University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2008

LMMSE turbo equalization based on factor graphs

Qinghua Guo

City University of Hong Kong, qguo@uow.edu.au

Li Ping

City University of Hong Kong

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

LMMSE turbo equalization based on factor graphs

Abstract

In this paper, a vector-form factor graph representation is derived for intersymbol interference (ISI) channels. The resultant graphs have a tree-structure that avoids the short cycle problem in existing graph approaches. Based on a joint Gaussian approximation, we establish a connection between the LLR (log-likelihood ratio) estimator for a linear system driven by binary inputs and the LMMSE (linear minimum mean-square error) estimator for a linear system driven by Gaussian inputs. This connection facilitates the application of the recently proposed Gaussian message passing technique to the cycle-free graphs for ISI channels. We also show the equivalence between the proposed approach and the Wang-Poor approach based on the LMMSE principle. An attractive advantage of the proposed approach is its intrinsic parallel structure. Simulation results are provided to demonstrate this property. © 2008 IEEE.

Keywords

turbo, equalization, factor, lmmse, graphs

Disciplines

Engineering | Science and Technology Studies

Publication Details

Q. Guo & L. Ping, "LMMSE turbo equalization based on factor graphs," IEEE Journal on Selected Areas in Communications, vol. 26, (2) pp. 311-319, 2008.

LMMSE Turbo Equalization Based on Factor Graphs

Qinghua Guo, *Student Member, IEEE* and Li Ping, *Senior Member, IEEE*

Abstract—In this paper, a vector-form factor graph representation is derived for intersymbol interference (ISI) channels. The resultant graphs have a tree-structure that avoids the short cycle problem in existing graph approaches. Based on a joint Gaussian approximation, we establish a connection between the LLR (log-likelihood ratio) estimator for a linear system driven by binary inputs and the LMMSE (linear minimum mean-square error) estimator for a linear system driven by Gaussian inputs. This connection facilitates the application of the recently proposed Gaussian message passing technique to the cycle-free graphs for ISI channels. We also show the equivalence between the proposed approach and the Wang-Poor approach based on the LMMSE principle. An attractive advantage of the proposed approach is its intrinsic parallel structure. Simulation results are provided to demonstrate this property.

Index Terms—Turbo equalization, intersymbol interference, factor graphs, Gaussian message passing, linear MMSE.

I. INTRODUCTION

TURBO equalization [1]–[8] is an effective technique to alleviate intersymbol interference (ISI) in multi-path channels. A turbo equalizer consists of two basic processors: a soft-in soft-out channel equalizer and an *a posteriori* probability (APP) channel decoder. The two processors operate in an iterative manner.

The optimal realization of the channel equalizer is based on the maximum *a posteriori* probability (MAP) algorithm [1]–[3] that has exponential complexity. Low-complexity alternatives based on the linear minimum mean-square error (LMMSE) principle [5]–[9] provide a good trade-off between performance and complexity. The LMMSE approaches involve solving a large matrix equation, which can still be very costly. A common solution is to perform estimation based on truncated observations using a sliding or extending window [6]–[9]. The window size must be large enough to avoid performance loss.

Inspired by the success of low-density parity-check (LDPC) decoding [11], factor graph techniques [14]–[16] have been investigated for channel equalization. An ISI channel can be represented by a factor graph in a straightforward form [13] (see Fig. 2 below) to which the sum-product algorithm can be applied. This approach has two difficulties. First, its

complexity grows exponentially with the number of non-zero channel coefficients. Second, except for some special cases (e.g., in the case of sparse ISI channels [13]), the presence of short cycles can cause noticeable performance loss. The stretching technique can be used to relieve the short cycle problem [13], but high complexity is still a concern.

In this paper, a vector-form graph representation is derived for ISI channels. The resultant graphs have a tree-structure that avoids the short cycle problem mentioned earlier in existing graph approaches. Based on a joint Gaussian approximation, we establish a connection between the LLR (log-likelihood ratio) estimator for a linear system driven by binary inputs and the LMMSE estimator for a linear system driven by Gaussian inputs. This connection facilitates the application of the recently proposed Gaussian message passing (GMP) technique [16]–[18], [25] to the tree-structured graphs for ISI channels. We also show the equivalence between the proposed approach and the Wang-Poor approach [9] based on the LMMSE principle. (Thus, the proposed approach provides an efficient graph implementation for the Wang-Poor approach in ISI channels without resorting to the windowing techniques as in [6]–[9].) An attractive advantage of this approach is its intrinsic parallel structure. Simulation results are provided to demonstrate this advantage.

The notations used in this paper are as follows. Lower case letters (e.g., x) denote scalars, bold lower case letters (e.g., $\mathbf{x}$) denote column vectors, and bold upper case letters (e.g., $\mathbf{X}$) denote matrices. The superscript T denotes the transpose operation. The symbol $\mathbf{I}$ denotes an identity matrix with proper size. The letters m , v , and $\mathbf{m}$, $\mathbf{V}$ denote the means and variances of scalar and vector random variables, respectively. For example, m_x and v_x denote the mean and variance of the scalar random variable x ; $\mathbf{m}_x$ and $\mathbf{V}_x$ represent the mean vector and covariance matrix of the vector random variable $\mathbf{x}$. Matrix $\mathbf{W}$ (scalar w) is defined as $\mathbf{W} = \mathbf{V}^{-1}$ ($w = 1/v$) [16].

II. PRINCIPLES OF TURBO EQUALIZATION

Consider a coded linear system characterized by the following equation (see the upper part of Fig. 1)

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1)$$

where $\mathbf{r} = [r_0, r_1, \dots, r_{N-1}]^T$ is a length- N observation vector, $\mathbf{H}$ an $N \times J$ system transfer matrix (usually representing the multiplicative effect of the channel), $\mathbf{x} = [x_0, x_1, \dots, x_{J-1}]^T$ a length- J transmitted sequence formed by the outputs of a forward error correction (FEC) encoder with binary phase shift keying (BPSK) modulation, and $\mathbf{n}$ an additive white Gaussian noise (AWGN) vector with zero mean and covariance matrix

Manuscript received April 2, 2007; revised October 26, 2007. This work was fully supported by a grant from the Research Grant Council of the Hong Kong SAR, China, under Project CityU 117305. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Adelaide, Australia, September 4–9, 2005.

The authors are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR (e-mail: 50007550@student.cityu.edu.hk; eeliping@cityu.edu.hk).

Digital Object Identifier 10.1109/JSAC.2008.080208.

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

$\sigma^2 \mathbf{I}$. We assume that the coded bit sequence is permuted by an interleaver Π before transmission. The system under consideration is “binary” in the sense that the elements of $\mathbf{x}$ are binary, i.e., $x_j \in \{+1, -1\}$, $j = 0, 1, \dots, J-1$. Although we only discuss the real system here, our discussion can be extended to a system with quadrature phase shift keying (QPSK) signaling and a complex channel matrix, since a complex matrix equation can always be represented by an equivalent real one by equating the real and imaginary parts separately.

A. The Iterative Equalization Process

A turbo receiver is shown in the lower part of Fig. 1. It consists of a channel equalizer and a decoder interconnected by an interleaver Π and the corresponding deinterleaver Π^{-1} . The following is a brief outline of the turbo detection principle (refer to [6] for details).

As shown in Fig.1, the extrinsic LLR about a coded bit x_j generated by the equalizer is the difference between its *a posteriori* LLR and *a priori* LLR

$$L_{Equ}(x_j) = \ln \frac{\Pr(x_j = +1|\mathbf{r})}{\Pr(x_j = -1|\mathbf{r})} - \ln \frac{\Pr(x_j = +1)}{\Pr(x_j = -1)}, \quad j = 0, \dots, J-1. \quad (2)$$

It is computed based on the channel observations and *a priori* information about x_i ($i = 0, 1, \dots, j-1, j+1, \dots, J-1$). Notice that the *a priori* information about x_j itself is excluded here and hence the term “extrinsic” [19]. Similarly, we define $L_{Dec}(x_j)$ as the extrinsic LLR generated by the decoder [19]. The two sets of LLRs defined above are computed following the iterative process outlined below.

- Based on the *a priori* information $\{L_{Dec}(x_j), \forall j\}$ from the decoder, the channel equalizer computes $\{L_{Equ}(x_j), \forall j\}$ ignoring the FEC coding constraint (i.e., as if $\{x_j\}$ are un-coded bits). For the initial step, there is no *a priori* information available to the channel equalizer. Hence, we set $L_{Dec}(x_j) = 0, \forall j$.
- Using $\{L_{Equ}(x_j)\}$ as inputs, the decoder computes $\{L_{Dec}(x_j)\}$ based on the FEC coding constraint. Then, $\{L_{Dec}(x_j)\}$ are fed back to the channel equalizer for the next iteration.
- The above process continues iteratively. The decoder makes hard decisions on the information bits during the final iteration.

The realization of the decoder involves a standard APP decoding. In what follows, we will concentrate on the realization of the channel equalizer.

B. Realization of the Channel Equalizer

The MAP Approach: The optimal realization of the channel equalizer is based on the MAP principle (which can be realized using the BCJR algorithm [12], [24]). The complexity grows exponentially with the channel length.

The Factor Graph Approach in [13]: In this approach, an ISI channel is represented by a factor graph (that generally contains short cycles) in a straightforward way (see Fig. 2). The sum-product algorithm is applied to the graph. The complexity of this approach increases exponentially with the number of non-zero channel coefficients. This approach is most useful in the case of sparse ISI channels.

The Wang-Poor Approach: The Wang-Poor approach was derived in [9] for multi-user detection and later studied for turbo equalization in [6] and [7]. It provides a good trade-off between performance and complexity. Since this approach is based on the LMMSE principle, it is also widely referred to as an LMMSE approach.

The Joint Gaussian (JG) Approach [10]: In this approach, equation (1) is rewritten as

$$\mathbf{r} = \mathbf{h}_j x_j + \boldsymbol{\xi}_j, \quad (3)$$

where x_j is the concerned coded bit, $\mathbf{h}_j$ is the j th column of $\mathbf{H}$, and

$$\boldsymbol{\xi}_j = \sum_{i \neq j} \mathbf{h}_i x_i + \mathbf{n} \quad (4)$$

representing the noise-plus-interference component with respect to x_j . The basic assumption of the JG approach is that the entries of $\boldsymbol{\xi}_j$ are jointly Gaussian (according to the central limit theorem) with mean vector and covariance matrix given by

$$\mathbf{m}_{\boldsymbol{\xi}_j} = \mathbf{H} \mathbf{m}_{\mathbf{x}}^{prio} - \mathbf{h}_j m_{x_j}^{prio}, \quad (5)$$

$$\mathbf{V}_{\boldsymbol{\xi}_j} = \mathbf{H} \mathbf{V}_{\mathbf{x}_{(j)}} \mathbf{H}^T + \sigma^2 \mathbf{I}, \quad (6)$$

where $\mathbf{m}_{\mathbf{x}}^{prio} = [m_{x_0}^{prio}, \dots, m_{x_{J-1}}^{prio}]^T$, $\mathbf{x}_{(j)} = [x_0, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_{J-1}]^T$, $\mathbf{V}_{\mathbf{x}_{(j)}} = \text{diag}[v_{x_0}^{prio}, \dots, v_{x_{j-1}}^{prio}, 0, v_{x_{j+1}}^{prio}, \dots, v_{x_{J-1}}^{prio}]$, and the superscript “prio” means “*a priori*”. The *a priori* information is computed based on the outputs of the decoder as follows

$$\begin{aligned} m_{x_j}^{prio} &= \frac{\Pr(x_j = +1) - \Pr(x_j = -1)}{\Pr(x_j = +1) + \Pr(x_j = -1)} \\ &= \frac{\exp(L_{Dec}(x_j)) - 1}{\exp(L_{Dec}(x_j)) + 1}, \end{aligned} \quad (7)$$

$$v_{x_j}^{prio} = E[(x_j - m_{x_j}^{prio})^2] = 1 - (m_{x_j}^{prio})^2. \quad (8)$$

Based on the joint Gaussian assumption, the extrinsic LLR about x_j can be computed as

$$\begin{aligned} L_{Equ}^{JG}(x_j) &= \ln \frac{p(\mathbf{r}|x_j = +1)}{p(\mathbf{r}|x_j = -1)} \\ &= \ln \frac{\exp\left[-\frac{1}{2}(\mathbf{r} - \mathbf{h}_j - \mathbf{m}_{\boldsymbol{\xi}_j})^T \mathbf{V}_{\boldsymbol{\xi}_j}^{-1} (\mathbf{r} - \mathbf{h}_j - \mathbf{m}_{\boldsymbol{\xi}_j})\right]}{\exp\left[-\frac{1}{2}(\mathbf{r} + \mathbf{h}_j - \mathbf{m}_{\boldsymbol{\xi}_j})^T \mathbf{V}_{\boldsymbol{\xi}_j}^{-1} (\mathbf{r} + \mathbf{h}_j - \mathbf{m}_{\boldsymbol{\xi}_j})\right]} \\ &= 2\mathbf{h}_j^T \mathbf{V}_{\boldsymbol{\xi}_j}^{-1} (\mathbf{r} - \mathbf{H} \mathbf{m}_{\mathbf{x}}^{prio} + \mathbf{h}_j m_{x_j}^{prio}). \end{aligned} \quad (9)$$

Fig. 2. Factor graph for an ISI channel with memory length 2.

Both the JG approach outlined above and the Wang-Poor approach studied in [6], [7] and [9] involve Gaussian approximations of the noise-plus-interference component with respect to a particular transmitted coded bit. These two approaches have the following difference in their derivations.

- In the JG approach, the joint Gaussian approximation is applied directly to the channel output in (3), which results in a more concise and straightforward derivation (e.g., compared with the derivation in [9]).
- In the Wang-Poor approach, the Gaussian approximation is applied after the LMMSE filtering of the channel output. It assumes a scalar Gaussian distribution for the residual distortion after LMMSE filtering.

We may expect that the Wang-Poor approach and the JG approach are equivalent. This is indeed the case, as proved in Appendix A.

Direct evaluation of (9) involves matrix inversion. The related complexity can be quite high if the size of $\mathbf{H}$ is large. The approaches with sliding or extending window have been discussed in [6]-[9] to alleviate the complexity problem. In what follows, we will present an alternative graph-based technique.

C. The Connection between the Extrinsic LLR $L_{Equ}^{JG}(x_j)$ and the LMMSE Estimate

Definition 1: The Gaussian companion of the binary system (1) is defined as

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (10)$$

where $\{x_j\}$ (the entries of $\mathbf{x}$) are independent Gaussian random variables with the same *a priori* means and variances as their binary counterparts in (1).

We emphasize that $\{x_j\}$ are binary in (1) and Gaussian in (10). Given the *a priori* information $\{m_{x_j}^{prio}\}$, $\{v_{x_j}^{prio}\}$ and the observation vector $\mathbf{r}$ in (10), we can compute the *a posteriori* information $\{m_{x_j}^{post}\}$ and $\{v_{x_j}^{post}\}$ based on the MMSE principle [20].

The proposition below establishes a key connection between the extrinsic LLR $L_{Equ}^{JG}(x_j)$ defined in (9) for system (1) driven by binary inputs and the LMMSE estimation for (10) driven by Gaussian inputs. The proof of the proposition is

given in Appendix B. (Note that the LMMSE estimation discussed below is for a Gaussian input system. It should be distinguished from the Wang-Poor approach for a binary input system.)

Proposition 1: Assume that we have the same observation vector $\mathbf{r}$ for both (1) and (10). Then

$$L_{Equ}^{JG}(x_j) = 2 \left(\frac{m_{x_j}^{post}}{v_{x_j}^{post}} - \frac{m_{x_j}^{prio}}{v_{x_j}^{prio}} \right), \quad (11)$$

where $L_{Equ}^{JG}(x_j)$ defined in (9) is computed based on (1), and the *a posteriori* mean and variance are the computed based on (10).

Equation (11) suggests an alternative approach to the channel equalizer, i.e., first performing the LMMSE estimation for the Gaussian companion (10) to get $\{m_{x_j}^{post}\}$ and $\{v_{x_j}^{post}\}$, and then generating the extrinsic LLRs $\{L_{Equ}^{JG}(x_j)\}$ using (11). This offers an efficient implementation technique, since $\{m_{x_j}^{post}\}$ and $\{v_{x_j}^{post}\}$ in the Gaussian companion can be found efficiently using the recently proposed GMP techniques [16]-[18], [25], as detailed below.

III. TURBO EQUALIZATION BASED ON GMP

GMP is a graph technique to perform LMMSE estimation for a linear Gaussian system such as the one described in (10). This approach allows the use of message computation rules for “local” computations corresponding to the building blocks of the system model. The underlying assumption in this approach is that all the variables involved have Gaussian distributions that can be characterized by their means and variances. The *a posteriori* means and variances of $\{x_j\}$ can be found by processing the messages (means and variances) over a graph for (10).

In the following, we will consider some details in applying the GMP technique to channel equalization.

A. Factor Graph Representation of an ISI Channel

Consider the system model in (10), where $\mathbf{H}$ represents the ISI channel effect. Although we focus on time invariant ISI channels for simplicity in this paper, the extension of our discussion to time variant ISI channels is straightforward. Let the number of channel taps be $L = M + 1$ (i.e., the channel memory length is M), and $\mathbf{h} = [h_M, h_{M-1}, \dots, h_0]^T$ be the coefficient vector of an ISI channel. The system described by (10) can be rewritten as

$$\begin{bmatrix} r_0 \\ \vdots \\ r_j \\ \vdots \\ r_{J+M-1} \end{bmatrix} = \begin{bmatrix} h_0 & & & \\ & \ddots & & \\ & & h_0 & \\ h_M & \vdots & \ddots & \\ & h_M & & h_0 \\ & & \ddots & \vdots \\ & & & h_M \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_j \\ \vdots \\ x_{J-1} \end{bmatrix} + \begin{bmatrix} n_0 \\ \vdots \\ n_j \\ \vdots \\ n_{J+M-1} \end{bmatrix}. \quad (12)$$

The $(j-1)$ th entry of $\mathbf{r}$ in the above equation is given by

$$\begin{aligned} r_{j-1} &= h_M x_{j-1-M} + h_{M-1} x_{j-M} + \dots + h_0 x_{j-1} + n_{j-1} \\ &= \mathbf{h}^T \mathbf{x}_{j-1} + n_{j-1}, \end{aligned} \quad (13)$$

Fig. 3. Cycle-free factor graph for an ISI channel.

where

$$\underline{x}_j = [x_{j-M}, x_{j-M+1}, \dots, x_j]^T, \quad (14)$$

and $x_{j-m} = 0$ if $j - m < 0$.

We use Forney-style factor graphs [14], [16] where edges represent variables and boxes factors. The factor graph for an ISI channel with $M = 2$ in a straightforward form is shown in Fig. 2, where the function $g_j(x_{j-M}, \dots, x_j)$ is the transition probability of the channel (see (3) in [13]). We can clearly see the presence of short cycles in this graph. This problem can be alleviated when ISI channels are sparse [13]. However, short cycles constitute a major concern when the message passing algorithm is applied to this graph for a general ISI channel.

We now consider an alternative factor graph representation. Define

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_M \\ 0 & \mathbf{0}^T \end{bmatrix}_{L \times L} \text{ and } \mathbf{f} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}_L, \quad (15)$$

where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix and $\mathbf{0}$ a zero column vector with length M . The matrix $\mathbf{G}$ can be decomposed into

$$\mathbf{G} = \mathbf{G}'' \mathbf{G}', \quad (16)$$

where $\mathbf{G}'' = [\mathbf{I}_M \quad \mathbf{0}]^T$ and $\mathbf{G}' = [\mathbf{0} \quad \mathbf{I}_M]$. It is easy to see that

$$\underline{x}_j = \underline{y}_j + \mathbf{f} x_j, \quad (17)$$

where

$$\underline{y}_j = \mathbf{G} \underline{x}_{j-1}. \quad (18)$$

The factor graph of (13), (17) and (18) is shown in the dashed box in Fig. 3. It is used as a building block to form a complete graph of (12) in Fig. 3. We can see that the graph has a tree-structure, in which the short cycles are “absorbed” into the vector-based representation. The exact (Gaussian) marginal information on every variable can be efficiently found based on the GMP in this graph after forward and backward recursions.

B. Realization of the Channel Equalizer based on GMP in Factor Graphs

The principles of the GMP can be found in [16]–[18] and [25]. The enhanced message computation rules for GMP were firstly presented in [18]. For completeness, we select some related GMP computation rules in [18] and list them in Tables I and II in Appendix C. Note that all messages and marginal functions in the graph are Gaussian, which can be described either by the mean vector $\mathbf{m}$ and the covariance matrix $\mathbf{V}$ or by the weight matrix $\mathbf{W}$ ($\mathbf{W} = \mathbf{V}^{-1}$) and the transformed mean $\mathbf{W}\mathbf{m}$. In Table I, we list the basic GMP computation rules. However, the direct application of these basic computation rules involves frequent matrix inversions, incurring relatively high computational complexity ($O(L^3)$). Matrix inversions can be avoided by using the message computation rules for composite blocks listed in Table II. In what follows, we first outline the main GMP operations for a building block (see the dashed box in Fig. 3). Then, we will discuss the GMP schedules for the overall graph in sub-section C. For notational convenience, we use the arrows to indicate the directions of messages. For example, in Fig. 3, $\vec{\mathbf{m}}_{\underline{x}_j}$ and $\vec{\mathbf{V}}_{\underline{x}_j}$ denote the mean vector and covariance matrix of $\underline{x}_j$ in the direction from left to right. The letters $\mathbf{m}$, $\mathbf{V}$ and $\mathbf{W}$ (without directions) denote messages after marginalization. For example, $\mathbf{m}_{\underline{x}_j}$ denotes the *a posteriori* mean vector of $\underline{x}_j$.

- **Forward Recursion:** Assume that $\{\vec{\mathbf{m}}_{\underline{x}_{j-1}}, \vec{\mathbf{V}}_{\underline{x}_{j-1}}\}$ are available, which represent the messages provided by the sub-graph left to the j th building block. In this process we compute $\{\vec{\mathbf{m}}_{\underline{x}_j}, \vec{\mathbf{V}}_{\underline{x}_j}\}$ based on $\{\vec{\mathbf{m}}_{\underline{x}_{j-1}}, \vec{\mathbf{V}}_{\underline{x}_{j-1}}\}$, $\{m_{x_j}^\perp, v_{x_j}^\perp\}$ and channel observation r_{j-1} , where $\{m_{x_j}^\perp, v_{x_j}^\perp\}$ (the *a priori* messages) are calculated based on the feedback from the decoder using (7) and (8). Equations (48) and (49) in Table II can be used to obtain $\{\vec{\mathbf{m}}_{\underline{x}'_{j-1}}, \vec{\mathbf{V}}_{\underline{x}'_{j-1}}\}$. From the messages at $\underline{x}'_{j-1}$, $\{\vec{\mathbf{m}}_{\underline{x}_j}, \vec{\mathbf{V}}_{\underline{x}_j}\}$ can be obtained using (42), (43), (37) and (39) in Table I. The complexity in this process

Fig. 4. A sub-graph consisting of Z building blocks, in which the message passing is processed by a processor using the two-way schedule.

is about $1.5L^2$ flops¹.

- **Backward Recursion:** Assume that $\{\bar{\mathbf{W}}_{\mathbf{x}_j} \bar{\mathbf{m}}_{\mathbf{x}_j}, \bar{\mathbf{W}}_{\mathbf{x}_j}\}$ are available, which represent the messages provided by the sub-graph right to the j th building block. In this process, we compute $\{\bar{\mathbf{W}}_{\mathbf{x}_{j-1}} \bar{\mathbf{m}}_{\mathbf{x}_{j-1}}, \bar{\mathbf{W}}_{\mathbf{x}_{j-1}}\}$ based on $\{\bar{\mathbf{W}}_{\mathbf{x}_j} \bar{\mathbf{m}}_{\mathbf{x}_j}, \bar{\mathbf{W}}_{\mathbf{x}_j}\}$, $\{m_{x_j}^\downarrow, v_{x_j}^\downarrow\}$ and r_{j-1} . Equations (51) and (52) in Table II can be used to compute $\{\bar{\mathbf{W}}_{\mathbf{y}_j} \bar{\mathbf{m}}_{\mathbf{y}_j}, \bar{\mathbf{W}}_{\mathbf{y}_j}\}$. Equations (46) and (47) in Table I can be used to get $\{\bar{\mathbf{W}}_{\mathbf{x}'_{j-1}} \bar{\mathbf{m}}_{\mathbf{x}'_{j-1}}, \bar{\mathbf{W}}_{\mathbf{x}'_{j-1}}\}$ and $\{\bar{\mathbf{W}}_{\mathbf{x}''_{j-1}} \bar{\mathbf{m}}_{\mathbf{x}''_{j-1}}, \bar{\mathbf{W}}_{\mathbf{x}''_{j-1}}\}$. Finally, $\{\bar{\mathbf{W}}_{\mathbf{x}_{j-1}} \bar{\mathbf{m}}_{\mathbf{x}_{j-1}}, \bar{\mathbf{W}}_{\mathbf{x}_{j-1}}\}$ are obtained using (33) and (34). It can be shown that the complexity in this process is about L^2 flops.
- **Output Messages:** After the forward and backward recursions, the messages about $\mathbf{x}_j$ in the two directions ($\{\bar{\mathbf{m}}_{\mathbf{x}_j}, \bar{\mathbf{V}}_{\mathbf{x}_j}\}$ and $\{\bar{\mathbf{W}}_{\mathbf{x}_j} \bar{\mathbf{m}}_{\mathbf{x}_j}, \bar{\mathbf{W}}_{\mathbf{x}_j}\}$) are available. Then, its *a posteriori* covariance matrix and mean vector can be calculated as [18]

$$\mathbf{V}_{\mathbf{x}_j} = \left(\bar{\mathbf{V}}_{\mathbf{x}_j}^{-1} + \bar{\mathbf{W}}_{\mathbf{x}_j} \right)^{-1}, \quad (19)$$

$$\mathbf{m}_{\mathbf{x}_j} = \mathbf{V}_{\mathbf{x}_j} \left(\bar{\mathbf{V}}_{\mathbf{x}_j}^{-1} \bar{\mathbf{m}}_{\mathbf{x}_j} + \bar{\mathbf{W}}_{\mathbf{x}_j} \bar{\mathbf{m}}_{\mathbf{x}_j} \right). \quad (20)$$

The diagonal elements of $\mathbf{V}_{\mathbf{x}_j}$ and the elements of $\mathbf{m}_{\mathbf{x}_j}$ are the *a posteriori* variances and means of $x_{j-M}, \dots, x_{j-1}$ and x_j , respectively (note that $\mathbf{x}_j = [x_{j-M}, x_{j-M+1}, \dots, x_j]^T$). This can also be shown as follows. Partition $\mathbf{V}_{\mathbf{x}_j}$ and $\mathbf{m}_{\mathbf{x}_j}$ into the following forms:

$$\mathbf{V}_{\mathbf{x}_j} = \begin{bmatrix} \mathbf{A} & \mathbf{a} \\ \mathbf{a}^T & a \end{bmatrix} \text{ and } \mathbf{m}_{\mathbf{x}_j} = \begin{bmatrix} \mathbf{c} \\ c \end{bmatrix}. \quad (21)$$

Note that the matrix $[\mathbf{G}'' \mathbf{f}]$ is an identity matrix. From (53) and (54) in Table II, we can see that $m_{x_j} = c$ and $v_{x_j} = a$. The same relationship holds for x_{j-1} and $\mathbf{x}_{j-1}$. From (35), (36), (44), (45), (53) and (54), we can represent $\mathbf{V}_{\mathbf{x}_{j-1}}$ and $\mathbf{m}_{\mathbf{x}_{j-1}}$ as

$$\mathbf{V}_{\mathbf{x}_{j-1}} = \begin{bmatrix} b & \mathbf{b}^T \\ \mathbf{b} & \mathbf{A} \end{bmatrix} \text{ and } \mathbf{m}_{\mathbf{x}_{j-1}} = \begin{bmatrix} d \\ \mathbf{c} \end{bmatrix}. \quad (22)$$

Hence, the *a posteriori* messages of x_{j-1} are already included in $\mathbf{V}_{\mathbf{x}_j}$ and $\mathbf{m}_{\mathbf{x}_j}$. Finally, the output extrinsic LLRs can be obtained using (11). For the complexity, computing (19) and (20) requires about $1.7L^3$ flops². Note that we only need to calculate them once every L

building blocks. Therefore, the output process requires about $1.7L^2$ flops per bit.

From the discussion above, the overall complexity of GMP is about $4.2L^2$ flops per bit. As a comparison, the low-complexity extending window approach proposed in [8] requires about LN_1 flops, where N_1 is the non-causal filter length. Normally, to ensure good performance, it requires that $N_1 > 2L$. Thus the complexities of these two approaches are comparable. Other alternatives (such as the sliding window approach in [6]) require higher complexity.

Note that, with the procedure outlined above, the estimation of each bit in the proposed approach is based on the entire observation vector instead of its truncated version. (On the contrary, the estimation in [6]-[9] are all based on truncated observations to avoid high complexity.)

C. The Serial and Parallel Schedules

The factor graph in Fig. 3 has a tree-structure. We can thus apply either the two-way schedule or flooding schedule [21] in message passing. With the two-way schedule, the building blocks in the factor graph operate in a sequential order. The two-way schedule can find the optimal solution using a minimum number of operations. However, due to its serial nature, the two-way schedule may not be efficient if multiple processors are available. With the flooding schedule, every block in the factor graph operates simultaneously and passes messages to its neighbors. This schedule is suited for parallel implementation. However, full implementation of the schedule may require a large number of processors. The hybrid schedule discussed below is more attractive in practice which can be flexibly adjusted according to available hardware resources.

In the hybrid schedule, we partition the overall graph into K sub-graphs. Each sub-graph shown in Fig. 4 contains Z building blocks, where Z is an integer. The k th processor carries out the GMP algorithm for the k th sub-graph using the two-way schedule. Let F_{k-1} and B_{k+1} (see Fig. 4) be the inbound messages to the k th sub-graph in the i th turbo iteration (referring to the iterative process between the equalizer and the decoder). The following highlights the difference between the hybrid and two-way schedules.

- With the two-way schedule, F_{k-1} and B_{k+1} are produced, respectively, by the $(k-1)$ th and $(k+1)$ th sub-graphs in the i th turbo iteration.
- With the hybrid schedule, F_{k-1} and B_{k+1} are produced, respectively, by the $(k-1)$ th and $(k+1)$ th sub-graphs in the $(i-1)$ th turbo iteration.

With the two-way schedule, the K processors have to operate in serial, e.g., the k th processor will not start operation for the forward recursion in a turbo iteration before the $(k-1)$ th processor complete its operation. With the hybrid schedule, all K operators can operate in parallel using the delayed versions of F_{k-1} and B_{k+1} as inputs. By adjusting Z , the latter can provide a compromise between parallelism and hardware complexity.

IV. NUMERICAL RESULTS

Consider a system employing a rate-1/2 nonsystematic convolutional code with generator $(23, 35)_8$. The encoded bit

¹A flop consists of one multiplication and one addition operations.

²Inverting a symmetric matrix with size $L \times L$ takes about $(5/6)L^3$ flops.

Fig. 5. Performance of various approaches in a severely distorted ISI channel.

Fig. 6. Performance of various approaches in a sparse ISI channel.

stream is permuted by a randomly generated interleaver before transmission in BPSK format. We assess the performance of the turbo equalizers in terms of BER versus E_b/N_0 . The APP decoder is implemented based on the BCJR algorithm [24]. The channel equalizer is implemented using different approaches: the MAP approach (based on the BCJR algorithm), the factor graph approach proposed in [13] (the stretching technique is not applied), and the GMP approaches with two-way and hybrid schedules proposed in this paper. These approaches are denoted by “MAP”, “SP”, “GMP-TwoWay” and “GMP-Hybrid”, respectively. In GMP-Hybrid, each sub-graph consists of L building blocks. In all the examples, the performance of the convolutional code over an AWGN channel is also given for reference.

We first consider a severely distorted 5-tap channel (taken from [22]) with coefficients $[0.227, 0.460, 0.688, 0.460, 0.227]$. The data length is set to be 32768. The performance of various approaches is shown in Fig. 5. We can see from this figure that the convergence of GMP-TwoWay is faster than that of GMP-Hybrid in the first few iterations, but their difference becomes marginal after 20 iterations. We can also observe a noticeable performance gap between the GMP based approaches and the MAP approach in this severe ISI channel, which is caused by the sub-optimality of the LMMSE approach. The SP approach can not converge in this case.

We next consider a 6-tap sparse ISI channel with coefficients $[0.408, 0, 0, 0, 0.816, 0.408]$, which is taken from [13]. This channel has girth 6 and represents a more benign channel compared with the one used in Fig. 5. The data length is set to be 4096. Fig. 6 shows the performance of various approaches. The number of turbo iterations is 5 for all the approaches. We can see that, in this channel, all the graph approaches (SP, GMP-TwoWay and GMP-Hybrid) have roughly the same performance that is close to the performance of the optimal MAP approach.

From Figs. 5 and 6, we can see that the convergence behaviors of the GMP and SP approaches depend on channel conditions. To further illustrate their performance difference,

we provide the simulation results in a number of randomly generated ISI channels in Fig. 7. The data length is 8192, iteration number is 30, and the channel length is fixed at 5. The coefficients of each ISI channel are randomly drawn from the Rayleigh distribution and the energy of each ISI channel is normalized to 1. From the results, we can see that statistically, GMP can achieve better convergence than SP. We are currently investigating ways to quantify such performance comparison over all possible channel distributions.

In the above discussions, we have focused on the implementation issues for the channel equalizer. The speed of the turbo receiver in Fig. 1 is also affected by the decoder. The parallel implementation of decoders has been widely studied. For example, the flooding decoding schedule of an LDPC code is highly parallel [15] and the parallel decoding of turbo type codes is studied in [26]–[29].

The following example compares the combined effect of serial and parallel processing for both equalization and decoding. We adopt a length-10 ISI channel with coefficients $[0.2620, 0.2974, 0.4080, -0.0199, 0.4065, 0.0704, -0.4412, -0.0462, -0.1558, -0.5337]$. This example also demonstrates the complexity advantage of the proposed approach since it is difficult to apply the MAP and SP approaches to such a long channel. We still use the convolutional code with generator $(23, 35)_8$. The data length is 8192. The decoding is based on the so-called parallel BCJR algorithm [28], [29]. The trellis for the convolutional code is partitioned into a number of blocks with length 64, and the BCJR algorithm is applied to each block simultaneously. The initialization of each block is based on the related results of its adjacent blocks in the last turbo iteration (which is similar to the discussion related to Fig. 4). Fig. 8 shows the performance of the serial scheme (with equalizer implemented by GMP-TwoWay and decoder implemented by the standard BCJR algorithm) and the parallel scheme (with equalizer implemented by GMP-Hybrid and decoder implemented by the parallel BCJR algorithm). Again, we can see that the performance difference between the parallel and serial schemes is marginal after 10 iterations.

Fig. 7. Comparison of GMP and SP approaches over 10 randomly generated ISI channels.

V. CONCLUSION

We have presented a novel cycle-free graph approach to the channel equalizer in turbo equalization. The equalizer can be realized efficiently based on the recently proposed GMP technique. We have shown the equivalence between the proposed approach and the Wang-Poor approach. We have also demonstrated the advantage of the proposed approach in parallel processing.

APPENDIX A

PROOF OF THE EQUIVALENCE BETWEEN THE WANG-POOR AND JG APPROACHES

The Wang-Poor approach is originally proposed in [9] for multi-user detection and later studied in [6] for ISI channel equalization. The extrinsic LLR derived in [6] can be expressed as (see (8) in [6])

$$L_{Equ}^{WP}(x_j) = \frac{2\mathbf{h}_j^T \mathbf{C}_j^{-1} (\mathbf{r} - \mathbf{H}\mathbf{m}_x^{prio} + \mathbf{h}_j m_{x_j}^{prio})}{1 - \mathbf{h}_j^T \mathbf{C}_j^{-1} \mathbf{h}_j}, \quad (23)$$

where

$$\mathbf{C}_j = \sigma^2 \mathbf{I} + \mathbf{H}\mathbf{V}_x \mathbf{H}^T + (1 - v_{x_j}^{prio}) \mathbf{h}_j \mathbf{h}_j^T, \quad (24)$$

and $\mathbf{V}_x = \text{diag}[v_{x_0}^{prio}, \dots, v_{x_{j-1}}^{prio}]$. Comparing (24) and (6), we have

$$\mathbf{C}_j = \mathbf{V}_{\xi_j} + \mathbf{h}_j \mathbf{h}_j^T. \quad (25)$$

With the matrix inversion lemma [23], we obtain

$$\begin{aligned} \mathbf{h}_j^T (\mathbf{V}_{\xi_j} + \mathbf{h}_j \mathbf{h}_j^T)^{-1} &= \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} - \frac{\mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1}}{1 + \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j} \\ &= \frac{\mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1}}{1 + \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j}. \end{aligned} \quad (26)$$

Substituting (26) into (23) yields

$$\begin{aligned} L_{Equ}^{WP}(x_j) &= 2\mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} (\mathbf{r} - \mathbf{H}\mathbf{m}_x^{prio} + \mathbf{h}_j m_{x_j}^{prio}) \\ &= L_{Equ}^{JG}(x_j). \end{aligned} \quad (27)$$

Fig. 8. Performance of the serial and parallel schemes in a length-10 ISI channel.

APPENDIX B

PROOF OF PROPOSITION 1

By performing the LMMSE estimation [20] for the Gaussian companion (10), we have

$$\mathbf{m}_x^{post} = \mathbf{m}_x^{prio} + \text{Cov}(\mathbf{x}, \mathbf{r}) \mathbf{V}_r^{-1} (\mathbf{r} - \mathbf{H}\mathbf{m}_x^{prio}), \quad (28)$$

where $\mathbf{V}_r = \text{Cov}(\mathbf{r}, \mathbf{r}) = \mathbf{H}\mathbf{V}_x \mathbf{H}^T + \sigma^2 \mathbf{I}$ and $\mathbf{V}_x = \text{diag}[v_{x_0}^{prio}, \dots, v_{x_{j-1}}^{prio}]$. Define $\mathbf{V}_{x(j)} = \text{diag}[v_{x_0}^{prio}, \dots, v_{x_{j-1}}^{prio}, 0, v_{x_{j+1}}^{prio}, \dots, v_{x_{J-1}}^{prio}]$ and rewrite $\mathbf{V}_r$ as

$$\mathbf{V}_r = \mathbf{V}_{\xi_j} + v_{x_j}^{prio} \mathbf{h}_j \mathbf{h}_j^T, \quad (29)$$

where $\mathbf{V}_{\xi_j} = \mathbf{H}\mathbf{V}_{x(j)} \mathbf{H}^T + \sigma^2 \mathbf{I}$.

We focus on x_j . According to (28), $m_{x_j}^{post}$ can be written as

$$\begin{aligned} m_{x_j}^{post} &= m_{x_j}^{prio} + v_{x_j}^{prio} \mathbf{h}_j^T \mathbf{V}_r^{-1} (\mathbf{r} - \mathbf{H}\mathbf{m}_x^{prio}) \\ &= m_{x_j}^{prio} + v_{x_j}^{prio} \mathbf{h}_j^T (\mathbf{V}_{\xi_j} + v_{x_j}^{prio} \mathbf{h}_j \mathbf{h}_j^T)^{-1} (\mathbf{r} - \mathbf{H}\mathbf{m}_x^{prio}) \\ &= m_{x_j}^{prio} + v_{x_j}^{prio} \mathbf{h}_j^T \left(\mathbf{V}_{\xi_j}^{-1} - \frac{\mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1}}{\mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j + 1/v_{x_j}^{prio}} \right) (\mathbf{r} - \mathbf{H}\mathbf{m}_x^{prio}) \\ &= \frac{m_{x_j}^{prio}/v_{x_j}^{prio} + \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} (\mathbf{r} - \mathbf{H}\mathbf{m}_x^{prio} + \mathbf{h}_j m_{x_j}^{prio})}{1/v_{x_j}^{prio} + \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j}. \end{aligned} \quad (30)$$

Its a posterior variance $v_{x_j}^{post}$ is given by [20]

$$\begin{aligned} v_{x_j}^{post} &= E((x_j - m_{x_j}^{post})^2) \\ &= v_{x_j}^{prio} - \text{Cov}(x_j, \mathbf{r}) \mathbf{V}_r^{-1} \text{Cov}(\mathbf{r}, x_j) \\ &= v_{x_j}^{prio} - (v_{x_j}^{prio})^2 \mathbf{h}_j^T \mathbf{V}_r^{-1} \mathbf{h}_j \\ &= v_{x_j}^{prio} - (v_{x_j}^{prio})^2 \mathbf{h}_j^T \left(\mathbf{V}_{\xi_j}^{-1} - \frac{\mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1}}{\mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j + 1/v_{x_j}^{prio}} \right) \mathbf{h}_j \\ &= \frac{1}{1/v_{x_j}^{prio} + \mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \mathbf{h}_j}. \end{aligned} \quad (31)$$

TABLE I
MESSAGE COMPUTATION RULES FOR BASIC BUILDING BLOCKS

Blocks	Rules
	$\vec{W}_z = \vec{W}_x + \vec{W}_y^\dagger \quad (33)$ $\vec{W}_z \vec{m}_z = \vec{W}_x \vec{m}_x + \vec{W}_y^\dagger \vec{m}_y^\dagger \quad (34)$ $\vec{m}_x = \vec{m}_y = \vec{m}_z \quad (35)$ $V_x = V_y = V_z \quad (36)$
	$\vec{V}_z = \vec{V}_x + \vec{V}_y^\dagger \quad (37)$ $\vec{V}_x = \vec{V}_z + \vec{V}_y^\dagger \quad (38)$ $\vec{m}_z = \vec{m}_x + \vec{m}_y^\dagger \quad (39)$ $\vec{m}_x = \vec{m}_z - \vec{m}_y^\dagger \quad (40)$ $\vec{m}_x + \vec{m}_y - \vec{m}_z = 0 \quad (41)$
	$\vec{V}_y = A \vec{V}_x A^H \quad (42)$ $\vec{m}_y = A \vec{m}_x \quad (43)$ $V_y = A V_x A^H \quad (44)$ $\vec{m}_y = A \vec{m}_x \quad (45)$
	$\vec{W}_x = A^H \vec{W}_y A \quad (46)$ $\vec{W}_x \vec{m}_x = A^H \vec{W}_y \vec{m}_y \quad (47)$

TABLE II
MESSAGE COMPUTATION RULES FOR COMPOSITE BLOCKS

Blocks	Rules
	$\vec{m}_z = \vec{m}_x + c \left(\vec{m}_y^\dagger - a^H \vec{m}_x \right) \vec{V}_x a \quad (48)$ $\vec{V}_z = \vec{V}_x - c \vec{V}_x a a^H \vec{V}_x \quad (49)$ with $c = 1 / \left(\vec{V}_y^\dagger + a^H \vec{V}_x a \right)$
	$\vec{m}_z = -\vec{m}_x - a \vec{m}_y^\dagger \quad (50)$ $\vec{W}_z = \vec{W}_x - d \vec{W}_x a a^H \vec{W}_x \quad (51)$ with $d = 1 / \left(\vec{W}_y^\dagger + a^H \vec{W}_x a \right)$ $\vec{W}_z \vec{m}_z = -\vec{W}_x \vec{m}_x + d \vec{W}_x a a^H \vec{W}_x \vec{m}_x - d \vec{W}_x a \vec{W}_y^\dagger \vec{m}_y^\dagger \quad (52)$
	If $[A \ B]$ is a nonsingular square matrix: $\begin{bmatrix} \vec{m}_x \\ \vec{m}_y \end{bmatrix} = [A \ B]^{-1} \vec{m}_z \quad (53)$ $\begin{bmatrix} V_x & V_{xy} \\ V_{yx} & V_y \end{bmatrix} = [A \ B]^{-1} V_z \begin{bmatrix} A^H & B^H \end{bmatrix}^{-1} \quad (54)$

REFERENCES

- [1] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glarieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507-511, Sept./Oct. 1995.
- [2] G. Bauchand V. Franz, "A comparison of soft-in/soft-out algorithms for 'Turbo detection'," in *Proc. Int. Conf. Telecommun.*, June 1998, pp. 259-263.
- [3] P. Magniez, P. Duhamel, A. Roumy, and I. Fijalkow, "Turbo-equalization applied to trellis-coded-modulation" in *Proc. IEEE VTC99-Fall*, Amsterdam, The Netherlands, Sept. 1999, pp. 2556-2560.
- [4] A. Glarieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel" in *Proc. Int. Symp. Turbo codes*, Brest, France, Sept 1997, pp. 96-102.
- [5] D. Reynolds and X. Wang, "Low complexity turbo-equalization for diversity channels," *Signal Processing*, vol. 81, pp. 989-995, May 2001.
- [6] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: principles and new results" *IEEE Trans. Commun.*, vol. 50, pp. 754-767, May 2002.
- [7] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Signal Processing*, vol. 50, pp. 673-683, Mar. 2002.
- [8] L. Liu and Li Ping, "An extending window MMSE turbo equalization algorithm," *IEEE Signal Processing Lett.*, vol. 11, pp. 891-894, Nov. 2004.
- [9] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 1046-1061, July 1999.
- [10] L. Liu, W. K. Leung and Li Ping, "Simple chip-by-chip multiuser detection for CDMA systems," *IEEE Vehicular Tech. Conf.*, VTC'03, pp. 2157-2161.
- [11] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533-547, Sept. 1981.
- [12] T. S. Rappaport, *Wireless Communications-Principles and Practice*, Prentice-Hall, Upper Saddle, River, NJ, US, 1996.
- [13] G. Colavolpe and G. Germei, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Trans. Commun.*, vol. 53, pp. 818-825, May 2005.
- [14] G. D. Forney, Jr., "Codes on graph: Normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, no.2, pp.520-548, Feb. 2001.
- [15] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498-519, Feb. 2001.

Combining (30) and (31), we have

$$2\mathbf{h}_j^T \mathbf{V}_{\xi_j}^{-1} \left(\mathbf{r} - \mathbf{H} \mathbf{m}_x^{prio} + \mathbf{h}_j \mathbf{m}_{x_j}^{prio} \right) = 2 \left(\frac{m_{x_j}^{post}}{v_{x_j}^{post}} - \frac{m_{x_j}^{prio}}{v_{x_j}^{prio}} \right). \quad (32)$$

The above derivation is for the Gaussian companion (10). However, recall from Definition 1 that the means $\{m_{x_j}^{prio}\}$ and variances $\{v_{x_j}^{prio}\}$ in the Gaussian system (10) and the binary system (1) are the same. This implies V_{ξ_j} in the system (10) is also the same as that in system (1). Hence, equation (11) holds since the left hand side of (32) mathematically equals the last line in (9).

APPENDIX C MESSAGE COMPUTATION RULES FOR GMP

The direct application of the basic rules listed in Table I may involve matrix inversions, which can be avoided by using the message computation rules for the composite blocks listed in Table II. (In the tables, the superscript H denotes the conjugate transpose.)

ACKNOWLEDGMENT

The second author is grateful to Prof. Hans-Andrea Loeliger for enlightening discussions when he visited ETH in 2004. The contribution of Prof. Loeliger to the Gaussian message passing principle is the basis of the work in this paper.

- [16] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Mag.*, pp. 28-41, Jan. 2004.
- [17] H.-A. Loeliger, "Least squares and Kalman filtering on Forney graphs," in *Codes, Graphs, and Systems*, (festschrift in honor of David Forney on the occasion of his 60th birthday), R.E. Blahut and R. Koetter, eds., Kluwer, 2002, pp. 113-135.
- [18] H.-A. Loeliger, J. Hu, S. Korl, Q. Guo and Li Ping, "Gaussian message passing on linear models: an update," *Int. Symp. on Turbo codes and Related Topics*, Apr. 2006.
- [19] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proc. 1993 Int. Conf. Commun.*, pp. 1064-1070.
- [20] Steven M. Kay, *Fundamentals of statistical signal processing*, Prentice-Hall PTR, 1993.
- [21] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Select. Areas Commun.*, pp. 219-230, Feb. 1998.
- [22] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.
- [23] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.
- [24] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp.284-287, Mar. 1974.
- [25] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, Li Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. of the IEEE*, vol. 95, pp.1295-1322, Jun. 2007.
- [26] B. J. Frey, F. R. Kschischang, and P. G. Gulak, "Concurrent turbo-decoding," in *Proc. IEEE Int. Symp. Inform. Theory*, Ulm, Germany, June/July 1997.
- [27] C. Argon and S. W. McLaughlin, "A parallel decoder for low latency decoding of turbo product codes," *IEEE Commun., Letters*, vol. 6, pp.70-72, Feb. 2002.
- [28] J. Hsu and C. Wang, "A parallel decoding scheme for turbo codes," in *IEEE Int. Conf. on Circuits and Systems*, vol. 4, June 1998, pp.445-448.
- [29] U. Dasgupta and K. R. Narayanan, "Parallel decoding of turbo codes using soft output T-Algorithms," *IEEE Commun., Letters*, vol. 5, pp. 352-354, Aug. 2001.

Qinghua Guo (S'07) received his B.E. degree in electronic engineering and M.E. degree in signal and information processing from Xidian University, China, in 2001 and 2004, respectively. He is currently working towards the Ph.D. degree at City University of Hong Kong. His research interests include statistical signal processing and broadband wireless communications.

Li Ping (S'87-M'91-SM'06) received his Ph.D. degree at Glasgow University in 1990. He lectured at Department of Electronic Engineering, Melbourne University, from 1990 to 1992, and worked as a member of research staff at Telecom Australia Research Laboratories from 1993 to 1995. He has been with the Department of Electronic Engineering, City University of Hong Kong, since January 1996, where he is now a chair professor. His research interests are communications systems and coding theory. Dr. Li Ping was awarded a British Telecom-

Royal Society Fellowship in 1986, the IEE J J Thomson premium in 1993 and a Croucher Senior Research Fellowship in 2005.