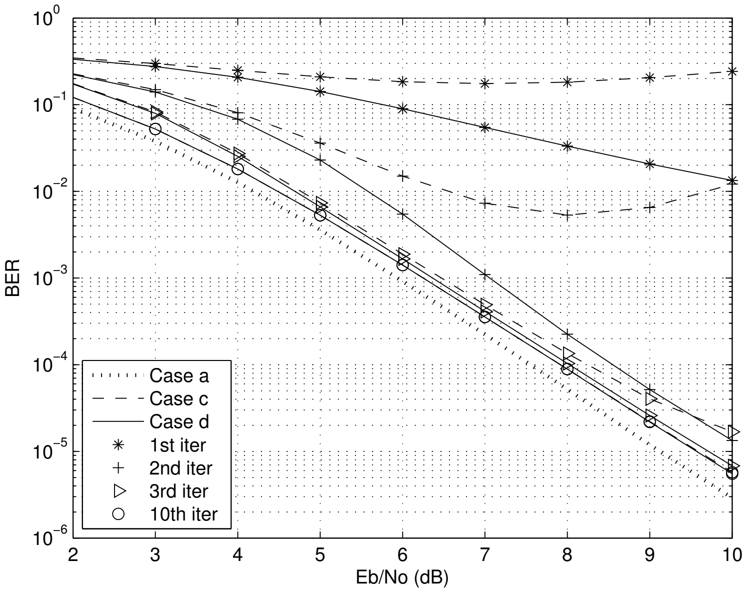


**Figure 4.8:** The BER performance with and without employing the modified noise models when CP is not exploited.

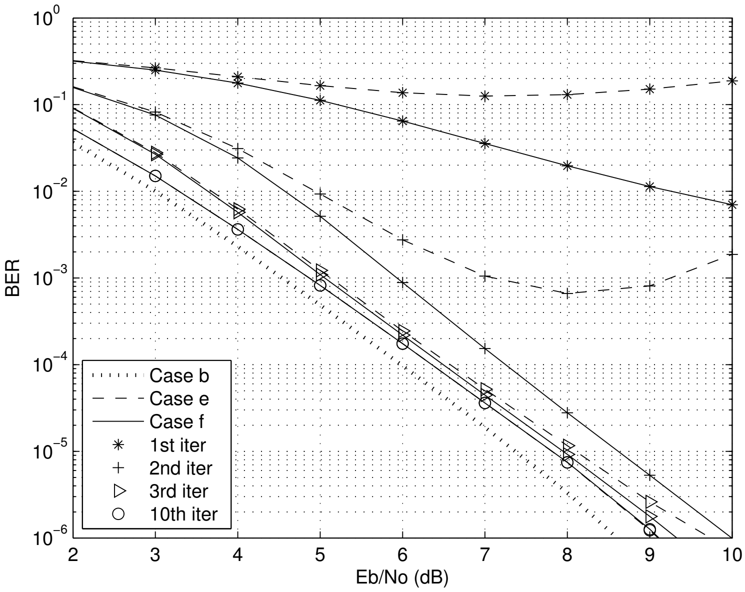


**Figure 4.9:** The BER performance with and without employing the modified noise models when CP is exploited for data detection.

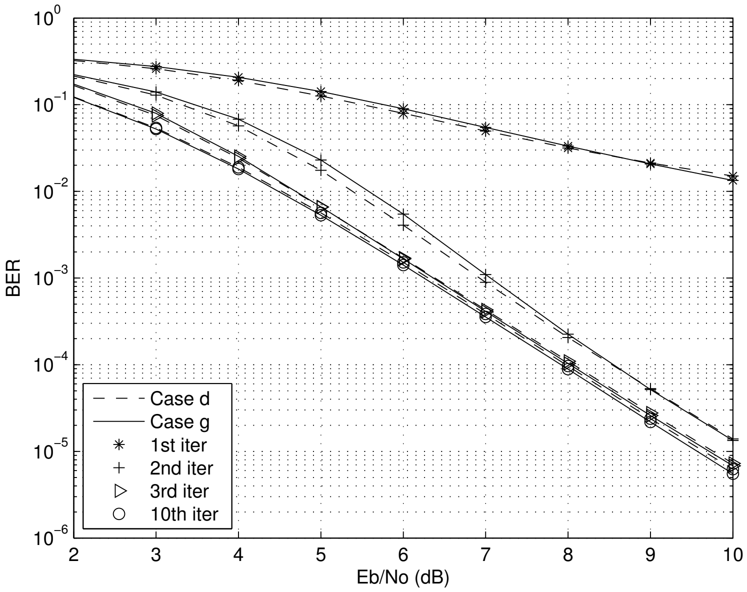


**Figure 4.10:** The BER performance when CP is not exploited and when CP is exploited for channel estimation only.

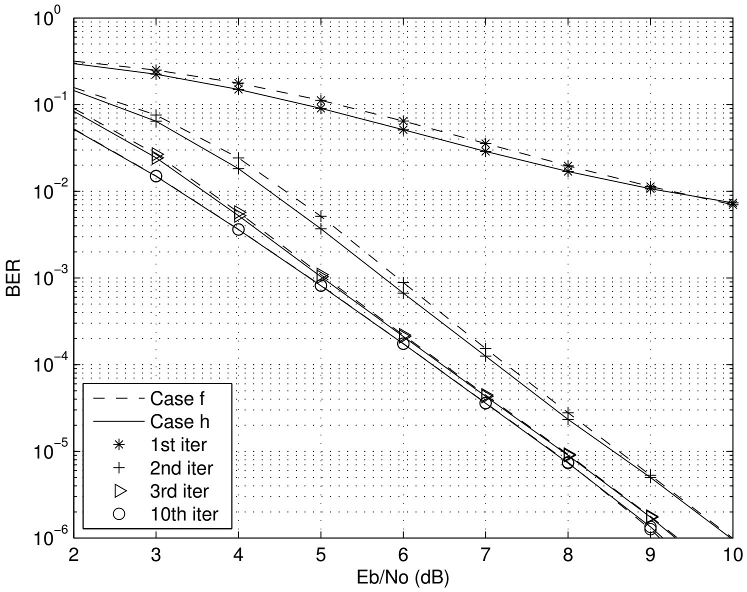


**Figure 4.11:** The BER performance when CP is exploited for data detection only and when CP is exploited for both channel estimation and data detection.

### 4.4.3 BER Performance of Set 2

With the modified noise models (4.54) and (4.55) employed, the BER performance of the four combinations of CP being used for channel estimation and/or data detection (Cases $d$, $f$, $g$, $h$) is compared in Fig. 4.10 and Fig. 4.11. Note that Case $g$ is similar to the proposed equalizer in [63] in that they both use CP for channel estimation only, but they are different because [63] does not consider different noise models and if listed in Table 4.5, it would be equivalent to the 'N-Y-N' case and have similar performance to Case $c$ (see [72]). Comparing Case $d$ with Case $g$ (or $f$ with $h$), using CP for channel estimation brings about a gain of 0.2dB, while comparing Case $d$ with Case $f$ (or $g$ with $h$) shows a 1.8dB gain obtained by using CP for data detection. Apparently, the latter way of using CP is more beneficial than the former in terms of system BER performance. Also, as seen from Table 4.4, $\{\mathrm{L_D}\}$ costs less than $\{\mathrm{L_C}\}$ in terms of complexity, making using CP for data detection even more favorable.

### 4.4.4 BER Performance of Set 3

Fig. 4.12 shows that with a higher number of inner iterations (Case $i$) the system can converge faster. In Fig. 4.13, the inner iterations are exchanged for an equal number of outer iterations, i.e. 3 inner iterations and 10 outer iterations in Case $h$ are changed to 1 inner iteration and 30 outer iterations in Case $j$. In the inner*outer iteration scale, Case $h$ converges after the 9th iteration (its 3rd outer iteration), while Case $j$ converges after the 6th iteration (its 6th outer iteration). Note that in both Fig. 4.12 and Fig. 4.13, faster convergence is achieved at the cost of higher complexity, i.e. more calculations in the equalizer or in the decoder.

With a different order of message passing through the three sub-graphs, Fig. 4.14 shows that {R-AR-L}*3*10 (Case $h$) produces a slightly better BER performance

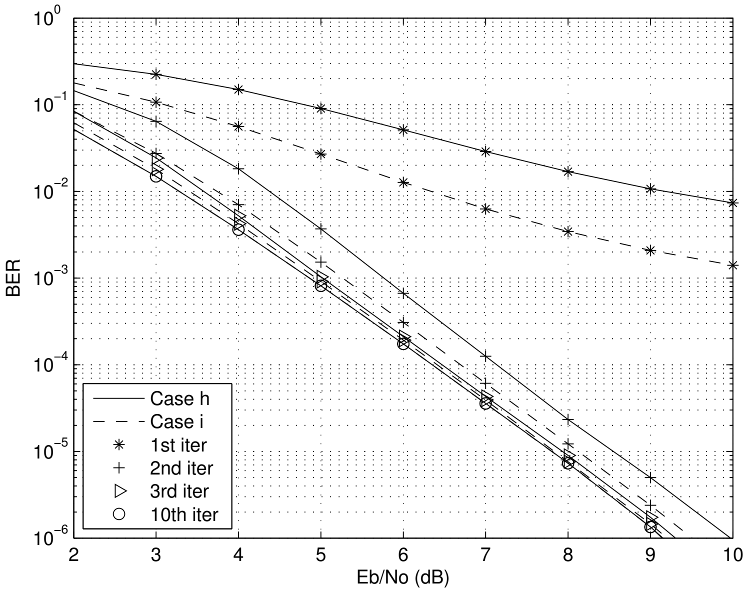


**Figure 4.12:** The BER performance when the number of inner iterations is increased.

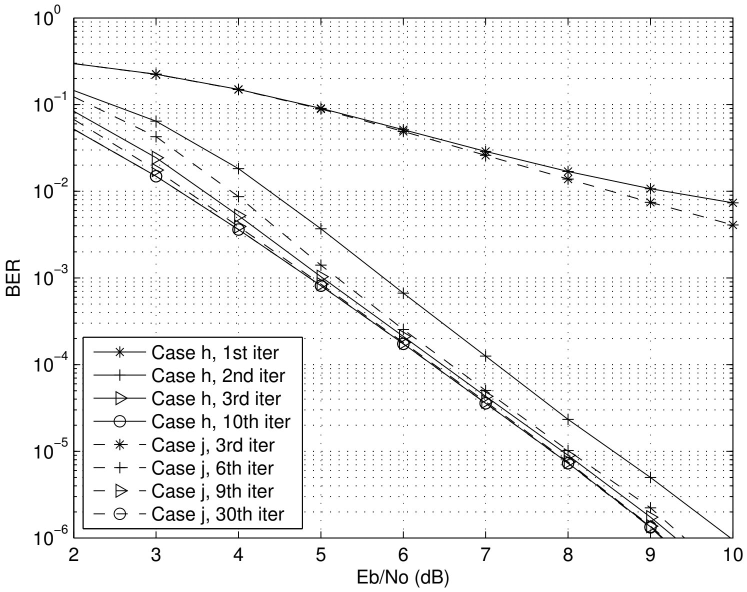


**Figure 4.13:** The BER performance when the inner iterations are exchanged for an equal number of outer iterations.

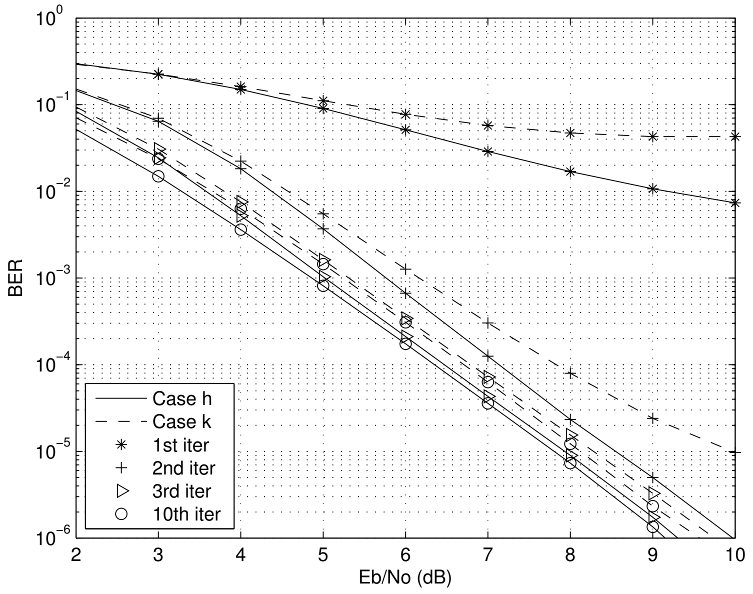


**Figure 4.14:** The BER performance when the order of message passing through three sub-graphs is changed.

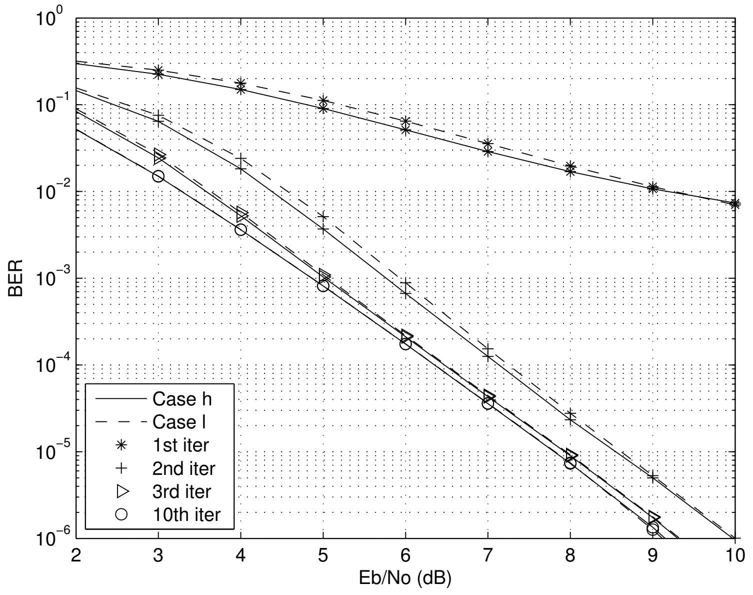


**Figure 4.15:** The BER performance when different numbers of iterations are employed over three sub-graphs.

than {R-L-AR}*3*10 (Case $k$). The reason is that in each inner iteration, {R-AR} provides more accurate channel estimate than {R} does, improving the performance of interference cancellation and the update of the augmented noise in {L}.

As the trade-off between performance and complexity (see Section 4.3.2), Case $l$ achieves similar BER performance to Case $h$ as shown in Fig. 4.15, but with lower complexity since {$\mathrm{L_D}$} is less costly than {L} in complexity (see Table 4.4) and also there is less iterations over {$\mathrm{L_D}$} than over {R-AR}. The reason for the BER performance of two cases to be so close is that the channel estimate from {R-AR} alone is reliable enough for {$\mathrm{L_D}$} and the equalizer to detect the data properly.

## 4.5 Summary

We have proposed an approach to exploiting the CP for joint detection, decoding and channel estimation via the EM algorithm and message passing in OFDM systems. Models of the non-CP part of data, CP part of data and the time-varying multipath channel are presented in FFGs and message passing processes through graphs and scheduling schemes between graphs are discussed. Modification is made in the system noise models to accommodate both the means and variances of channel estimates. A scheduling scheme is proposed as an effective trade-off between the system performance and complexity. Simulation results show that the proposed approach effectively achieves channel estimation and data detection and the modification in system noise models improves the BER performance at early iterations. Also, the proposed scheduling scheme achieves good BER performance with lower complexity.

# Chapter 5

# Conclusion

## 5.1 Thesis Summary

This thesis has considered applications of the turbo receiver in SC-FDE and OFDM systems. Equalization algorithms have been proposed based on the linear MMSE principle with the purpose of exploiting the cyclic prefix for data detection and/or channel estimation. Factor graphs and the Gaussian message passing technique in factor graphs are employed to facilitate describing the systems and developing low-complexity algorithms. Compared to the conventional CP-discarding schemes, the proposed CP-exploiting schemes have been shown to offer significant improvement in system performance while exhibiting very low complexity.

Firstly, we have proposed a low-complexity equalization algorithm for the iterative SC-FDE systems. With complexity of $\mathcal{O}(NlogN)$ per data block per iteration where $N$ is the length of the data block, a gain of 0.7dB is achieved under both the AWGN channel and the ISI channel when the CP ratio is $1/4$.

Secondly, we have proposed a low-complexity equalization algorithm for the turbo OFDM system. With complexity of $\mathcal{O}(NlogN)$ per data block per iteration, a gain of 0.97dB under the AWGN channel and 2dB under the ISI channels are achieved when the CP ratio is 1/4. We have provided an SNR analysis to verify the performance improvement of the proposed equalizer over the conventional equalizer. The sensitivity of the proposed low-complexity algorithm to channel estimation error is also examined.

Lastly, we have proposed a joint channel estimation, data detection and decoding approach in turbo OFDM systems by using the message passing EM algorithm. This approach achieves good channel estimation performance and its data detection performance is close to that with perfect channel information. We have accommodated both the means and variances of the channel estimate in the message passing EM algorithm. Different ways of using CP and different scheduling schemes have been thoroughly discussed. Using CP for equalization has been shown to be more beneficial than using it for channel estimation, and the proposed scheduling scheme is verified to be a novel tradeoff between complexity and performance.

## 5.2 Future Work

In Chapter 2 and Chapter 3, through exploiting CP, different gains have been obtained for the SC-FDE and OFDM systems. With the same CP ratio of 1/4 and both under the AWGN channel, the bound of gain 0.97dB has been achieved in OFDM system but not in SC-FDE system. Improvement on the proposed algorithm for it to achieve the same bound of gain in SC-FDE system is possible since in general SC-FDE system should produce the same performance (and therefore performance gain) as OFDM system under the AWGN channel.

For the ISI channel, a comparison between the mechanisms of SC-FDE and OFDM systems may be worthwhile in explaining the big difference in performance gains in the two systems (0.7dB for SC-FDE and 2dB for OFDM).

This thesis has focused on exploiting CP in the context of SISO systems. It is also possible to extend the proposed approach to MIMO systems. In fact, the proposed approach is of a general nature and can be applied to any other communication contexts where similar redundancy is present and exploitation of the redundancy is reasonable.

This thesis has demonstrated the performance improvement of the proposed approach over the conventional approach under certain system settings. Future work can be done to investigate the performance of the proposed approach when the system is with different ECC such as the LDPC code (which is stronger than the convolutional code this thesis has used), different mapping such as the set partitioning [88], or different channels such as the Porat and Freidlander channel [89].

# Appendix A

# A Fast Approach to Calculating $\mathbf{\Phi}_k$ in (3.42) and (3.43)

From the approximation (3.42), we have

$$\mathbf{\Phi}_k = \left(\mathbf{B}^H \tilde{\mathbf{H}}_k^H \tilde{\mathbf{H}}_k \mathbf{B}\right)_{diag} \tag{A.1}$$

with $\mathbf{B} = \mathbf{F}_G \mathbf{Q} \mathbf{A} \mathbf{F}_N^H$. As $\tilde{\mathbf{H}}_k^H \tilde{\mathbf{H}}_k$ is diagonal, the matrix $\mathbf{\Delta}_1 = \mathbf{F}_G^H \tilde{\mathbf{H}}_k^H \tilde{\mathbf{H}}_k \mathbf{F}_G$ is Hermitian circulant and its first column is given by

$$\mathbf{b} = \frac{1}{\sqrt{G}} \mathbf{F}_G^H \mathbf{a} \tag{A.2}$$

where the vector $\mathbf{a}$ contains all the diagonal entries of $\tilde{\mathbf{H}}_k^H \tilde{\mathbf{H}}_k$. Then we define matrix $\mathbf{\Delta}_2 = \mathbf{Q}^H \mathbf{\Delta}_1 \mathbf{Q}$, which is an Hermitian Toeplitz matrix

$$\mathbf{\Delta}_2 = \begin{bmatrix} b_0 & b_1^* & \cdots & b_{P-2}^* & b_{P-1}^* \\ b_1 & b_0 & \ddots & b_{P-3}^* & b_{P-2}^* \\ \vdots & b_1 & \cdots & \ddots & \vdots \\ b_{P-2} & \vdots & \ddots & \vdots & b_1^* \\ b_{P-1} & b_{P-2} & \cdots & b_1 & b_0 \end{bmatrix} \tag{A.3}$$

where the superscript $*$ denotes the conjugate operation and the first column $[b_0, b_1, ..., b_{P-1}]^T$ is the first $P$ entries of vector $\mathbf{b}$. Therefore, the matrix $\mathbf{\Delta}_3 = \mathbf{A}^H \mathbf{\Delta}_2 \mathbf{A}$ takes the following form

$$\mathbf{\Delta}_3 = \begin{bmatrix} \mathbf{0}_{(N-P)\times(N-P)} & \mathbf{0}_{(N-P)\times P} \\ \mathbf{0}_{P\times(N-P)} & \mathbf{\Delta}_2 \end{bmatrix}. \tag{A.4}$$

The $j$th diagonal entry of $\mathbf{\Phi}_k = \left(\mathbf{F}_N \mathbf{\Delta}_3 \mathbf{F}_N^H\right)_{diag}$ is calculated as

$$\begin{aligned} &\mathbf{f}_j^H \mathbf{\Delta}_3 \mathbf{f}_j \\ =&\frac{1}{N}\Big[Pb_0 + (P-1)b_1 e^{-2\pi i \frac{j}{N}} + (P-2)b_2 e^{-2\pi i \frac{2j}{N}} \\ &+... + b_{P-1} e^{-2\pi i \frac{(P-1)j}{N}} + (P-1)b_1^* e^{2\pi i \frac{j}{N}} \\ &+(P-2)b_2^* e^{2\pi i \frac{2j}{N}} + ... + b_{P-1}^* e^{2\pi i \frac{(P-1)j}{N}}\Big] \\ =&\frac{1}{N}\sum_{n=0}^{P-1}(P-n)b_n e^{-2\pi i \frac{nj}{N}} + \frac{1}{N}\sum_{n=1}^{P-1}(P-n)b_n^* e^{2\pi i \frac{nj}{N}} \end{aligned} \tag{A.5}$$

where $\mathbf{f}_j$ is the $j$th column of $\mathbf{F}_N^H$, $j = 0, ..., N-1$. The first term in (A.5) can be obtained by taking the FFT of the sequence $\boldsymbol{\eta}$ and the second term is just the conjugate of the first term, and

$$\boldsymbol{\eta} = \begin{cases} (P-n)b_n, & n = 0, ..., P-1 \\ \\ 0, & n = P, ..., N-1 \end{cases} \tag{A.6}$$

Thus the diagonal entries of $\mathbf{\Phi}_k$ can be calculated as

$$Diag\left(\mathbf{\Phi}_k\right) = \frac{1}{\sqrt{N}}\mathbf{F}_N\boldsymbol{\eta} + \left(\frac{1}{\sqrt{N}}\mathbf{F}_N\boldsymbol{\eta}\right)^* - \frac{P}{N}b_0^*\mathbf{I}_{N\times 1}. \tag{A.7}$$

The proposed approach of calculating $\mathbf{\Phi}_k$ only needs one $G$-point FFT for (A.2) and one $N$-point FFT for (A.7).

# Appendix B

# The Derivation of $\mathrm{Var}(\boldsymbol{\zeta}_k)_{diag}^{prop}$ in (3.50)

Using (3.48) and $\mathrm{E}(\boldsymbol{\zeta}_k) = \mathbf{0}$, the covariance matrix of $\boldsymbol{\zeta}_k$ is given by

$$
\begin{aligned}
\mathrm{Var}(\boldsymbol{\zeta}_k) =& \,\mathrm{E}(\boldsymbol{\zeta}_k\boldsymbol{\zeta}_k^H) \\
=& \mathbf{V}_k^{ext}\Big[\beta^{-1}\big(\mathbf{I}+\boldsymbol{\Theta}_k\big)\mathbf{H}_k^H\mathbf{H}_k\big(\mathbf{I}+\boldsymbol{\Theta}_k\big)^H + \boldsymbol{\Theta}_k\big(\overrightarrow{\mathbf{V}}_{a_k}^{-1}\big)^H\boldsymbol{\Theta}_k^H + \lambda_k^{-1}\boldsymbol{\Psi}_k\Big]\big(\mathbf{V}_k^{ext}\big)^H
\end{aligned}
\tag{B.1}
$$

where the variance matrices $\mathrm{Var}(\mathbf{n}_k) = \beta\mathbf{I}_N$, $\mathrm{Var}(\boldsymbol{\epsilon}_k) = \overrightarrow{\mathbf{V}}_{a_k}$ and $\mathrm{Var}(\boldsymbol{\omega}_k) = \lambda_k\mathbf{I}_G$ are used, and the cross-covariance matrices between any two of $\mathbf{n}_k$, $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\omega}_k$ are treated as zero since they are independent of each other. Meantime, we have $\boldsymbol{\Psi}_k = \boldsymbol{\Psi}_k^H$, $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}_k^H$, and $\big(\mathbf{V}_k^{ext}\big)^H = \mathbf{V}_k^{ext}$ and $\big(\overrightarrow{\mathbf{V}}_{a_k}^{-1}\big)^H = \overrightarrow{\mathbf{V}}_{a_k}^{-1}$. After simplification, (B.1) can be rewritten as

$$
\mathrm{Var}(\boldsymbol{\zeta}_k) = \mathbf{V}_k^{ext} + \mathbf{V}_k^{ext}\Big[\lambda_k^{-1}\boldsymbol{\Theta}_k\big(\boldsymbol{\Phi}_k - \boldsymbol{\Psi}_k\big) + \beta^{-1}\boldsymbol{\Theta}_k\mathbf{H}_k^H\mathbf{H}_k - \overrightarrow{\mathbf{V}}_{a_k}^{-1}\boldsymbol{\Theta}_k^H\Big]\mathbf{V}_k^{ext}. \tag{B.2}
$$

Since $\mathbf{V}_k^{ext}$ is diagonal and $\big(\boldsymbol{\Theta}_k\big)_{diag} = \mathbf{0}$ ($\boldsymbol{\Phi}_k$ and $\boldsymbol{\Psi}_k$ have the same diagonal entries), the two terms $\beta^{-1}\boldsymbol{\Theta}_k\mathbf{H}_k^H\mathbf{H}_k$ and $\overrightarrow{\mathbf{V}}_{a_k}^{-1}\boldsymbol{\Theta}_k^H$ have no contribution to the diagonal entries of $\mathrm{Var}(\boldsymbol{\zeta}_k)$ at all. After removing them, the following diagonal matrix is obtained.

$$\mathrm{Var}(\boldsymbol{\zeta}_k)_{diag} = \mathbf{V}_k^{ext} + \lambda_k^{-1}\mathbf{V}_k^{ext}\Big[\boldsymbol{\Theta}_k\big(\boldsymbol{\Phi}_k - \boldsymbol{\Psi}_k\big)\Big]_{diag}\mathbf{V}_k^{ext}. \tag{B.3}$$

Substituting (3.44) into (B.3), we have the diagonal entries of $\mathrm{Var}(\boldsymbol{\zeta}_k)$ for the proposed algorithm as (3.50).

# Appendix C

# The Derivation of $\overrightarrow{\mathbf{V}}_{\mathbf{g}_k}$ in (4.52) and $\overrightarrow{\mathbf{V}}_{\mathbf{g}'_k}$ in (4.53)

The covariance matrices of $\mathbf{g}_k$ and $\mathbf{g}'_k$ are derived as follows using (4.48), (4.49), (4.50) and (4.51). The covariance matrix of $\mathbf{g}_k$ is

$$\begin{aligned}
\overrightarrow{\mathbf{V}}_{\mathbf{g}_k} &= \mathrm{E}\left\{\mathbf{g}_k\mathbf{g}_k^H\right\} \\
&= \mathrm{E}\left\{N\mathbf{F}_N^H Diag\big(\mathbf{u}_k\big)\mathbf{F}_N\hat{\mathbf{S}}\,\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^H\hat{\mathbf{S}}^H\mathbf{F}_N^H Diag\big(\mathbf{u}_k\big)^H\mathbf{F}_N\right\} \\
&= \mathrm{E}\left\{N\mathbf{F}_N^H Diag\big(\mathbf{u}_k\big)\mathbf{F}_N\hat{\mathbf{S}}\,\mathbf{V}_{\hat{\mathbf{h}}_k}\hat{\mathbf{S}}^H\mathbf{F}_N^H Diag\big(\mathbf{u}_k\big)^H\mathbf{F}_N\right\} \\
&= N\mathbf{F}_N^H\,\mathrm{E}\left\{Diag\big(\mathbf{u}_k\big)\mathbf{F}_N\hat{\mathbf{S}}\,\mathbf{V}_{\hat{\mathbf{h}}_k}\hat{\mathbf{S}}^H\mathbf{F}_N^H Diag\big(\mathbf{u}_k\big)^H\right\}\mathbf{F}_N \\
&= N\mathbf{F}_N^H\,\boldsymbol{\Omega}\,\mathbf{F}_N
\end{aligned} \tag{C.1}$$

where

$$\boldsymbol{\Omega} = \mathrm{E}\left\{Diag\big(\mathbf{u}_k\big)\boldsymbol{\Theta}Diag\big(\mathbf{u}_k\big)^H\right\}, \tag{C.2}$$

$$\boldsymbol{\Theta} = \mathbf{F}_N\hat{\mathbf{S}}\,\mathbf{V}_{\hat{\mathbf{h}}_k}\hat{\mathbf{S}}^H\mathbf{F}_N^H. \tag{C.3}$$

The matrix $\boldsymbol{\Theta}$ is a circulant matrix with its diagonal entries $\theta_{p,p}$, $p = 0, ..., N-1$, given by

$$\theta_{p,p} = \mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right)/N. \tag{C.4}$$

The $(p, q)$th element of $\boldsymbol{\Omega}$, $p = 0, ..., N-1$, $q = 0, ..., N-1$, is given by

$$\begin{aligned}\omega_{p,q} &= \mathrm{E}\left(u_{k,p}\, \theta_{p,q}\, u_{k,q}^*\right) \\ &= \theta_{p,q}\, \mathrm{E}\left(u_{k,p}\, u_{k,q}^*\right)\end{aligned} \tag{C.5}$$

where $\theta_{p,q}$ is the $(p, q)$th element of $\boldsymbol{\Theta}$ and

$$\mathrm{E}\left(u_{k,p} u_{k,q}^*\right) = \begin{cases} 1, & \text{when } p = q \\ 0, & \text{when } p \neq q \end{cases} \tag{C.6}$$

since $\mathbf{V}_{\mathbf{u}_k} = \mathbf{I}_N$ as in (4.51). Therefore, we have

$$\boldsymbol{\Omega} = \frac{\mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right)}{N} \mathbf{I}_N, \tag{C.7}$$

and the covariance matrix of $\mathbf{g}_k$ is

$$\begin{aligned}\overrightarrow{\mathbf{V}}_{\mathbf{g}_k} &= N \mathbf{F}_N^H \frac{\mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right)}{N} \mathbf{I}_N \mathbf{F}_N \\ &= \mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right) \mathbf{I}_N\end{aligned}$$

as in (4.52).

Similarly, the covariance matrix of $\mathbf{g}'_k$ is

$$\begin{aligned}\overrightarrow{\mathbf{V}}_{\mathbf{g}'_k} &= \mathrm{E}\left\{\mathbf{g}'_k\mathbf{g}'^{H}_k\right\}\\ &= \mathrm{E}\left\{G\mathbf{F}_G^H Diag\big(\mathbf{B}\mathbf{u}_k\big)\mathbf{F}_G\check{\mathbf{S}}\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^H\check{\mathbf{S}}^H\mathbf{F}_G^H Diag\big(\mathbf{B}\mathbf{u}_k\big)^H\mathbf{F}_G\right\}\\ &= \mathrm{E}\left\{G\mathbf{F}_G^H Diag\big(\mathbf{B}\mathbf{u}_k\big)\mathbf{F}_G\check{\mathbf{S}}\mathbf{V}_{\hat{\mathbf{h}}_k}\check{\mathbf{S}}^H\mathbf{F}_G^H Diag\big(\mathbf{B}\mathbf{u}_k\big)^H\mathbf{F}_G\right\}\\ &= G\mathbf{F}_G^H\,\mathrm{E}\left\{Diag\big(\mathbf{B}\mathbf{u}_k\big)\mathbf{F}_G\check{\mathbf{S}}\mathbf{V}_{\hat{\mathbf{h}}_k}\check{\mathbf{S}}^H\mathbf{F}_G^H Diag\big(\mathbf{B}\mathbf{u}_k\big)^H\right\}\mathbf{F}_G\\ &= G\mathbf{F}_G^H\,\boldsymbol{\Omega}'\,\mathbf{F}_G \qquad (C.8)\end{aligned}$$

where

$$\boldsymbol{\Omega}' = \mathrm{E}\left\{Diag\big(\boldsymbol{\xi}\big)\boldsymbol{\Theta}' Diag\big(\boldsymbol{\xi}\big)^H\right\}, \qquad (C.9)$$

$$\boldsymbol{\Theta}' = \mathbf{F}_G\check{\mathbf{S}}\mathbf{V}_{\hat{\mathbf{h}}_k}\check{\mathbf{S}}^H\mathbf{F}_G^H, \qquad (C.10)$$

$$\boldsymbol{\xi} = Diag\big(\mathbf{B}\mathbf{u}_k\big) = Diag\big(\mathbf{F}_G\mathbf{Q}\mathbf{A}\mathbf{F}_N^H\mathbf{u}_k\big). \qquad (C.11)$$

The matrix $\boldsymbol{\Theta}'$ is a circulant matrix with its diagonal entries $\theta'_{p,p}$, $p = 0, ..., G-1$, given by

$$\theta'_{p,p} = \mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right)/G. \qquad (C.12)$$

The $(p,q)$th element of $\boldsymbol{\Omega}'$, $p = 0, ..., G-1$, $q = 0, ..., G-1$, is given by

$$\begin{aligned}\omega'_{p,q} &= \mathrm{E}\left(\xi_p\,\theta'_{p,q}\,\xi_q^*\right)\\ &= \theta'_{p,q}\,\mathrm{E}\left(\xi_p\,\xi_q^*\right) \qquad (C.13)\end{aligned}$$

where $\theta'_{p,q}$ is the $(p,q)$th element of $\mathbf{\Theta}'$,

$$\begin{aligned}
\mathrm{E}\left(\xi_p\,\xi_q^*\right) &= \mathrm{E}\left[\left(\mathbf{B}\mathbf{u}_k\right)_p \cdot \left(\mathbf{u}_k^H\mathbf{B}^H\right)_q\right] \\
&= \mathrm{E}\left(\mathbf{f}_p\mathbf{Q}\mathbf{A}\mathbf{F}_N^H\mathbf{u}_k\mathbf{u}_k^H\mathbf{F}_N\mathbf{A}^H\mathbf{Q}^H\mathbf{f}_q^H\right) \\
&= \mathbf{f}_p\mathbf{Q}\mathbf{A}\mathbf{F}_N^H\,\mathrm{E}\left(\mathbf{u}_k\mathbf{u}_k^H\right)\mathbf{F}_N\mathbf{A}^H\mathbf{Q}^H\mathbf{f}_q^H \\
&= \mathbf{f}_p\mathbf{Q}\mathbf{A}\mathbf{F}_N^H\mathbf{V}_{\mathbf{u}_k}\mathbf{F}_N\mathbf{A}^H\mathbf{Q}^H\mathbf{f}_q^H \\
&= \mathbf{f}_p\mathbf{Q}\mathbf{Q}^H\mathbf{f}_q^H \\
&= \frac{1}{G}\sum_{j=0}^{P-1} e^{-2\pi i\frac{j(p-q)}{G}},
\end{aligned} \tag{C.14}$$

and $\mathbf{f}_p$ and $\mathbf{f}_q$ are the $p$th and $q$th rows of the DFT matrix $\mathbf{F}_G$ respectively, $\mathbf{V}_{\mathbf{u}_k} = \mathbf{I}_N$ as in (4.51) and $\mathbf{A}\mathbf{A}^H = \mathbf{I}_P$. Specially, we have

$$\mathrm{E}\left(\xi_p\,\xi_q^*\right) = \frac{P}{G}, \quad \text{when } p = q. \tag{C.15}$$

When $p \neq q$, $\mathrm{E}\left(\xi_p\,\xi_q^*\right)$ is much smaller than $P/G$ (see (C.14)). Hence, we can approximate $\mathbf{\Omega}'$ by its diagonal matrix as

$$\mathbf{\Omega}' \approx \frac{\mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right)}{G} \cdot \frac{P}{G}\mathbf{I}_G, \tag{C.16}$$

and the covariance matrix of $\mathbf{g}'_k$ can be given by

$$\begin{aligned}
\overrightarrow{\mathbf{V}}_{\mathbf{g}'_k} &\approx G\mathbf{F}_G^H\frac{\mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right)}{G} \cdot \frac{P}{G}\mathbf{I}_G\,\mathbf{F}_G \\
&\approx \frac{P}{G}\,\mathrm{tr}\left(\mathbf{V}_{\hat{\mathbf{h}}_k}\right)\mathbf{I}_G
\end{aligned}$$

as in (4.53).

# Bibliography

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

[2] D. Tse and P. Viswanath, *Fundamentals of Wireless communication.* Cambridge university press, 2005.

[3] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall, 1993.

[4] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding.* John Wiley & Sons, 2002.

[5] R. Koetter, A. C. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Processing Mag.*, vol. 21, no. 1, pp. 67–80, 2004.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *IEEE International Conference on Communications (ICC), 1993*, vol. 2, pp. 1064–1070.

[7] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *Proc. Int. Symp. Turbo Codes*, 1997, pp. 96–102.

[8] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Proc. International Symposium on Turbo Codes and Related Topics*, 1997, pp. 1–11.

[9] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 673–683, 2002.

[10] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, 2002.

[11] M. Tüchler and A. Singer, "Turbo equalization: An overview," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 920–952, 2011.

[12] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, A. Glavieux *et al.*, "Iterative correction of intersymbol interference: Turbo-equalization," *Eur. Trans. Telecomm.*, vol. 6, no. 5, pp. 507–511, 1995.

[13] G. Bauch, H. Khorram, and J. Hagenauer, "Iterative equalization and decoding in mobile communications systems," *3rd ITG-Fachbericht Mobile Kommunikation*, pp. 307–312, 1997.

[14] J. Hagenauer and P. Hoeher, "A viterbi algorithm with soft-decision outputs and its applications," in *IEEE Global Telecommunications Conference (GLOBECOM), 1989.* IEEE, 1989, pp. 1680–1686.

[15] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, 1996.

[16] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[17] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[18] S. A. Abrantes, "From bcjr to turbo decoding: Map algorithms made easier," 2004.

[19] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE International Conference on Communications(ICC), 1995*, vol. 2. IEEE, 1995, pp. 1009–1013.

[20] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *Eur. Trans. Telecomm.*, vol. 8, no. 2, pp. 119–125, 1997.

[21] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, Prentice Hall, 1983.

[22] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," *TDA Progress report*, vol. 42, no. 122, pp. 56–65, 1995.

[23] F. Gray, "Pulse code communication," US Patent 2 632 058, Mar. 17, 1953.

[24] A. Hero and T. Marzetta, "Cutoff rate and signal design for the quasi-static rayleigh-fading space-time channel," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2400–2416, 2001.

[25] Q. Guo and D. Huang, "A concise representation for the soft-in soft-out LMMSE detector," *IEEE Commun. Lett.*, vol. 15, no. 99, pp. 1–3, 2011.

[26] S. L. Lauritzen, *Graphical Models.* Oxford University Press, 1996.

[27] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[28] G. D. Forney Jr, “Codes on graphs: Normal realizations,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 520–548, 2001.

[29] H. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, “The factor graph approach to model-based signal processing,” *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[30] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[31] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inform. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[32] N. Wiberg, *Codes and decoding on general graphs.* Ph.D. dissertation, Linköping University, Linköping, Sweden, 1996.

[33] N. Wiberg, H.-A. Loeliger, and R. Kotter, “Codes and iterative decoding on general graphs,” *Eur. Trans. Telecomm.*, vol. 6, no. 5, pp. 513–525, 1995.

[34] J. Dauwels, A. Eckford, S. Korl, and H.-A. Loeliger, “Expectation maximization as message passing-Part I: Principles and Gaussian messages,” *arXiv preprint arXiv: 0910.2832.*

[35] G. Colavolpe and G. Germi, “On the application of factor graphs and the sum-product algorithm to ISI channels,” *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 818–825, 2005.

[36] H. Sari, G. Karam, and I. Jeanclaude, “Transmission techniques for digital terrestrial tv broadcasting,” *IEEE Commun. Mag.*, vol. 33, no. 2, pp. 100–109, 1995.

[37] D. Falconer, S. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, “Frequency domain equalization for single-carrier broadband wireless systems,” *IEEE Commun. Mag.*, vol. 40, no. 4, pp. 58–66, 2002.

[38] F. Pancaldi, G. Vitetta, R. Kalbasi, N. Al-Dhahir, M. Uysal, and H. Mheidat, “Single-carrier frequency domain equalization,” *IEEE Signal Processing Mag.*, vol. 25, no. 5, pp. 37–56, 2008.

[39] H. Rohling, *OFDM: Concepts for Future Communication Systems.* Springer, 2011.

[40] H. G. Myung, J. Lim, and D. J. Goodman, “Single carrier fdma for uplink wireless transmission,” *IEEE Veh. Technol. Mag.*, vol. 1, no. 3, pp. 30–38, 2006.

[41] H. Ekstrom, A. Furuskar, J. Karlsson, M. Meyer, S. Parkvall, J. Torsner, and M. Wahlqvist, “Technical solutions for the 3g long-term evolution,” *IEEE Commun. Mag.*, vol. 44, no. 3, pp. 38–45, 2006.

[42] *3rd Generation Partnership Project: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation*, European Telecommunications Standards Institute TS 36.211, 2012.

[43] *Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers*, European Telecommunications Standards Institute ETS 300 401, Feb. 1995.

[44] U. Reimers, “DVB-T: the COFDM-based system for terrestrial television,” *Electronics & Communication Engineering Journal*, vol. 9, no. 1, pp. 28–32, 1997.

[45] A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P. Karlsson, “A comparison of the HIPERLAN/2 and IEEE 802.11a wireless LAN standards,” *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 172–180, 2002.

[46] *Multimedia mobile access communication systems (MMAC)*, Std. [Online]. Available: http://www.arib.or.jp/mmac/e/index.htm

[47] J. Khun-Jush, P. Schramm, U. Wachsmann, and F. Wenger, “Structure and performance of the HIPERLAN/2 physical layer,” in *IEEE Vehicular Technology Conference (VTC), 1999*, vol. 5, pp. 2667–2671.

[48] B. Devillers, J. Louveaux, and L. Vandendorpe, “Exploiting cyclic prefix for performance improvement in single carrier systems,” in *IEEE 7th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2006*, pp. 1–5.

[49] G. Bottomley and L. Wilhelmsson, “Recovering signal energy from the cyclic prefix in OFDM,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 3205–3211, 2008.

[50] A. Tarighat and A. Sayed, “An optimum OFDM receiver exploiting cyclic prefix for improved data estimation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2003*, vol. 4, pp. IV–217.

[51] L. Vangelista, M. Rotoloni, and A. Morello, “Improved data detection exploiting full cyclic prefix for the evolution of DVB-T,” in *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), 2008*, pp. 1070–1074.

[52] K. Hueske and J. Götze, “Improving OFDM data estimation by overlapping based cyclic prefix reuse,” in *14th International OFDM Workshop (InOWo), 2009*, Hamburg, Germany.

[53] G. Garbo and S. Mangione, “An OFDM receiver exploiting multipath diversity,” in *IEEE Mosharaka International Conference on Communications, Computers and Applications (MIC-CCA), 2008*, pp. 65–70.

[54] G. Garbo and S. Mangione, “An improved detection technique for cyclic-prefixed OFDM,” *Journal of Networks*, vol. 5, no. 7, pp. 759–765, 2010.

[55] G. Garbo and S. Mangione, “An improved receiver architecture for cyclic-prefixed OFDM,” in *IEEE Wireless Communications and Networking Conference (WCNC), 2009*, pp. 1–6.

[56] A. Quadeer, “Enhanced equalization in OFDM systems using cyclic prefix,” in *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS), 2010*, pp. 40–44.

[57] T. Al-Naffouri and A. Quadeer, “Blind maximum-likelihood data recovery in OFDM,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2008*, pp. 2829–2832.

[58] T. Al-Naffouri and A. Quadeer, “Cyclic prefix based enhanced data recovery in OFDM,” *IEEE Trans. Signal Processing*, vol. 58, no. 6, pp. 3406–3410, 2010.

[59] T. Palenik and P. Farkas, “Exploiting redundancy in an OFDM SDR receiver,” *International Journal of Digital Multimedia Broadcasting*, vol. 2009, 2009.

[60] T. Palenik and P. Farkas, “Exploiting cyclic prefix redundancy in OFDM to improve performance of Tanner: graph based decoding,” *Analog Integrated Circuits and Signal Processing*, vol. 69, no. 2, pp. 143–152, 2011.

[61] X. Wang and K. Liu, “Performance analysis for adaptive channel estimation exploiting cyclic prefix in multicarrier modulation systems,” *IEEE Trans. Commun.*, vol. 51, no. 1, pp. 94–105, 2003.

[62] G. Al-Rawi, T. Al-Naffouri, A. Bahai, and J. Cioffi, “Exploiting error-control coding and cyclic-prefix in channel estimation for coded OFDM systems,” *IEEE Commun. Lett.*, vol. 7, no. 8, pp. 388–390, 2003.

[63] G. Al-Rawi, T. Al-Naffouri, A. Bahai, and J. Cioffi, “An iterative receiver for coded OFDM systems over time-varying wireless channels,” in *IEEE International Conference on Communications (ICC), 2003*, vol. 5, pp. 3371–3376.

[64] T. Al-Naffouri, "An EM-based forward-backward Kalman filter for the estimation of time-variant channels in OFDM," *IEEE Trans. Signal Processing*, vol. 55, no. 7, pp. 3924–3930, 2007.

[65] A. A. Quadeer and M. S. Sohail, "Enhanced channel estimation using cyclic prefix in MIMO STBC OFDM sytems," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2010*, pp. 277–282.

[66] J. Van de Beek, M. Sandell, and P. Borjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Trans. Signal Processing*, vol. 45, no. 7, pp. 1800–1805, 1997.

[67] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," in *Communication Theory Mini-Conference, 1999*, pp. 106–109.

[68] U. Tureli, H. Liu, and M. Zoltowski, "OFDM blind carrier offset estimation: ESPRIT," *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1459–1461, 2000.

[69] J. Yang, Q. Guo, D. Huang, and S. Nordholm, "Exploiting cyclic prefix in Turbo FDE systems using factor graph," in *IEEE Wireless Communications and Networking Conference (WCNC), 2013*, pp. 2536–2541.

[70] J. Yang, Q. Guo, D. Huang, and S. Nordholm, "Enhanced data detection in OFDM systems using factor graph," in *IEEE International Conference on Wireless Communications and Signal Processing (WCSP), 2013*, pp. 1–5.

[71] J. Yang, Q. Guo, D. D. Huang, and S. Nordholm, "A factor graph approach to exploiting cyclic prefix for equalization in OFDM systems," *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 4972–4983, 2013.

[72] J. Yang, Q. Guo, D. D. Huang, and S. Nordholm, "Exploiting cyclic prefix for joint detection, decoding and channel estimation in OFDM via EM algorithm and message passing," in *IEEE International Conference on Communications (ICC), 2014*, accepted.

[73] J. Yang, Q. Guo, D. D. Huang, and S. Nordholm, “Exploiting cyclic prefix for iterative OFDM receiver design via message passing based EM algorithm,” *IEEE Trans. Commun.*, submitted.

[74] B. Ng, C. Lam, and D. Falconer, “Turbo frequency domain equalization for single-carrier broadband wireless systems,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 759–767, 2007.

[75] T.-D. Chiueh, P.-Y. Tsai, and I.-W. Lai, *Baseband Receiver Design for Wireless MIMO-OFDM Communications*, 2nd ed. Wiley, 2012.

[76] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*, IEEE 802.11a Std., Sept. 1999.

[77] V. Tarokh, A. Naguib, N. Seshadri, and A. Calderbank, “Space-time codes for high data rate wireless communication: performance criteria in the presence of channel estimation errors, mobility, and multiple paths,” *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 199–207, 1999.

[78] S. Y. Park, Y. G. Kim, and C. G. Kang, “Iterative receiver for joint detection and channel estimation in OFDM systems under mobile radio channels,” *IEEE Trans. Veh. Technol.*, vol. 53, no. 2, pp. 450–460, 2004.

[79] M. Zhao, Z. Shi, and M. Reed, “Iterative turbo channel estimation for OFDM system over rapid dispersive fading channel,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 3174–3184, 2008.

[80] S. Song, A. C. Singer, and K.-M. Sung, “Soft input channel estimation for turbo equalization,” *IEEE Trans. Signal Processing*, vol. 52, no. 10, pp. 2885–2894, 2004.

[81] Q. Guo and D. Huang, “EM-based joint channel estimation and detection for frequency selective channels using Gaussian message passing,” *IEEE Trans. Signal Processing*, vol. 59, no. 8, pp. 4030–4035, 2011.

[82] P. Bello, “Characterization of randomly time-variant linear channels,” *IEEE Trans. Commun.*, vol. 11, no. 4, pp. 360–393, 1963.

[83] Q. Guo and D. Huang, “A frequency domain state-space approach to LS estimation and its application in turbo equalization,” *IEEE Trans. Signal Processing*, vol. 59, no. 7, pp. 3288–3300, 2011.

[84] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* John Wiley & Sons, 2007.

[85] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 3rd ed. John Wiley & Sons, 2011.

[86] D. Chu, “Polyphase codes with good periodic correlation properties,” *IEEE Trans. Inform. Theory*, vol. 18, no. 4, pp. 531–532, 1972.

[87] N. Chen and G. T. Zhou, “Superimposed training for OFDM: a peak-to-average power ratio analysis,” *IEEE Trans. Signal Processing*, vol. 54, no. 6, pp. 2277–2287, 2006.

[88] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inform. Theory*, vol. 28, no. 1, pp. 55–67, 1982.

[89] B. Porat and B. Friedlander, “Blind equalization of digital communication channels using high-order moments,” *IEEE Trans. Signal Processing*, vol. 39, no. 2, pp. 522–526, 1991.