

University of Wollongong

## Research Online

---

Faculty of Engineering and Information  
Sciences - Papers: Part A

Faculty of Engineering and Information  
Sciences

---

1-1-2012

### A soft-in soft-out detection approach using partial gaussian approximation

Qinghua Guo

*University of Wollongong*, [qguo@uow.edu.au](mailto:qguo@uow.edu.au)

Licai Fang

*University Of Western Australia*

Defeng (David) Huang

*University Of Western Australia*, [david.huang@uwa.edu.au](mailto:david.huang@uwa.edu.au)

Sven Nordholm

*Curtin University*

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](mailto:research-pubs@uow.edu.au)

---

## A soft-in soft-out detection approach using partial gaussian approximation

### Abstract

This paper concerns the implementation of the softin soft-out detector in an iterative detection system. A detection approach is proposed based on the properties of Gaussian functions. In this approach, for the computation of the APP (a posteriori probability) of a concerned symbol, the other symbols are distinguished based on their contributions to the APP of the concerned symbol, and the symbols with less contributions are treated as Gaussian variables to reduce the computational complexity. The exact APP detector and the well-known LMMSE (linear minimum mean square error) detector are two special cases of the proposed detector. Simulation results show that the proposed detector can significantly outperform the LMMSE detector, and achieve a good trade-off between complexity and performance.

### Keywords

detection, approximation, out, partial, soft, approach, gaussian

### Disciplines

Engineering | Science and Technology Studies

### Publication Details

Q. Guo, L. Fang, D. Huang & S. Nordholm, "A soft-in soft-out detection approach using partial gaussian approximation," in International Conference on Wireless Communications and Signal Processing, 2012, pp. 1-6.

# A Soft-in Soft-out Detection Approach Using Partial Gaussian Approximation

Qinghua Guo<sup>\*†</sup>, Licai Fang<sup>†</sup>, Defeng (David) Huang<sup>†</sup>, and Sven Nordholm<sup>‡</sup>

<sup>\*</sup>School of Electrical, Computer and Telecommunications Engineering, University of Wollongong  
Email: qinghua\_guo@uow.edu.au

<sup>†</sup>School of Electrical, Electronic and Computer Engineering, The University of Western Australia  
Email: {qinghua.guo, david.huang}@uwa.edu.au, licaifang@gmail.com

<sup>‡</sup>Department of Electrical and Computer Engineering, Curtin University  
Email: s.nordholm@curtin.edu.au

**Abstract**— This paper concerns the implementation of the soft-in soft-out detector in an iterative detection system. A detection approach is proposed based on the properties of Gaussian functions. In this approach, for the computation of the APP (a posteriori probability) of a concerned symbol, the other symbols are distinguished based on their contributions to the APP of the concerned symbol, and the symbols with less contributions are treated as Gaussian variables to reduce the computational complexity. The exact APP detector and the well-known LMMSE (linear minimum mean square error) detector are two special cases of the proposed detector. Simulation results show that the proposed detector can significantly outperform the LMMSE detector, and achieve a good trade-off between complexity and performance.

## I. INTRODUCTION

Recently, the implementation of the soft-in soft-out (SISO) detector for iterative detection in a coded system has received tremendous attention as a turbo (iterative) receiver may achieve huge performance gain over a conventional non-iterative receiver [1]–[11]. The structure of a coded system with a turbo receiver is shown in Fig. 1, where the receiver consists of a SISO detector and a SISO decoder, working in an iterative manner. The task of the SISO detector is to compute the extrinsic log-likelihood ratio (LLR) for each code bit with the output extrinsic LLRs of the decoder as a priori information [1]–[11].

The optimal implementation of the detection is the a posteriori probability (APP) detector [1]. However, the complexity of the APP detector grows exponentially, and it may be too high to be used in practice. A low-complexity alternative is based on the linear minimum mean square error (LMMSE) principle [2], [3]. The complexity of the LMMSE detector is in quadratic level, which is significantly lower than that of the APP detector. If the system transform matrix is a circulant one, the LMMSE detector can be implemented with the fast Fourier transform (FFT), reducing the complexity to logarithmic level [8]–[11]. The LMMSE detector is an attractive option due to its low-complexity compared with the APP counterpart. However, it may suffer from significant performance loss for channels that severely distort the transmitted signal.

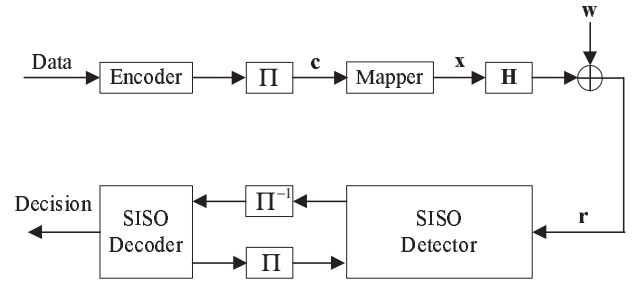


Fig. 1. The structure of a coded system with a turbo receiver, where  $\Pi$  and  $\Pi^{-1}$  denote the interleaver and the deinterleaver, respectively.

The key task of the APP detector is to compute the APP of each (discrete-valued) symbol, which involves high-dimensional summation over all the other symbols. In this work, we propose an approximate approach to compute the APP based on the properties of Gaussian functions. Suppose that the number of the symbols involved is  $N$ . In the computation of the APP of a concerned symbol, we distinguish all the other  $N - 1$  symbols based on their contributions to the APP of the symbol. We select  $M$  ( $M \leq N - 1$ ) important symbols with greater contributions to the APP of the concerned symbol, and keep them as discrete random variables without any approximation. On the other hand, the remaining  $N - 1 - M$  symbols with less contributions are treated as Gaussian random variables to reduce the complexity. When  $M = N - 1$ , the proposed detector is the exact APP detector. It can also be shown that the well-known LMMSE detector [2], [3] is a special case of the proposed detector with  $M = 0$ . Simulation results show that the proposed detector with a small  $M$  (e.g.,  $M = 2$ ) can significantly outperform the LMMSE detector, and achieve a good trade-off between complexity and performance.

The notations used in this paper are as follows. Lower and upper case letters denote scalars. Bold lower and upper case letters represent column vectors and matrices, respectively. The probability density function (PDF) of a continuous random variable and the probability mass function of a discrete random variable are represented by  $p(\cdot)$  and  $P(\cdot)$ , respectively.

<sup>†</sup>This work was supported by Australian Research Council's Discovery Projects DP1093000 and DP110100736, and DECRA Grant DE120101266.

We use  $\propto$  to denote equality of functions up to a scale factor. The superscripts “ $T$ ” and “ $H$ ” denote the transpose and conjugate transpose, respectively. The identity matrix with size  $N$  is denoted by  $\mathbf{I}_N$ .

## II. PRELIMINARIES

In this section, we list some properties of the Gaussian functions, which will be used to derive the new SISO detector. We use  $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V})$  to denote a complex multivariate Gaussian function with  $\mathbf{x}$  as variable, i.e.,

$$\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) = \frac{1}{\pi^N \det(\mathbf{V})} \exp[-(\mathbf{x} - \mathbf{m})^H \mathbf{V}^{-1} (\mathbf{x} - \mathbf{m})] \quad (1)$$

where  $N$  is the length of  $\mathbf{x}$ , and  $\mathbf{V}$  is a Hermitian matrix.

### A. Marginal Distribution of a Complex Gaussian Vector

Assume that  $\mathbf{x}$  is a length- $N$  complex Gaussian random vector with PDF

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) \quad (2)$$

and  $\mathbf{x}_s$  with length  $M$  ( $M \leq N$ ) is a sub-vector of  $\mathbf{x}$ , i.e.,

$$\mathbf{x}_s = \mathbf{S}\mathbf{x} \quad (3)$$

where  $\mathbf{S}$  is an  $M \times N$  selecting matrix (i.e., each of its row vectors is a row of the identity matrix  $\mathbf{I}_N$ ). It can be shown that the distribution of  $\mathbf{x}_s$

$$\begin{aligned} p(\mathbf{x}_s) &= \int_{\mathbf{x}_r} \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) d\mathbf{x}_r \\ &= \mathcal{N}(\mathbf{x}_s; \mathbf{S}\mathbf{m}, \mathbf{S}\mathbf{V}\mathbf{S}^T) \end{aligned} \quad (4)$$

where  $\mathbf{x}_r$  consists of the remaining elements in  $\mathbf{x}$  after removing those in  $\mathbf{x}_s$ .

### B. Multiplication of Two Multivariate Gaussian Functions

The multiplication of two Gaussian functions is another Gaussian function [15]<sup>1</sup>, i.e.,

$$\mathcal{N}(\mathbf{x}; \mathbf{m}_a, \mathbf{V}_a) \mathcal{N}(\mathbf{x}; \mathbf{m}_b, \mathbf{V}_b) \propto \mathcal{N}(\mathbf{x}; \mathbf{m}_c, \mathbf{V}_c) \quad (5)$$

where

$$\mathbf{V}_c = (\mathbf{V}_a^{-1} + \mathbf{V}_b^{-1})^{-1} \quad (6)$$

$$\mathbf{m}_c = \mathbf{V}_c(\mathbf{V}_a^{-1}\mathbf{m}_a + \mathbf{V}_b^{-1}\mathbf{m}_b). \quad (7)$$

From the above, we have

$$\frac{\mathcal{N}(\mathbf{x}; \mathbf{m}_c, \mathbf{V}_c)}{\mathcal{N}(\mathbf{x}; \mathbf{m}_a, \mathbf{V}_a)} \propto \mathcal{N}(\mathbf{x}; \mathbf{m}_b, \mathbf{V}_b) \quad (8)$$

and

$$\mathbf{V}_b = (\mathbf{V}_c^{-1} - \mathbf{V}_a^{-1})^{-1} \quad (9)$$

$$\mathbf{m}_b = \mathbf{V}_b(\mathbf{V}_c^{-1}\mathbf{m}_c - \mathbf{V}_a^{-1}\mathbf{m}_a). \quad (10)$$

<sup>1</sup>The results in [15] are for real multivariate Gaussian functions. They can be extended to the case of complex multivariate Gaussian functions.

## III. IMPLEMENTATION OF THE SISO DETECTOR

### A. Problem Description

The coded system shown in Fig. 1 can be represented as

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{w} \quad (11)$$

where  $\mathbf{r}$  denotes a length- $Z$  observation vector,  $\mathbf{H}$  denotes a  $Z \times N$  system transfer matrix,  $\mathbf{w}$  denotes a length- $Z$  circularly symmetric additive white Gaussian noise (AWGN) vector with PDF  $\mathcal{N}(\mathbf{w}; 0, \sigma^2 \mathbf{I}_Z)$ , and  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$  is mapped from an interleaved code sequence  $\mathbf{c}$ , i.e., each  $x_n \in \mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_{2^Q}\}$  ( $|\mathcal{A}| = 2^Q$ ) corresponds to a length- $Q$  subsequence of  $\mathbf{c}$  denoted by  $\mathbf{c}_n = [c_{n,1}, c_{n,2}, \dots, c_{n,Q}]^T$ . The task of the detector is to compute the log-likelihood ratio (LLR) for each code bit  $c_{n,q}$  (without taking coding into account), which can be expressed as [1]-[3]

$$\begin{aligned} L(c_{n,q}) &= \ln \frac{P(c_{n,q} = 0 | \mathbf{r})}{P(c_{n,q} = 1 | \mathbf{r})} \\ &= \ln \frac{\sum_{x_n \in \mathcal{A}_q^0} P(x_n | \mathbf{r})}{\sum_{x_n \in \mathcal{A}_q^1} P(x_n | \mathbf{r})} \end{aligned} \quad (12)$$

where  $\mathcal{A}_q^0$  ( $\mathcal{A}_q^1$ ) denotes the subset of all  $\alpha_i \in \mathcal{A}$  corresponding to a binary subsequence with the  $q$ th bit given by 0 (1). In a turbo receiver as shown in Fig. 1, the extrinsic LLR [1]-[3]

$$L^e(c_{n,q}) = L(c_{n,q}) - L^a(c_{n,q}) \quad (13)$$

will be input to the decoder, where  $L^a(c_{n,q})$  is the output extrinsic LLR of the decoder in the last iteration.

It can be seen that the key task of the detector is to compute the APP  $P(x_n | \mathbf{r})$  for each symbol  $x_n$ , which is the focus of this work.

### B. APP Detector

According to Bayesian's rule

$$\begin{aligned} P(x_n | \mathbf{r}) &= \sum_{\tilde{\mathbf{x}}_n} P(\mathbf{x} | \mathbf{r}) \\ &\propto \sum_{\tilde{\mathbf{x}}_n} P(\mathbf{x}) p(\mathbf{r} | \mathbf{x}) \end{aligned} \quad (14)$$

where the length- $(N-1)$  vector  $\tilde{\mathbf{x}}_n$  consists of the elements of  $\mathbf{x}$  except  $x_n$ . Given  $\mathbf{x}$ ,  $\mathbf{r}$  is Gaussian distributed, i.e.,  $p(\mathbf{r} | \mathbf{x}) = \mathcal{N}(\mathbf{r}; \mathbf{H}\mathbf{x}, \sigma^2 \mathbf{I})$ . Hence (14) can be rewritten as

$$P(x_n | \mathbf{r}) \propto \sum_{\tilde{\mathbf{x}}_n} P(\mathbf{x}) \exp \left[ -\frac{(\mathbf{r} - \mathbf{H}\mathbf{x})^H (\mathbf{r} - \mathbf{H}\mathbf{x})}{\sigma^2} \right]. \quad (15)$$

In (15), assuming that the interleaved code bits are independent of each other, we have [1]-[3]

$$P(\mathbf{x}) = \prod_i P(x_i) \quad (16)$$

$$P(x_i) = \prod_q P(c_{i,q}) \quad (17)$$

where  $P(c_{i,q})$  can be calculated based on the LLRs fed back from the decoder.

By using (15), we can calculate the APPs  $\{P(x_n = \alpha_i | \mathbf{r}), \forall i\}$  up to some constant. Although the constant can be found through normalization, we do not need to care about it in the calculation of the LLR using (12) as it will be cancelled out. Due to the high-dimensional summation involved in (15), the computational complexity of the APP detector is  $\mathcal{O}(N|\mathcal{A}|^N)$  per symbol (assuming that  $Z$  and  $N$  are in the same order), which is prohibitive even for moderate  $N$ .

Note that, when model (11) is used to represent that signal  $\mathbf{x}$  is transmitted over an inter-symbol interference (ISI) channel  $\mathbf{h} = [h_0, h_1, \dots, h_L]^T$ , i.e.,  $\mathbf{r}$  is the convolution of  $\mathbf{x}$  and  $\mathbf{h}$ , corrupted by AWGN  $\mathbf{w}$  (in this case,  $\mathbf{H}$  is a Toeplitz matrix). The APP detector can be implemented using the BCJR algorithm with complexity  $\mathcal{O}(|\mathcal{A}|^L)$  [13]. However, when  $L$  is large (e.g., in broadband wireless communications or underwater acoustic communications), the use of the BCJR-based APP detector is still impractical due to the high complexity. In the following, we consider approximate approaches to the calculation of  $P(x_n | \mathbf{r})$ .

### C. Approximate APP Detector

We assume that  $\mathbf{H}^H \mathbf{H}$  is invertible, and define  $\mathbf{a} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{r}$ . It is easy to verify that

$$(\mathbf{r} - \mathbf{H}\mathbf{x})^H (\mathbf{r} - \mathbf{H}\mathbf{x}) = (\mathbf{x} - \mathbf{a})^H \mathbf{H}^H \mathbf{H} (\mathbf{x} - \mathbf{a}) - \mathbf{r}^H (\mathbf{H}(\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H - \mathbf{I}_Z) \mathbf{r}. \quad (18)$$

As a result,

$$\exp \left[ -\frac{(\mathbf{r} - \mathbf{H}\mathbf{x})^H (\mathbf{r} - \mathbf{H}\mathbf{x})}{\sigma^2} \right] = \varphi g(\mathbf{x}) \quad (19)$$

where

$$g(\mathbf{x}) = \exp \left[ -\frac{(\mathbf{x} - \mathbf{a})^H \mathbf{H}^H \mathbf{H} (\mathbf{x} - \mathbf{a})}{\sigma^2} \right] \quad (20)$$

and

$$\varphi = \exp \left[ \frac{\mathbf{r}^H (\mathbf{H}(\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H - \mathbf{I}_Z) \mathbf{r}}{\sigma^2} \right] \quad (21)$$

is independent of  $\mathbf{x}$ .

Now we reformulate the APP in (15) using (19). Noting that the likelihood function  $p(\mathbf{r} | \mathbf{x})$  in (14) is a function of  $\mathbf{x}$  (rather than  $\mathbf{r}$  which is fixed), we can safely ignore  $\varphi$  in (19). Hence

$$P(x_n | \mathbf{r}) \propto \sum_{\tilde{\mathbf{x}}_n} P(\mathbf{x}) g(\mathbf{x}) \quad (22)$$

$$\propto \sum_{\tilde{\mathbf{x}}_n} P(\mathbf{x}) \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) \quad (23)$$

where  $\mathbf{A} = \sigma^2 (\mathbf{H}^H \mathbf{H})^{-1}$ . To obtain (23), we have used

$$g(\mathbf{x}) \propto \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}). \quad (24)$$

Here, we emphasize that the notation  $\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})$  in (23) and (24) just denotes a function of  $\mathbf{x}$  as shown in (1), and it does not mean that  $\mathbf{x}$  is a Gaussian random vector.

Although the exact calculation of (23) still involves high-dimensional summation, it enables us to derive low-complexity approximate approaches based on the properties of Gaussian functions in Section II.

If we treat  $\mathbf{x}$  as a Gaussian random vector with (a priori) PDF  $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V})$  (instead of a discrete random vector with a priori probability mass function  $P(\mathbf{x})$ ), the summation in (23) is changed to integration accordingly, so we have

$$\tilde{P}(x_n | \mathbf{r}) \propto \int_{\tilde{\mathbf{x}}_n} \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) d\tilde{\mathbf{x}}_n \quad (25)$$

where we use  $\tilde{P}(x_n | \mathbf{r})$  to denote an approximation of  $P(x_n | \mathbf{r})$  due to the above Gaussian approximation. In (25), the a priori covariance matrix  $\mathbf{V}$  is a diagonal matrix as  $\{x_i\}$  are independent of each other. A straightforward way to determine  $m_i$  (the  $i$ th element of  $\mathbf{m}$ ) and  $v_i$  (the  $i$ th diagonal element of  $\mathbf{V}$ ) is to use the actual first and second moments of  $x_i$ , i.e.,

$$m_i = \sum_{\alpha_k \in \mathcal{A}} \alpha_k P(x_i = \alpha_k) \quad (26)$$

$$v_i = \sum_{\alpha_k \in \mathcal{A}} |\alpha_k - m_i|^2 P(x_i = \alpha_k). \quad (27)$$

Although (25) involves high-dimensional integration, it can be efficiently calculated based on the properties of Gaussian functions shown by (5)-(7) and (4). The complexity is about  $\mathcal{O}(N^2)$  per symbol (assuming that  $N$  and  $Z$  are in the same order), which is much lower than that of exactly calculating (15) with complexity  $\mathcal{O}(N|\mathcal{A}|^N)$ . However, simulation results in Section IV show that this approach delivers very poor performance. As all the symbols are treated as Gaussian variables in this approach, we call it full Gaussian approximation approach.

Now we go back to (22) and (20). By expanding the exponent of  $g(\mathbf{x})$  in (20) to scalar form, it can be found that some of the non-zero off-diagonal elements of  $\mathbf{H}^H \mathbf{H}$  generate the cross terms between  $x_n$  and  $\{x_i, i \neq n\}$ . If all of these cross terms are zeroes,  $\{x_i, i \neq n\}$  make no contributions to the APP of  $x_n$ . Due to the non-zero cross terms,  $\{x_i, i \neq n\}$  may have different contributions to the APP of  $x_n$ . Although it is hard to evaluate their contributions exactly, we can use the following approach to distinguish them roughly. Define  $\mathbf{p}_n$  as the  $n$ th column vector of matrix  $\mathbf{H}^H \mathbf{H}$ . We can find that the  $i$ th element of  $\mathbf{p}_n$  is the coefficient of the cross term between  $x_n$  and  $x_i$  up to a constant. We distinguish the contributions of different symbols based on the magnitudes of the elements of  $\mathbf{p}_n$ , i.e., the larger the magnitude of an element, the larger the contribution of its corresponding symbol to the APP of  $x_n$ . According to this, from  $\{x_i, i \neq n\}$ , we can select  $M$  most important symbols to  $x_n$ . We keep the  $M$  most important symbols as discrete variables without any approximation, but treat the remaining  $N - M - 1$  symbols as Gaussian variables. Including  $x_n$  itself, the number of discrete variables is  $M + 1$ , and they are denoted by

$$\mathbf{x}_n^D = \mathbf{S}_n \mathbf{x} \quad (28)$$

where  $\mathbf{S}_n$  is the corresponding  $(M+1) \times N$  selecting matrix for  $x_n$ . The remaining  $N - M - 1$  symbols to be treated as Gaussian variables are denoted by  $\mathbf{x}_n^G$ , and their means and variances can be computed as (26) and (27). We denote the mean vector and (diagonal) covariance matrix of  $\mathbf{x}_n^G$  by  $\mathbf{m}_n^G$  and  $\mathbf{V}_n^G$ , respectively.

Based on the above partial Gaussian approximation, part of the summation in (23) will be changed to integration, and the APP of  $x_n$  can be approximated as

$$\tilde{P}(x_n | \mathbf{r}) \propto \sum_{\tilde{\mathbf{x}}_n^D} P(\mathbf{x}_n^D) h(\mathbf{x}_n^D) \quad (29)$$

where  $\tilde{\mathbf{x}}_n^D$  consists of the elements in  $\mathbf{x}_n^D$  except  $x_n$ , and

$$h(\mathbf{x}_n^D) = \int_{\mathbf{x}_n^G} \mathcal{N}(\mathbf{x}_n^G; \mathbf{m}_n^G, \mathbf{V}_n^G) \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) d\mathbf{x}_n^G. \quad (30)$$

It can be seen that, assuming  $h(\mathbf{x}_n^D)$  is available, (29) involves an  $M$ -dimensional summation. To reduce the complexity, we may set  $M$  to be a small integer. In the following, we discuss how to compute  $h(\mathbf{x}_n^D)$  in (30).

Note that (30) involves the multiplication of two Gaussian functions with different dimensions (noting that  $\mathbf{x}_n^G$  is a sub-vector of  $\mathbf{x}$ ), and hence (5)-(7) can not be directly applied. To overcome this problem, we rewrite  $h(\mathbf{x}_n^D)$  as

$$\begin{aligned} h(\mathbf{x}_n^D) &= \frac{\mathcal{N}(\mathbf{x}_n^D; \mathbf{m}_n^D, \mathbf{V}_n^D) \int_{\mathbf{x}_n^G} \mathcal{N}(\mathbf{x}_n^G; \mathbf{m}_n^G, \mathbf{V}_n^G) \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) d\mathbf{x}_n^G}{\mathcal{N}(\mathbf{x}_n^D; \mathbf{m}_n^D, \mathbf{V}_n^D)} \\ &= \frac{\int_{\mathbf{x}_n^G} \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) d\mathbf{x}_n^G}{\mathcal{N}(\mathbf{x}_n^D; \mathbf{m}_n^D, \mathbf{V}_n^D)} \end{aligned} \quad (31)$$

where  $\mathbf{m}_n^D$  and  $\mathbf{V}_n^D$  are the mean vector and (diagonal) covariance matrix of  $\mathbf{x}_n^D$ , respectively, and<sup>2</sup>

$$\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) = \mathcal{N}(\mathbf{x}_n^G; \mathbf{m}_n^G, \mathbf{V}_n^G) \mathcal{N}(\mathbf{x}_n^D; \mathbf{m}_n^D, \mathbf{V}_n^D). \quad (32)$$

From the property of the multiplication of Gaussian functions (5)-(7),  $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) \propto \mathcal{N}(\mathbf{x}; \mathbf{c}, \mathbf{C})$ , where

$$\mathbf{C} = (\mathbf{V}^{-1} + \frac{1}{\sigma^2} \mathbf{H}^H \mathbf{H})^{-1} \quad (33)$$

$$\mathbf{c} = \mathbf{C}(\mathbf{V}^{-1} \mathbf{m} + \frac{1}{\sigma^2} \mathbf{H}^H \mathbf{r}). \quad (34)$$

Then we have

$$\begin{aligned} h(\mathbf{x}_n^D) &\propto \frac{\int_{\mathbf{x}_n^G} \mathcal{N}(\mathbf{x}; \mathbf{c}, \mathbf{C}) d\mathbf{x}_n^G}{\mathcal{N}(\mathbf{x}_n^D; \mathbf{m}_n^D, \mathbf{V}_n^D)} \\ &\propto \frac{\mathcal{N}(\mathbf{x}_n^D; \mathbf{S}_n \mathbf{c}, \mathbf{S}_n \mathbf{C} \mathbf{S}_n^T)}{\mathcal{N}(\mathbf{x}_n^D; \mathbf{m}_n^D, \mathbf{V}_n^D)} \end{aligned} \quad (35)$$

$$\propto \mathcal{N}(\mathbf{x}_n^D; \mathbf{z}_n, \mathbf{Z}_n) \quad (36)$$

<sup>2</sup>Here we note that the multiplication of Gaussian functions in (32) is straightforward as there are no common elements between  $\mathbf{x}_n^D$  and  $\mathbf{x}_n^G$ , which is different from that in (5).

where (35) follows (4), and from (8)-(10),

$$\mathbf{Z}_n = \left( (\mathbf{S}_n \mathbf{C} \mathbf{S}_n^T)^{-1} - (\mathbf{V}_n^D)^{-1} \right)^{-1} \quad (37)$$

$$\mathbf{z}_n = \mathbf{Z}_n \left( (\mathbf{S}_n \mathbf{C} \mathbf{S}_n^T)^{-1} \mathbf{S}_n \mathbf{c} - (\mathbf{V}_n^D)^{-1} \mathbf{m}_n^D \right). \quad (38)$$

Finally, the approximate APP of  $x_n$  in (29) can be represented as

$$\tilde{P}(x_n | \mathbf{r}) \propto \sum_{\tilde{\mathbf{x}}_n^D} P(\mathbf{x}_n^D) \exp \left[ -(\mathbf{x}_n^D - \mathbf{z}_n)^H \mathbf{Z}_n^{-1} (\mathbf{x}_n^D - \mathbf{z}_n) \right]. \quad (39)$$

The partial Gaussian approximation approach is summarized as follows:

- **Step 1.** Calculate the mean vector  $\mathbf{m}$  and (diagonal) covariance matrix  $\mathbf{V}$  of  $\mathbf{x}$ , and then  $\mathbf{c}$  and  $\mathbf{C}$  using (33) and (34).
- **Step 2.** Choose the  $M$  most important symbols for each  $x_n$  based on the  $n$ th column vector of  $\mathbf{H}^H \mathbf{H}$ , and calculate  $\mathbf{z}_n$  and  $\mathbf{Z}_n$  for each  $x_n$  using (37) and (38).
- **Step 3.** Calculate the approximate APP for each  $x_n$  using (39).

The computational complexity involved in Step 1 is  $\mathcal{O}(N^3)$ . As it can be shared by all the symbols, the complexity per symbol is  $\mathcal{O}(N^2)$ . The complexity of Step 2 and that of Step 3 depend on  $M$ , which are  $\mathcal{O}((M+1)^3)$  and  $\mathcal{O}((M+1)^2 |A|^{(M+1)})$  per symbol (noting that  $\mathbf{Z}_n^{-1}$  needed in Step 3 is already available in Step 2), respectively. If  $M$  is much smaller than  $N$  (e.g.,  $M = 2$  will be used in our simulations), the complexity of the algorithm can be significantly lower than that of exact APP approach ( $\mathcal{O}(N|A|^N)$  per symbol).

It is obvious that the exact APP approach is a special case of the proposed approach with  $M = N - 1$ . In this case, Steps 1 and 2 are not needed. When  $M = 0$  (i.e., all the symbols except  $x_n$  are treated as Gaussian), the proposed approach is equivalent to the well-known LMMSE detector in [2] and [3] (and the proof will be given in our full paper).

#### D. Implementation for Circulant Matrix $\mathbf{H}$

In some scenarios,  $\mathbf{H}$  can be a circulant matrix, e.g., model (11) represents a system in which a cyclic prefixed data block is transmitted through an ISI channel [11]. In this case, the complexity of the proposed approach can be significantly reduced further.

A useful property of a circulant matrix is that it can be diagonalized by the discrete Fourier transform (DFT) matrix, i.e.,  $\mathbf{F} \mathbf{H} \mathbf{F}^H = \mathbf{D}$ , or equivalently

$$\mathbf{H} = \mathbf{F}^H \mathbf{D} \mathbf{F} \quad (40)$$

where  $\mathbf{H}$  is a  $N \times N$  circulant matrix,  $\mathbf{F}$  is a normalized DFT matrix (i.e., its  $(m, n)$ -th element is given by  $N^{-1/2} e^{-j2\pi mn/N}$ , where  $j = \sqrt{-1}$ ), and  $\mathbf{D}$  is a diagonal matrix whose diagonal elements are given by  $\sqrt{N} \mathbf{F} \mathbf{h}_1$ , where  $\mathbf{h}_1$  is the first column vector of  $\mathbf{H}$ . As in the implementation



of the LMMSE detector in the frequency domain, we approximate the diagonal covariance matrix  $\mathbf{V}$  to be a scaled identity matrix (e.g., in [8] and [10]), i.e.,

$$\mathbf{V} \approx \alpha \mathbf{I}_N \quad (41)$$

where  $\alpha$  is the average of the diagonal elements of  $\mathbf{V}$ .

1) *Implementation of Step 1:* With (40) and (41),  $\mathbf{C}$  in (33) can be rewritten as

$$\mathbf{C} = \mathbf{F}^H \Lambda \mathbf{F} \quad (42)$$

where  $\Lambda = (\alpha^{-1} \mathbf{I}_N + \frac{1}{\sigma^2} \mathbf{D}^H \mathbf{D})^{-1}$  is a diagonal matrix. Hence  $\mathbf{C}$  is a circulant and Hermitian matrix and its first column vector is given by

$$\rho = N^{-1/2} \mathbf{F}^H \eta \quad (43)$$

where the length- $N$  vector  $\eta$  consists of the diagonal elements of  $\Lambda$ . Vector  $\mathbf{c}$  in (34) can be represented as

$$\mathbf{c} = \mathbf{F}^H \Lambda \left( \alpha^{-1} \mathbf{F} \mathbf{m} + \frac{1}{\sigma^2} \mathbf{D}^H \mathbf{F} \mathbf{r} \right). \quad (44)$$

By using the FFT algorithm, the complexity of Step 1 can be reduced to  $\mathcal{O}(\log N)$  per symbol.

2) *Implementation of Step 2:* We can find that  $\mathbf{H}^H \mathbf{H}$  whose  $n$ th column is used to select the most important  $M$  symbols to  $x_n$  is also a circulant and Hermitian matrix. Hence its first column vector can be found using the FFT algorithm (its other columns are cyclic shifts of the first column). As  $\mathbf{C}$  is a circulant Hermitian matrix, it is not hard to verify that  $\mathbf{S}_n \mathbf{C} \mathbf{S}_n^T$  in (37) is independent of  $n$ . We use the approximation that  $\mathbf{V}_n^D \approx \alpha \mathbf{I}_{M+1}$ , so  $\mathbf{Z}_n$  in (37) is also independent of  $n$ . This means that  $(\mathbf{S}_n \mathbf{C} \mathbf{S}_n^T)^{-1}$  and  $\mathbf{Z}_n$  only need to be calculated once and can be shared by all the  $N$  symbols. The complexity of calculating  $\mathbf{z}_n$  is  $\mathcal{O}((M+1)^2)$ . The total complexity of this step is  $\mathcal{O}(\log N) + \mathcal{O}((M+1)^2)$  per symbol.

We can see that the main complexity of the whole approach is on Step 3, which is  $\mathcal{O}((M+1)^2 |\mathcal{A}|^{(M+1)})$  per symbol. To make the complexity low, we can set  $M$  to be a small integer. In Section IV, we will show that the proposed detector with  $M = 2$  can achieve significant performance gain over the frequency domain LMMSE detector (which is equivalent to the proposed detector with  $M = 0$ ).

#### IV. SIMULATION RESULTS

Consider a single-carrier block transmission system over ISI channels, where the cyclic prefixing technique is used, i.e.,  $\mathbf{H}$  is a circulant matrix. We assume that a rate-1/2 nonsystematic convolutional code with generator  $(5, 7)_8$  is employed, and the APP decoder is implemented using the BCJR algorithm. The sequence of the code bits is interleaved and mapped to a symbol sequence using QPSK with Gray mapping. Then the symbol sequence is divided into length-512 blocks, which are cyclic prefixed before transmission over an ISI channel. The number of the information bits is 4096, and so is the length of symbol sequence, i.e., the symbol sequence consists of 8 blocks. The number of iterations is 10. For the proposed detector, we set  $M = 2$ .

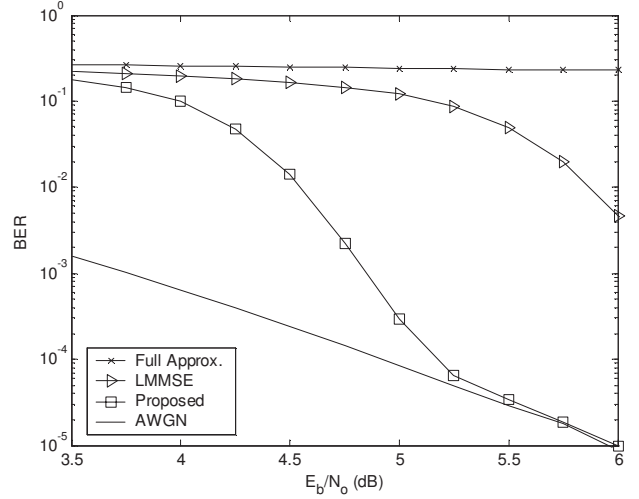


Fig. 2. BER performance of the turbo receiver over Proakis' 5-tap ISI channel.

We use Proakis' 5-tap ISI channel with coefficients  $[0.227, 0.460, 0.688, 0.460, 0.227]$  [14], which is widely used in the literature to test the performance of a detector. It generates severe distortion on the transmitted signal as shown in [14] by the frequency response of the channel. The bit error rate (BER) performance of the turbo receiver with different detectors is shown in Fig. 2. We can see from this figure that the detector with full Gaussian approximation (denoted by "Full Approx.") delivers very poor performance. It can also be seen that the receiver with the proposed detector significantly outperforms the frequency domain LMMSE detector (which is equivalent to the proposed detector with  $M = 0$ ), and approaches closely the AWGN bound (the performance of the system over AWGN channel) in the relatively high  $E_b/N_0$  range.

#### V. CONCLUSIONS

We have proposed a soft-in soft-out detector with partial Gaussian approximation, in which part of the symbols are treated as Gaussian variables to reduce the computational complexity. The exact APP detector and the well-known LMMSE detector are its special cases. Simulation results demonstrate that the proposed detector with a small  $M$  can achieve significant performance gain over the LMMSE detector, and provide a good trade-off between complexity and performance.

#### REFERENCES

- [1] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference : Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, no. 5, pp.507-511, Sep.-Oct. 1995.
- [2] M. Tüchler, A. C. Singer, R. Koetter, "Minimum mean squared error equalization using *a priori* information," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 673-683, Mar. 2002.
- [3] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 1046-1061, Jul. 1999.

- [4] C. Laot, A. Glavieux, and J. Labat, "Turbo equalization: Adaptive equalization and channel decoding jointly optimized," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1744-1752, Sep. 2001.
- [5] Q. Guo and Li Ping, "LMMSE turbo equalization based on factor graphs," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 311-319, Feb. 2008.
- [6] Q. Guo and D. Huang, "A concise representation for the soft-in soft-out LMMSE detector," *IEEE Commun. Lett.*, vol. 15, no. 5, pp. 566-568, May 2011.
- [7] L. Liu and Li Ping, "An extending window MMSE turbo equalization algorithm," *IEEE Signal Processing Lett.*, vol. 11, pp. 891-894, Nov. 2004.
- [8] H. Liu and P. Schniter, "Iterative frequency-domain channel estimation and equalization for single-carrier transmissions without cyclic-prefix," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3686-3691, Oct. 2008.
- [9] A. Gusmão, P. Torres, R. Dinis, and N. Esteves, "A turbo FDE technique for reduced-CP SC-based block transmission systems," *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 16-20, Jan. 2007.
- [10] Q. Guo, Li Ping, and D. Huang, "A low-complexity iterative channel estimation and detection technique for doubly selective channels," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4340-4349, Aug. 2009.
- [11] M. Tüchler and J. Hagenauer, "Turbo equalization using frequency domain equalizers," in *Proc. Allerton Conf.*, Monticello, IL, Oct. 2000, pp. 1234-1243.
- [12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.
- [13] 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.
- [14] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGrawHill, 1995.
- [15] S. Roweis, "Gaussian identities," available at <http://www.cs.nyu.edu/~roweis/notes/gaussid.pdf>.