Soft-in soft-out detection using partial Gaussian approximation

GUO, Qinghua; HUANG, Defeng (David); NORDHOLM, Sven; XI, Jiangtao; LI, Ping

Published in:
IEEE Access

Published: 01/01/2014

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2014.2322078

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 07/06/2024
Soft-In Soft-Out Detection Using Partial Gaussian Approximation

QINGHUA GUO1,2, (Member, IEEE), DEFENG (DAVID) HUANG2, (Senior Member, IEEE), SVEN NORDHOLM3, (Senior Member, IEEE), JIANGTAO XI1, (Senior Member, IEEE), AND LI PING4, (Fellow, IEEE)
1School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
2School of Electrical, Electronic and Computer Engineering, University of Western Australia, Perth, WA 6009, Australia
3Department of Electrical and Computer Engineering, Curtin University, Perth, WA 6102, Australia
4Department of Electronic Engineering, City University of Hong Kong, Hong Kong
Corresponding author: Q. Guo (qguo@uow.edu.au)
The work of Q. Guo, D. Huang, S. Nordholm, and J. Xi was supported in part by the Australian Research Council’s DECRA under Grant DE120101266 and in part by the Discovery Project under Grant DP1093000 and Grant DP110100736. The work of Li Ping was supported by the Research Grants Council, Hong Kong, under Project CityU 118014. Part of this work was presented in 2012 International Conference on Wireless Communications and Signal Processing (WCSP 2012).

ABSTRACT This paper concerns the soft-in soft-out detection in a coded communication system, where the transmitted symbols are discrete valued, and the exact a posteriori probability (APP) detection often involves prohibitive complexity. By using the properties of Gaussian functions, an approximate approach to the APP detection is devised with the idea that, in the computation of the APP of each symbol, the remaining symbols are distinguished based on their contributions to the APP of the concerned symbol, and the symbols with less contributions are approximated as (continuous) Gaussian variables [hence the name partial Gaussian approximation (PGA)] to reduce the computational complexity. The connection between the PGA detector and the reduced dimension maximum a posteriori detector (RDMAP) is investigated. It is shown that, PGA is equivalent to RDMAP, but it has a complexity much lower than that of RDMAP, i.e., PGA can be regarded as an efficient implementation of RDMAP. In addition, the application of PGA in intersymbol interference (ISI) channel equalization is also investigated. We show that PGA allows further significant complexity reduction by exploiting the circulant structure of the system transfer matrix, which makes PGA very attractive in handling severe ISI channels with large memory length.

INDEX TERMS Soft-in soft-out detection, turbo equalization, Gaussian functions, LMMSE detection, RDMAP.

I. INTRODUCTION
Recently, the implementation of the soft-in soft-out (SISO) detector for iterative detection in a coded communication system has received tremendous attention as a turbo (iterative) receiver can achieve huge performance gain over a conventional non-iterative receiver [1]–[15]. 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 which work 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 from the decoder as a priori information [1]–[15].

Although we focus on iterative detection, the developed detector in this work can also be employed in a non-iterative receiver.

The optimal implementation of the detection is the a posteriori probability (APP) detector [1]. However, the exponential complexity of the APP detector is often too
high for practical use. A low-complexity alternative is the linear minimum mean square error (LMMSE) detector [2]–[5]. The complexity of a LMMSE detector is quadratic in the length of the transmitted symbol vector or in the length of inter-symbol interference (ISI) channels in equalization, which is typically significantly lower than that of the APP one. If the system transform matrix is circulant, the LMMSE detector can be implemented with the fast Fourier transform (FFT), reducing the complexity to logarithmic level [10]–[16]. The LMMSE detection is attractive due to its low-complexity compared with the APP counterpart. However, the LMMSE approach may suffer from performance loss in handling severe ISI channels. This motivates the investigation of approximate APP detection to achieve a better trade-off between performance and complexity.

The key task of the SISO APP detector is to compute the APP for each symbol, and the high complexity is due to the high-dimensional summation over the discrete-valued symbols, which grows exponentially with the symbol number. To overcome this problem, a reduced-dimension maximum a posteriori detector (RDMAP) was proposed in [19] with a focus on multiple input multiple output (MIMO) detection. In RDMAP, the summation is carried over only a subset of the symbols, and the remaining symbols are treated as interference. After interference cancelation and whitening, the APPs are calculated with the assumption that the output noise of the whitening filter is Gaussian. RDMAP allows a flexible trade-off between complexity and performance by adjusting the subset size. However, the complexity of RDMAP may still be a serious concern, especially when the size of the system transfer matrix is large. Such situation may arise in ISI channel equalization.

In this work, we present an approximate APP approach that is equivalent to RDMAP (as revealed through a tedious proof in this paper), but with both entirely different derivation and completely different form of the final result. This approach offers an efficient implementation of RDMAP, which is referred to, for the convenience of discussion, as partial Gaussian approximation detection (PGA).

Different from RDMAP, PGA is derived by reformulating the exact APP representation of a transmitted symbol into a new form (see Eqn. (13)) and using the partial Gaussian approximation. Suppose that the number of the symbols involved is $N$. In the computation of the APP of each symbol, we distinguish the remaining $N - 1$ symbols based on their contributions to the APP of the symbol. We select a subset of $M$ ($M \leq N - 1$) 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 (hence the name PGA). PGA approximates the original high-dimension summation over the $N - 1$ discrete random variables as marginalization over $M$ discrete random variables and $N - 1 - M$ Gaussian random variables.

As shown in Section III.C, PGA can be regarded as an efficient implementation of RDMAP for a general system transfer matrix, e.g., in MIMO detection. With much lower complexity compared with RDMAP, PGA enjoys the same performance advantage as RDMAP in MIMO detection in [19]. Moreover, we will also show another appealing feature of PGA, i.e., it allows a further significant complexity reduction in ISI channel equalization by exploiting the circulant structure of the system transfer matrix. This makes PGA very attractive in handling severe ISI channels with large memory length.

This paper is organized as follows. Problem description and the exact APP detector are introduced in Section II. In Section III, PGA is derived by using some useful properties of the Gaussian functions listed in Appendix A, and the connection between RDMAP and PGA is investigated. The implementation of PGA for a circulant system transfer matrix in ISI channel equalization is detailed in Section IV. Simulation results are provided in Section V, followed by conclusions in Section VI.

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. We use $\propto$ to denote equality of functions up to a scale factor, and use $I_N$ to denote an $N \times N$ identity matrix. The superscriptions “$^T$” and “$^H$” denote the transpose and conjugate transpose, respectively. We use $\mathcal{N}(\mathbf{x}, \mathbf{m}, \mathbf{V})$ to denote a complex multivariate Gaussian function with $\mathbf{x}$ as variable and $\mathbf{m}$ and $\mathbf{V}$ as parameters, where $\mathbf{V}$ is a Hermitian matrix.

II. THE EXACT APP SISO DETECTOR
As shown in Fig. 1, the turbo receiver consists of a SISO detector and a SISO decoder, which are connected by an interleaver and its corresponding deinterleaver. The SISO detector and the SISO decoder exchange extrinsic information iteratively to achieve performance improvement with the iteration [1]–[15]. We focus on the implementation of the SISO detector in this work.

A. PROBLEM DESCRIPTION
The received signal in the coded system shown in Fig. 1 can be represented as

$$\mathbf{r} = \mathbf{Hx} + \mathbf{w}$$

(1)

where $\mathbf{r}$ denotes a length-$Z$ observation vector, $\mathbf{H}$ denotes a $Z \times N$ system transfer matrix (e.g., a MIMO channel matrix or an ISI channel 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 $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ is mapped from an interleaved code sequence $\mathbf{c}$, i.e., each $x_n \in \mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_{2^L}\}$ ($|\mathcal{A}| = 2^L$) corresponds to a length-$Q$ subsequence of $\mathbf{c}$ denoted by $\mathbf{c}_n = [c_{n,1}, c_{n,2}, \ldots, c_{n,Q}]^T$. 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]–[5]

$$L(c_{n,q}) = \ln \frac{P(c_{n,q} = 0 | r)}{P(c_{n,q} = 1 | r)} + \sum_{x_n \in A_q^0} P(x_n | r) = \ln \frac{\sum_{x_n \in A_q^0} P(x_n | r)}{\sum_{x_n \in A_q^1} P(x_n | r)}$$  \hspace{1cm} (2)

where $A_q^0 (A_q^1)$ denotes the subset of all $a_i \in A$ 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]–[5]

$$L^e(c_{n,q}) = L(c_{n,q}) - L^a(c_{n,q})$$  \hspace{1cm} (3)

will be the 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 $P(x_n | r)$ for each symbol $x_n$, which is the focus of this work.

**B. THE EXACT APP DETECTOR**

According to Bayes’ rule,

$$P(x_n | r) = \sum_{x_n} P(x | r) \propto \sum_{x_n} P(x)p(r | x)$$  \hspace{1cm} (4)

where the length-$(N - 1)$ vector $\mathbf{x}_a$ consists of the elements of $x$ except $x_n$. Given $x$, $r$ is Gaussian distributed, i.e., $p(r | x) = \mathcal{N}(r; Hx, \sigma^2 \mathbf{I})$. Hence (4) can be rewritten as

$$P(x_n | r) \propto \sum_{x_n} P(x) \exp \left[ -\frac{(r - Hx)^H(r - Hx)}{\sigma^2} \right].$$  \hspace{1cm} (5)

In (5), assuming that the interleaved code bits are independent of each other, we have [1]–[5]

$$P(x) = \prod_{i} P(x_i)$$  \hspace{1cm} (6)

$$P(x_i) = \prod_{q} P(c_{i,q})$$  \hspace{1cm} (7)

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

By using (5), we can calculate the APPs $P(x_n = a_i | r), \forall i$ 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 (2) as it will be cancelled out. Due to the high-dimensional summation involved in (5), the computational complexity of the APP detector is $O(N |A|^N)$ per symbol (assuming that $Z$ and $N$ are in the same order), which is prohibitive even for a moderate $N$.

Note that, when (1) is used to model the impact of an ISI channel $h = [h_0, h_1, \ldots, h_L]^T$ on the transmitted signal $x$, i.e. $r$ is the convolution of $x$ and $h$, corrupted by AWGN $w$, the system transfer matrix $H$ is a Toeplitz matrix. In this case, the APP detector can be implemented using the BCJR algorithm with complexity $O(|A|^2)$ [18]. 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 computation of $P(x_n | r)$.

**III. APPROXIMATE APP DETECTION USING PARTIAL GAUSSIAN APPROXIMATION**

**A. AN ALTERNATIVE REPRESENTATION FOR THE EXACT APP (5)**

Define vector $a = (H^H H)^{-1} H^H r$. It is easy to verify that

$$(r - Hx)^H(r - Hx) = (x - a)^H H^H H(x - a) - r^H (H^H H)^{-1} H^H - I_Z) r.$$  \hspace{1cm} (8)

As a result,

$$\exp \left[ -\frac{(r - Hx)^H(r - Hx)}{\sigma^2} \right] = \varphi g(x)$$  \hspace{1cm} (9)

where

$$g(x) = \exp \left[ -\frac{(x - a)^H H^H H(x - a)}{\sigma^2} \right]$$  \hspace{1cm} (10)

and

$$\varphi = \exp \left[ \frac{-r^H (H^H H)^{-1} H^H - I_Z)r}{\sigma^2} \right]$$  \hspace{1cm} (11)

is independent of $x$.

Now we reformulate the APP in (5) using (9). Noting that the likelihood function $p(r | x)$ in (4) is a function of $x$ (rather than $r$ which is fixed), we can safely ignore $\varphi$ in (9). Hence (5) can be rewritten as

$$P(x_n | r) \propto \sum_{x_n} P(x) g(x)$$  \hspace{1cm} (12)

$$\propto \sum_{x_n} P(x) \mathcal{N}(x; a, A)$$  \hspace{1cm} (13)

where $A = \sigma^2 (H^H H)^{-1}$. To obtain (13), we have used

$$g(x) \propto \mathcal{N}(x; a, A).$$  \hspace{1cm} (14)

Here, we emphasize that $\mathcal{N}(x; a, A)$ in (13) and (14) just denotes a function of $x$ as shown in (46), and it does not mean that $x$ is a Gaussian random vector.

Although the exact computation of the APP $P(x_n | r)$ using (13) still involves high-dimensional summation, it enables us to derive low-complexity approximate approaches based on the properties of Gaussian functions listed in Appendix A.

**B. PARTIAL GAUSSIAN APPROXIMATION**

If we treat $x$ as a Gaussian random vector with (a priori) PDF $\mathcal{N}(x; \mathbf{m}, V)$ (instead of a discrete random vector with a priori probability mass function $P(x)$), the summation in (13) is changed to integration accordingly. So we have

$$P_{FGA}(x_n | r) \propto \int_{\mathbf{x}_a} \mathcal{N}(x; \mathbf{m}, V) \mathcal{N}(x; a, A) d\mathbf{x}_a$$  \hspace{1cm} (15)
where we note that $P_{\text{PGA}}(x_n | r)$ is an approximation to $P(x_n | r)$ due to the above Gaussian approximation. In (15), the a priori covariance matrix $V$ is a diagonal matrix as $\{x_i\}$ are independent of each other. A straightforward way to determine $m_i$ (the $i$th element of $m$) and $v_i$ (the $i$th diagonal element of $V$) is to use the first and second moments of $x_i$, i.e.,

$$m_i = \sum_{\alpha_i \in A} \alpha_i P(x_i = \alpha_k)$$

(16)\n
$$v_i = \sum_{\alpha_i \in A} |\alpha_i - m_i|^2 P(x_i = \alpha_k).$$

(17)\n
Although (15) involves high-dimensional integration, it can be efficiently calculated based on the properties of Gaussian functions shown by (50)–(52) and (49) in Appendix A. The complexity is about $O(N^2)$ per symbol (assuming that $N$ and $Z$ are in the same order), which is much lower than that of calculating (5) exactly with complexity $O(N|A|^N)$. As all the symbols are treated as Gaussian variables in this approach, we call it full Gaussian approximation approach.\n
Simulation results in Section V show that the full Gaussian approximation approach delivers very poor performance. This motivates us to develop the partial Gaussian approximation approach, i.e., only part of the symbols are approximated as Gaussian variables.

To devise PGA, a primary question we need to answer is which symbols should be approximated as Gaussian variables. It is interesting that the alternative representation for the APP $P(x_n | r)$, i.e., (12) or (13) developed in Section III.A, provides a direct suggestion for us.

Now we re-examine (12) and (10). By expanding the exponent of $g(x)$ in (10) to a scalar form, it can be found that some of the non-zero off-diagonal elements of $H^T H$ generate the cross terms between $x_n$ and $\{x_i, i \neq n\}$. It is obvious that if all of these cross terms are zeroes (i.e., $H^T H$ is a diagonal matrix), $\{x_i, i \neq n\}$ 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 $p_n$ as the $n$th column vector of matrix $H^T H$. We can find that the $i$th element of $p_n$ 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 $p_n$, 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

$$x^D_n = S_n x$$

(18)\n
where $S_n$ 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 $x^G_n$, and their means and variances can be computed using (16) and (17). We denote the mean vector and (diagonal) covariance matrix of $x^G_n$ by $m^G_n$ and $V^G_n$, respectively.

Based on the above partial Gaussian approximation, part of the summation in (13) will be changed to integration, and the APP of $x_n$ can be approximated as

$$P_{\text{PGA}}(x_n | r) \propto \sum_{x^G_n} P(x^G_n) h(x^D_n)$$

(19)\n
where $x^D_n$ consists of the elements in $x^D_n$ except $x_n$, and

$$h(x^D_n) = \int \mathcal{N}(x^G_n; m^G_n, V^G_n) \mathcal{N}(x; a, A) dx^G_n$$

(20)\n
It can be seen that, assuming $h(x^D_n)$ is available, (19) 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 $h(x^D_n)$ in (20).

Note that (20) involves the multiplication of two Gaussian functions with different dimensions (noting that $x^G_n$ is a sub-vector of $x$), and hence (50)–(52) in Appendix A cannot be directly applied. To overcome this problem, we rewrite $h(x^D_n)$ as

$$h(x^D_n) = \frac{\mathcal{N}(x^G_n; m^D_n, V^D_n)}{\mathcal{N}(x^D_n; m^D_n, V^D_n)} \int \mathcal{N}(x; m, V) \mathcal{N}(x; a, A) dx^G_n$$

(21)\n
where $m^D_n$ and $V^D_n$ are the mean vector and (diagonal) covariance matrix of $x^D_n$, respectively, and

$$\mathcal{N}(x; m, V) = \mathcal{N}(x^G_n; m^G_n, V^G_n) \mathcal{N}(x^D_n; m^D_n, V^D_n).$$

(22)\n
From the property of the multiplication of Gaussian functions (50)–(52), $\mathcal{N}(x; m, V) \mathcal{N}(x; a, A) \propto \mathcal{N}(x; c, C)$, where

$$C = (V^{-1} + \frac{1}{\sigma^2} H^T H)^{-1},$$

(23)\n
$$c = C(V^{-1} m + \frac{1}{\sigma^2} H^T r).$$

(24)\n
Then from (21) we have

$$h(x^D_n) \propto \frac{\mathcal{N}(x^D_n, m^D_n, V^D_n)}{\mathcal{N}(x^D_n; m^D_n, V^D_n)} \propto \frac{\mathcal{N}(x^G_n; m^G_n, V^G_n)}{\mathcal{N}(x^G_n; m^G_n, V^G_n)} \propto \mathcal{N}(x^G_n; S_n a, S_n C S_n^T)$$

(25)\n
(26)\n
\footnote{We point out that the full Gaussian approximation approach is not equivalent to the well-known LMMSE approach. In the LMMSE approach, the concerned symbol $x_n$ is not treated to be Gaussian. Later, we will show that the LMMSE approach is a special case of PGA (see Proposition 1).}

\footnote{Here we note that the multiplication of Gaussian functions in (22) is straightforward as there are no common elements between $x^D_n$ and $x^G_n$, which is different from that in (50).}
Finally, by using (26), the approximate APP of \( x_n \) in (19) can be represented as
\[
P_{\text{PGA}}(x_n | r) \propto \sum_{\tilde{x}^0_n} P(x^0_n) \exp \left[ -\left( x^0_n - z_n \right)^H \mathbf{Z}_n^{-1} \left( x^0_n - z_n \right) \right].
\]
(29)

The steps of PGA are 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 (23) and (24).

- **Step 2.** Choose the \( M \) most important symbols for each \( x_n \) based on the \( n \)-th column vector of \( \mathbf{H}_n \mathbf{D} \), and calculate \( \mathbf{z}_n \) and \( \mathbf{Z}_n \) for each \( x_n \) using (27) and (28).

- **Step 3.** Calculate the approximate APP for each \( x_n \) using (29).

Before analyzing the complexity of PGA, we note that different \( x_n \) has different \( \mathbf{x}^D_n \) and \( \mathbf{x}^G_n \), and hence different \( \mathbf{Z}_n \) and \( \mathbf{z}_n \). That is why we put the subscript \( n \) for these variables. The computational complexity involved in Step 1 is \( \mathcal{O}(N^3 + 2N^2) \) or \( \mathcal{O}(N^3) \) with the assumption that \( Z \) and \( N \) are in the same order. We note that the results of Step 1 can be shared by all the symbols, so the complexity per symbol is \( \mathcal{O}(N^3) \). The complexity of Step 2 and that of Step 3 depend on the value of \( M \), which are \( \mathcal{O}(M + 1)^3 \) and \( \mathcal{O}(M + 1)^2 |A|^{(M+1)} \) per symbol (noting that \( \mathbf{Z}_n^{-1} \) required 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 the exact APP approach (\( \mathcal{O}(N|A|^N) \) per symbol).

It can be easily found that the exact APP approach is a special case of PGA with \( M = N - 1 \) (i.e., without Gaussian approximation). The well-known LMMSE detector in [2], [3], and [5] is also its special case with \( M = 0 \) as shown by Proposition 1.

**Proposition 1:** When \( M = 0 \) (i.e., all the symbols except \( x_n \) are treated to be Gaussian), PGA is equivalent to the LMMSE detector in [2], [3], and [5].

**Proof:** See Appendix B.

**C. CONNECTION WITH RDMAP IN [19]**

In [19], the original model (11) is rewritten in the following form with respect to each \( x_n^4 \)
\[
r = \mathbf{H}_n^A \mathbf{x}^A_n + \mathbf{H}_n^B \mathbf{x}^B_n + w
\]
(30)

where elements in \( \mathbf{x} \) are grouped into \( \mathbf{x}^A_n \) and \( \mathbf{x}^B_n \), \( x_n \) is included in \( \mathbf{x}^A_n \), and \( \mathbf{H}_n^A \) and \( \mathbf{H}_n^B \) consist of the column vectors in \( \mathbf{H} \).

which correspond to \( \mathbf{x}^A_n \) and \( \mathbf{x}^B_n \), respectively. \( \mathbf{x}^A_n \) is called reduced-dimension signal vector and \( \mathbf{x}^B_n \) is the interference vector. In RDMAP, to minimize the noise enhancement due to noise whitening, the elements in \( \mathbf{x}^A_n \) and \( \mathbf{x}^B_n \) are also determined by the channel correlation matrix \( \mathbf{H}_n^A \mathbf{H}_n^A \), which results in \( \mathbf{x}^A_n = \mathbf{x}^D_n \) and \( \mathbf{x}^B_n = \mathbf{x}^S_n \). Therefore we use
\[
r = \mathbf{H}_n^D \mathbf{x}^D_n + \mathbf{H}_n^G \mathbf{m}_n^G + w
\]
(31)
to replace (30) to facilitate the investigation of the connection between PGA and RDMAP.

In [19], the interference and noise \( \mathbf{H}_n^G \mathbf{m}_n^G + \mathbf{w} \) is treated as a colored noise with mean \( \mathbf{H}_n^G \mathbf{m}_n^G \) and covariance matrix
\[
\mathbf{R}_n = \mathbf{H}_n^G \mathbf{V}_n \mathbf{H}_n^G \mathbf{H}_n^G + \sigma^2 \mathbf{I}.
\]
(32)
The colored noise is whitened by removing its mean and then applying the noise-whitening filter given by
\[
\mathbf{W}_n = \Sigma_n^{-1/2} \mathbf{Q}_n^H
\]
(33)
where \( \Sigma_n \) and \( \mathbf{Q}_n \) are obtained from the eigenvalue decomposition of \( \mathbf{R}_n \)
\[
\mathbf{R}_n = \mathbf{Q}_n \Sigma_n \mathbf{Q}_n^H.
\]
(34)
i.e., \( \Sigma \) is a diagonal matrix and
\[
\Sigma_n \mathbf{Q}_n^H = \mathbf{I}.
\]
(35)
The whitened observation can be represented as
\[
\mathbf{p}_n = \mathbf{W}_n (r - \mathbf{H}_n^G \mathbf{m}_n^G).
\]
(36)
The residual noise at the filter output can be expressed as
\[
\mathbf{W}_n (\mathbf{H}_n^G \mathbf{x}_n^G - \mathbf{H}_n^G \mathbf{m}_n^G + \mathbf{w}) = \mathbf{p}_n - \mathbf{W}_n \mathbf{H}_n^D \mathbf{x}_n^D.
\]
(37)
Under the assumption that the noise is Gaussian, i.e.,
\[
\mathbf{p}_n - \mathbf{W}_n \mathbf{H}_n^D \mathbf{x}_n^D \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),
\]
(38)
the APP of \( x_n \) can be represented as,
\[
P_{\text{RDMAP}}(x_n | r) \propto \sum_{\tilde{x}^0_n} P(x^0_n) \exp \left[ -||\mathbf{p}_n - \mathbf{W}_n \mathbf{H}_n^D \mathbf{x}_n^D ||^2 \right]
\]
(39)
where \( \tilde{x}^0_n \) consists of the elements of \( x^D_n \) except \( x_n \). Once the right hand side of (39) is available, the LLRs can be calculated using (2). RDMAP needs to evaluate (39) for each \( x_n \).

As we can see from the above that the derivation for PGA is very different from that for RDMAP, and the final results, i.e., (29) for PGA and (39) for RDMAP are in completely different forms. However, in RDMAP, the whitened interference plus noise \( \mathbf{W}_n (\mathbf{H}_n^G \mathbf{x}_n^G - \mathbf{H}_n^G \mathbf{m}_n^G + \mathbf{w}) \) is assumed to be Gaussian, which can be reached by treating \( \mathbf{x}_n^G \) as a Gaussian vector as well. As a result, PGA and RDMAP have the same fundamental assumption that part of the elements in \( \mathbf{x} \) (i.e., \( \mathbf{x}_n^G \)) are treated as Gaussian variables and the remaining elements in \( \mathbf{x} \) (i.e., \( \mathbf{x}_n^D \)) are kept as discrete random variables without approximation. It can be shown that these two approaches are equivalent through a tedious proof in Appendix C.
Proposition 2: PGA is equivalent to RDMAP.

Proof: See Appendix C.

Remarks:
- Although PGA is equivalent to RDMAP, the complexity of PGA is much lower than that of RDMAP. Note that, in RDMAP, the noise-whitening filter $W_n$ is constructed based on the eigenvalue decomposition of $R_n$, and $W_n$ needs to be computed for each $x_n$ with complexity $O(N^3)$ per symbol. It can be shown that, the total complexity of RDMAP is $O(N^3) + O(N(N-M-1)|A|^{M+1})$ per symbol. In contrast, the total complexity of PGA is $O(N^2) + O((M+1)^2|A|^{M+1})$, which is significantly lower than that of RDMAP especially when $N$ is large and $M$ is much smaller than $N$. Hence, PGA can be regarded as an efficient implementation of RDMAP.
- It is interesting that both approaches use the channel correlation matrix $H^H H$ to determine the subsets $x_n^D$ and $x_n^G$ for each $x_n$, but the arguments are very different. The argument in RDMAP is minimizing the noise enhancement due to the noise whitening. In contrast, the fact that $H^H H$ appears in the new APP representation (12) (see $g(x)$ in (10)) in PGA directly suggests the use of $H^H H$ to distinguish the contributions of $x_n$ to the APP of $x_n$, thereby determining the subsets $x_n^D$ and $x_n^G$.
- In RDMAP, $x_n^G$ is treated as interference to $x_n^D$, and interference cancellation and whitening are used in (36) to handle the impact of the interference. The proof of the equivalence between RDMAP and PGA in Appendix C reveals that the same interference cancellation and whitening operations are hidden in PGA. With (71) and (78) in Appendix C, $z_n$ in PGA (see (28) and (29)) can be represented as

$$z_n = z_n (H_n^D)^H R_n^{-1} (r - H_n^G m_n^G)$$  
$$= z_n (H_n^D)^H Q_n \Sigma_n^{-1/2} W_n (r - H_n^G m_n^G).$$  

where (33) and (34) are used in the above derivation. The term $W_n (r - H_n^G m_n^G)$ in (40) indicates the hidden operations of interference cancellation and whitening in PGA.

IV. IMPLEMENTATION FOR CIRCULANT H

The previous section reveals that (for a general system transfer matrix) PGA is equivalent to RDMAP but with much lower complexity. However, in ISI channel equalization where the size of the system transfer matrix may be very large (i.e., $N$ is very large), the complexity of PGA (and RDMAP) in the previous section may still be a serious concern.

With the the cyclic prefixing technique, $H$ in model (1) turns into a circulant matrix [15], [16]. In the following, we will show another appealing feature of PGA, i.e., it allows a further significant complexity reduction by exploiting the circulant structure of $H$. In contrast, it is difficult to exploit the circulant structure in RDMAP.

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

$$H = F^H DF$$  

where $H$ is an $N \times N$ circulant matrix, $F$ is a normalized DFT matrix (i.e., its $(m, n)$-th element is given by $N^{-1/2} \delta_{m-n}\{j\}$, where $j = \sqrt{-1}$), $D$ is a diagonal matrix whose diagonal elements are given by $\sqrt{N}F_{h1}$, and $h_1$ is the first column vector of $H$. As in the implementation of the LMMSE detector in the frequency domain, we approximate the diagonal covariance matrix $V$ to be a scaled identity matrix (see [11], [13], [22]), i.e.,

$$V \approx \alpha I_N$$

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

1) IMPLEMENTATION OF STEP 1

With (41) and (42), $C$ in (23) can be rewritten as

$$C = F^H \Lambda F$$

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

$$\rho = N^{-1/2} F^H \eta$$

where the length-$N$ vector $\eta$ consists of the diagonal elements of $\Lambda$. Vector $c$ in (24) can be represented as

$$c = F^H \left( \alpha^{-1} Fm + \frac{1}{\sigma^2} D^H F \eta \right).$$

By using the FFT algorithm, the complexity of Step 1 can be reduced from $O(N^2)$ per symbol to $O(\log N)$ per symbol.

2) IMPLEMENTATION OF STEP 2

We can find that $H^H H$ 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 $C$ is a circulant Hermitian matrix, it is not hard to verify that $S_n C_n F_n^H$ in (27) is independent of $n$. We use the approximation that $V_n^D \approx \alpha I_{M+1}$, so $Z_n$ in (27) is also independent of $n$. This means that $(S_n C_n F_n^H)^{-1}$ and $Z_n$ only need to be calculated once and can be shared by all the $N$ symbols. The complexity of calculating $Z_n$ is $O((M+1)^2)$. The total complexity of this step is $O((log N) + O((M+1)^2))$ per symbol.

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

PGA can be applied to MIMO detection (assuming a general channel matrix \( \mathbf{H} \)) and ISI channel equalization. However, we focus on ISI channel equalization in this section as PGA is equivalent to RDMAP (i.e., PGA enjoys the same performance as RDMAP but with much lower complexity) and the simulations about MIMO detection and comparisons with other detectors such as the LMMSE detector and group MAP detector have been provided in [19].

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. In all the simulations the number of iterations is 10. For PGA, we set \( M = 2 \).

Example A: In this example, we use the Proakis’ 5-tap ISI channel with coefficients [0.227, 0.460, 0.688, 0.460, 0.227] (noting that the energy of the channel is 1), which incurs severe distortion on the transmitted signal as shown in [20] by the frequency response of the channel. The number of the information bits is set to 4096. The 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. The MAP detector (equivalent to PGA with \( M = N - 1 \)) delivers the best performance at the cost of high complexity. It can also be seen that the receiver with PGA significantly outperforms the frequency domain LMMSE detector (which is equivalent to PGA 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.

Example B: In this example, we test the detector using the power line reference channel 3 (REF-CH3), which is rated as a “bad” channel in [23]. The parameters of REF-CH3 are given in [23]. The transfer function of REF-CH3 is shown in Fig. 3, where we restrict the maximum frequency to 8.5 MHz due to the high attenuation of the channel with the increase of the frequency. The impulse response of the channel is represented by 128 channel coefficients. The energy of the ISI channel is normalized to 1 to facilitate performance comparison against the AWGN bound. The length of the information sequence is set to 7680. The BER performance is shown in Fig. 4. It can be seen that the full Gaussian approximation approach again exhibits poor performance. Compared with the LMMSE detector, PGA achieves a significant performance gain.

VI. CONCLUSION

We have presented an approximate APP detector with the partial Gaussian approximation, in which part of the symbols
are treated as Gaussian variables to reduce the computational complexity. We also revealed that PGA is equivalent to RDMAP in [19] but with much lower complexity, i.e., PGA can be regarded as an efficient implementation of RDMAP. We have shown that, by exploiting the circulant structure of the channel matrix, the complexity of PGA can be significantly further reduced, making it very attractive in severe ISI channel equalization. Simulation results demonstrated that PGA can achieve significant performance gain over the LMMSE detector, and provide a good trade-off between complexity and performance.

**APPENDIX A**

**PROPERTIES OF GAUSSIAN FUNCTIONS**

We use \( \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) \) to denote a complex multivariate Gaussian function with \( \mathbf{x} \) as variable and \( \mathbf{m} \) and \( \mathbf{V} \) as parameters, i.e.,

\[
\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) = \frac{1}{\pi^\frac{N}{2} \det(\mathbf{V})} \exp\left[-\frac{(\mathbf{x} - \mathbf{m})^H \mathbf{V}^{-1} (\mathbf{x} - \mathbf{m})}{2}\right]
\]

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

\[
p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V})
\]

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

\[
\mathbf{x}_s = \mathbf{S} \mathbf{x}
\]

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 \( \mathbf{x}_s \)

\[
p(\mathbf{x}_s) = \int \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) d\mathbf{x}_r
\]

\[
= \mathcal{N}(\mathbf{x}_s; \mathbf{S} \mathbf{m}, \mathbf{S} \mathbf{V} \mathbf{S}^H)
\]

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 [21], i.e.,

\[
\mathcal{N}(\mathbf{x}; \mathbf{m}_a, \mathbf{V}_a) \mathcal{N}(\mathbf{x}; \mathbf{m}_b, \mathbf{V}_b) \propto \mathcal{N}(\mathbf{x}; \mathbf{m}_c, \mathbf{V}_c)
\]

where

\[
\mathbf{V}_c = (\mathbf{V}_a^{-1} + \mathbf{V}_b^{-1})^{-1}
\]

\[
\mathbf{m}_c = \mathbf{V}_c (\mathbf{V}_a^{-1} \mathbf{m}_a + \mathbf{V}_b^{-1} \mathbf{m}_b).
\]

From the above, we have

\[
\mathcal{N}(\mathbf{x}; \mathbf{m}_c, \mathbf{V}_c) \propto \mathcal{N}(\mathbf{x}; \mathbf{m}_b, \mathbf{V}_b)
\]

and

\[
\mathbf{V}_b = (\mathbf{V}_c^{-1} - \mathbf{V}_a^{-1})^{-1}
\]

\[
\mathbf{m}_b = \mathbf{V}_b (\mathbf{V}_c^{-1} \mathbf{m}_c - \mathbf{V}_a^{-1} \mathbf{m}_a).
\]

**APPENDIX B**

**PROOF OF PROPOSITION 1**

**Proof:** When \( M = 0 \), \( \mathbf{S}_n \) is a row vector (i.e., the \( n \)th row of the identity matrix \( \mathbf{I}_N \)), and \( \mathbf{x}_n^D \) only contains \( x_n \).

In this case, \( \mathbf{z}_n \) and \( \mathbf{z}_n \) in (27) and (28) are both scalars, and can be represented as

\[
\mathbf{z}_n = \frac{\nu_n c_n}{\nu_n - c_n} \mathbf{H}_n - \nu_n \mathbf{C}_n
\]

where \( c_n \) is the \( n \)th diagonal element of \( \mathbf{C} \) and \( c_n \) is the \( n \)th element of \( e \). Using the matrix inversion lemma [17], and (23) and (24), we have

\[
c_n = v_n - v_n^2 h_n^H A^{-1} h_n
\]

\[
v_n - c_n m_n^H A^{-1} (\mathbf{r} - \mathbf{H} m + m_n \mathbf{h}_n)
\]

where \( h_n \) is the \( n \)th column of \( \mathbf{H} \), and \( \mathbf{A} = \mathbf{H} \mathbf{V} \mathbf{H}^H + \sigma^2 \mathbf{I} \). Substituting (58) and (59) into (56) and (57), we have

\[
\mathbf{z}_n = \frac{1 - v_n h_n^H A^{-1} h_n}{h_n^H A^{-1} h_n}
\]

As \( \mathbf{x}_n^D = x_n \), and \( \mathbf{z}_n \) and \( \mathbf{z}_n \) are both scalars, (29) can be rewritten as

\[
P_{PGA}(\mathbf{x}_n | \mathbf{r}) \propto P(\mathbf{x}_n) \exp\left[-\frac{|\mathbf{x}_n - \mathbf{z}_n|^2}{\mathbf{z}_n} \right]
\]

\[
\propto P(\mathbf{x}_n) \exp\left[-\frac{|h_n^H A^{-1} (\mathbf{r} - \mathbf{H} m + m_n \mathbf{h}_n) - x_n h_n^H A^{-1} h_n|^2}{h_n^H A^{-1} h_n - v_n (h_n^H A^{-1} h_n)^2} \right].
\]

From (62) and (2), it is easy to show that the extrinsic LLR \( L^e(c_n,q) \) in (3) and that in [2] are exactly the same.

**APPENDIX C**

**PROOF OF PROPOSITION 2**

**Proof:** Eqn (39) in RDMAP can be rewritten as

\[
P_{RDMAP}(\mathbf{x}_n | \mathbf{r}) \propto \sum_{\mathbf{x}_n^D} P(\mathbf{x}_n^D) \exp\left[-(\mathbf{r}_n - \mathbf{H}_n^G \mathbf{x}_n^G)^H \mathbf{R}_n^{-1} (\mathbf{r}_n - \mathbf{H}_n^G \mathbf{x}_n^G) \right]
\]

where

\[
\mathbf{r}_n = \mathbf{r} - \mathbf{H}_n^G \mathbf{m}_n^G.
\]
By comparing (63) with (29), we need to prove that
\[
\exp \left[ -\left( x_n^D - z_n \right)^H Z_n^{-1} \left( x_n^D - z_n \right) \right] \propto \exp \left[ -\left( r_n - H_n^D x_n^D \right)^H R_n^{-1} \left( r_n - H_n^D x_n^D \right) \right]
\]
with respect to \( x_n^D \). It is not hard to show that
\[
\exp \left[ -\left( r_n - H_n^D x_n^D \right)^H R_n^{-1} \left( r_n - H_n^D x_n^D \right) \right] \propto \exp \left[ -\left( x_n^D - e_n \right)^H \left( H_n^D \right)^H R_n^{-1} H_n^D \left( x_n^D - e_n \right) \right]
\]
with respect to \( x_n^D \), where
\[
e_n = \left( H_n^D \right)^H H_n^{-1} \left( H_n^D \right)^H \left( r_n - H_n^D x_n^D \right).
\]
From (65) and (66), we only need to prove that
\[
Z_n^{-1} = \left( H_n^D \right)^H R_n^{-1} H_n^D
\]
and
\[
z_n = e_n.
\]
Now we prove (68). Recall the definition of matrix \( C \) in (23). Using the matrix inversion lemma [17], we have
\[
S_n CS_n^T = S_n \left( V^{-1} + \frac{1}{\sigma^2} H^H H \right)^{-1} S_n^T
\]
\[
= S_n V S_n^T - S_n V H^H (H V H^H + \sigma^2 I)^{-1} H V S_n^T
\]
\[
= \left( H_n^D \right)^H H_n^{-1} \left( H_n^D \right)^H \left( r_n + H_n^D v_n \right) \left( H_n^D \right)^H \left( H_n^D \right)^H - \left( H_n^D \right)^H H_n^{-1} H_n^D v_n
\]
\[
= \left( H_n^D \right)^H \left( r_n - H_n^D x_n^D \right) \left( H_n^D \right)^H
\]
\[
= \left( S_n CS_n^T \right)^{-1} - V_n^D V_n^D
\]
\[
= \left( H_n^D \right)^H \left( r_n - H_n^D x_n^D \right) \left( H_n^D \right)^H
\]
Next, we prove (69). For the simplicity of notation, we define matrix \( A = H V H^H + \sigma^2 I \). With the matrix inversion lemma [17], we have
\[
\left( H_n^D \right)^H R_n^{-1} = \left( H_n^D \right)^H \left( A - H_n^D v_n \right) \left( H_n^D \right)^H \left( V_n^D \right)^{-1}
\]
\[
= \left( H_n^D \right)^H A^{-1} \left( H_n^D \right)^H \left( V_n^D \right)^{-1} - \left( H_n^D \right)^H A^{-1} \left( H_n^D \right)^H \left( H_n^D \right)^H A^{-1}
\]
\[
= \left( V_n^D \right)^{2} v_n \left( H_n^D \right)^H A^{-1} \left( H_n^D \right)^H \left( V_n^D \right)^{-1}
\]
\[
= \left( S_n CS_n^T \right)^{-1} S_n V H^H A^{-1}
\]
\[
= \left( S_n CS_n^T \right)^{-1} S_n V H^H A^{-1}
\]
Using the matrix inversion lemma [17] also yields
\[
V H^H A^{-1} = \frac{1}{\sigma^2} C H^H
\]
Substituting (73) to (72), we have
\[
\left( H_n^D \right)^H R_n^{-1} = \frac{1}{\sigma^2} \left( S_n CS_n^T \right)^{-1} S_n C H^H
\]
We use \( P_n \) to denote the selecting matrix for \( x_n^G \), i.e., \( P_n x = x_n^G \). Then, we have
\[
S_n C P_n^T S_n V H^H A^{-1} H V P_n^T = -S_n V H^H A^{-1} H V P_n^T
\]
where we note that \( S_n V P_n^T = 0 \). It is not hard to show that
\[
S_n C V^{-1} m = S_n C \left[ S_n^T P_n^T \right] \left( V P_n^T - 1 m_P^G \right)
\]
Based on (72), (75), and (76), we have
\[
\left( S_n CS_n^T \right)^{-1} S_n C V^{-1} m - \left( V P_n^T - 1 m_P^G \right)
\]
\[
= \left( S_n CS_n^T \right)^{-1} S_n C P_n^T \left( V P_n^T - 1 m_P^G \right)
\]
\[
= \left( S_n CS_n^T \right)^{-1} \left( S_n C \right) \left( V P_n^T - 1 m_P^G \right)
\]
\[
= \left( H_n^D \right)^H R_n^{-1} H_n^D v_n
\]
Using (24), (28), (68), (74), and (77), we have
\[
z_n = z_n \left( \left( S_n CS_n^T \right)^{-1} S_n C \left( V^{-1} m + \frac{1}{\sigma^2} H^H r \right) - \left( V P_n^T - 1 m_P^G \right) \right)
\]
\[
= \left( H_n^D \right)^H R_n^{-1} H_n^D v_n
\]
\[
= z_n
\]
REFERENCES
DEFENG (DAVID) HUANG (M’01–S’02–M’05–SM’07) received the B.E.E.E. and M.E.E.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1996 and 1999, respectively, and the Ph.D. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2004. He is currently a Professor with the School of Electrical, Electronic and Computer Engineering, University of Western Australia, Perth, WA, Australia. He serves as an Editor of the IEEE Wireless Communications Letters, and also served as an Editor of the IEEE Transactions on Wireless Communications (2005 to 2011).

QINGHUA GUO (S’07–M’08) received the B.E. degree in electronic engineering and the M.E. degree in signal and information processing from Xidian University, Xi’an, China, in 2001 and 2004, respectively, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2008. He is currently with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW, Australia, and also with the School of Electrical, Electronic, and Computer Engineering, University of Western Australia, Perth, WA, Australia. His research interests include statistical signal processing and telecommunication. He was a recipient of the Australian Research Council’s inaugural Discovery Early Career Researcher Award.

JIAHAN LI (S’87–M’91–SM’06–F’10) received the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1990. He was a Lecturer with the Department of Electronic Engineering, University of Melbourne, Parkville, VIC, Australia, from 1990 to 1992, and was a Research Staff with Telecom Australia Research Laboratories, Melbourne, Australia, from 1993 to 1995. Since 1996, he has been with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, where he is currently a Chair Professor of Information Engineering. He was a recipient of the Institution of Electrical Engineers J. J. Thomson Premium in 1993, the Croucher Foundation Award in 2005, and the British Royal Academy of Engineering Distinguished Visiting Fellowship in 2010. He served as a member of the Board of Governors for the IEEE Information Theory Society from 2010 to 2012.

SVEN NORDHOLM (SM’04) is a Professor with Curtin University, Bentley, WA, Australia. He received the M.Sc.E.E. (Civilingenjör) degree, the Licentiate of Engineering degree, and the Ph.D. degree in signal processing from Lund University, Lund, Sweden, in 1983, 1989, and 1992, respectively.

He has been with the Curtin University of Technology, Perth, WA, since 1999. From 1999 to 2002, he was the Director of the Australian Telecommunications Research Institute and a Professor with Curtin University of Technology. From 2002 to 2009, he was the Director of the Signal Processing Laboratory, Western Australian Telecommunication Research Institute, Perth, a joint institute between the University of Western Australia and Curtin University. Since 2009, he has been a Professor of Electrical and Computer Engineering at Curtin University. From 2012 to 2013, he served as the Head of the Department. He has also been the Chief Scientist and Co-Founder of a startup company Sensear, Perth, which provides voice communication in extreme noise conditions. He is also the Founder and Director of a hearing aid company Hearmore, Perth. He is an Associate Editor of EURASIP Journal on Advances in Signal Processing and Journal of the Franklin Institute. He is a member of the IEEE Technical Committee for Audio and Acoustic Signal Processing.