A Concise Representation for the Soft-in Soft-out LMMSE Detector

Qinghua Guo, Member, IEEE, and Defeng (David) Huang, Senior Member, IEEE

Abstract—An alternative simple derivation for the well-known soft-in soft-out LMMSE (linear minimum mean square error) detector in a turbo system is presented. The derivation leads to a concise representation for the LMMSE detector in terms of the extrinsic means and extrinsic variances of the data symbols, and provides new perspectives for its implementation.

Index Terms—Turbo detection, MMSE estimation.

I. INTRODUCTION

The turbo principle, originated from the decoding of the turbo codes [1], has been applied in iterative joint equalization/multuser detection and decoding to achieve impressive performance gain in coded communication systems [2]-[5]. A turbo receiver consists of a soft-in soft-out (SISO) detector and a SISO decoder (or a bank of decoders in the multuser case), between which extrinsic information in the form of extrinsic log-likelihood ratios (LLRs) is passed iteratively [2]-[5]. The well-known linear minimum mean square error (LMMSE) SISO detector [3]-[5] provides a good tradeoff between performance and complexity (refer to [4] and [5] for coded systems with BPSK (binary phase shift keying) modulation, and [3] for high-order modulations).

In the original derivation for the LMMSE detector [3]-[5], the instantaneous LMMSE filtering is first performed, where the a priori information of the concerned symbol is excluded in the computation of the instantaneous filter coefficients, then the output of the LMMSE filter is approximated to be Gaussian to obtain the extrinsic LLRs for the code bits carried by the symbol.

In this letter, we present an alternative derivation for the LMMSE detector in a coded system with high-order modulations. With the Gaussian assumption of the data symbols, we first perform the standard MMSE estimation to obtain the a posteriori estimates of the data symbols, then get the extrinsic mean and extrinsic variance of each symbol by taking its estimates of the data symbols, then get the extrinsic a priori information out from its a posteriori information from its a posteriori estimate. The extrinsic LLRs for the code bits associated with a symbol are calculated using the extrinsic mean and extrinsic variance of the symbol. It can be proved that our derivation leads to the same result as that in [3] (see Proposition 1). However, our derivation is more concise and much simpler. Moreover, it leads to a concise representation for the LMMSE detector, and provides new perspectives for its implementation.

Notations: Lower and upper case letters denote scalars. Bold lower (upper) case letters represent column vectors (matrices). $\mathcal{CN}(\mathbf{x}; \mathbf{m}, \mathbf{V})$ denotes a complex multivariate Gaussian probability density function (PDF) [6] of $\mathbf{x}$ with $\mathbf{m}$ as its mean vector and $\mathbf{V}$ as its covariance matrix. The notation $\propto$ denotes equality of functions up to a scale factor, Re$[x]$ (Im$[x]$) denotes the real (imaginary) part of $x$, and $p(\cdot)$ ($P(\cdot)$) denotes the PDF (probability mass function) of a continuous (discrete) random variable.

II. PRELIMINARY

A. Gaussian Linear Model and MMSE Estimation

Consider a complex Gaussian linear model

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

where $\mathbf{H}$ is an $M \times N$ complex matrix, $\mathbf{x}$ is a length-$N$ complex random vector with PDF $\mathcal{CN}(\mathbf{x}; \mathbf{m}, \mathbf{V})$, and $\mathbf{w}$ is a length-$M$ complex random vector with PDF $\mathcal{CN}(\mathbf{w}; \mathbf{0}, 2\sigma^2\mathbf{I})$ and is independent of $\mathbf{x}$. The a posteriori PDF of $\mathbf{x}$ is [6]

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{CN}(\mathbf{x}; \mathbf{m}^p, \mathbf{V}^p)$$

where the a posteriori mean vector $\mathbf{m}^p$, i.e., the MMSE estimate of $\mathbf{x}$, is given by [6]

$$\mathbf{m}^p = \mathbf{m} + \frac{1}{2\sigma^2} \mathbf{V}^H \mathbf{H}^H (\mathbf{z} - \mathbf{Hm})$$

(2a)

and the a posteriori covariance matrix $\mathbf{V}^p$ is given by

$$\mathbf{V}^p = \left( \mathbf{V}^{-1} + \frac{1}{2\sigma^2} \mathbf{H}^H \mathbf{H} \right)^{-1}.$$ (2b)

It may be sometimes more convenient to use the following expressions for computing $\mathbf{m}^p$ and $\mathbf{V}^p$ [6]

$$\mathbf{m}^p = \mathbf{m} + \mathbf{V}^H \mathbf{V}^{-1}_z (\mathbf{z} - \mathbf{Hm})$$

(3a)

$$\mathbf{V}^p = \mathbf{V} - \mathbf{V}^H \mathbf{V}^{-1}_z \mathbf{H} \mathbf{V}$$

(3b)

where $\mathbf{V}_z = \mathbf{H} \mathbf{V}^H + 2\sigma^2\mathbf{I}$. By using the matrix inversion lemma, we can convert (2) into (3), and vice versa.

B. Multiplication of Gaussian Functions

It is straightforward to verify that the multiplication of two Gaussian functions is another scaled Gaussian function, i.e.,

$$\mathcal{CN}(x; m_1, v_1) \cdot \mathcal{CN}(x; m_2, v_2) \propto \mathcal{CN}(x; m, v)$$

Manuscript received October 28, 2010. The associate editor coordinating the review of this letter and approving it for publication was T. Tsiftsis. The authors are with the School of EECE, The University of Western Australia, Australia (e-mail: {qghuo, huangdf}@ee.uwa.edu.au). This work was supported by the Australian Research Council’s Discovery Project DP1093000.

1The derivations for BPSK and QPSK modulations are its special cases.
2Modeling data symbols as Gaussian in communication systems has been used long time ago, e.g., in [10].
where
\[ v = (1/v_1 + 1/v_2)^{-1} \]
\[ m = v (m_1/v_1 + m_2/v_2). \]

From the above, we have
\[ \frac{\mathcal{CN}(x; m, v)}{\mathcal{CN}(x; m_1, v_1)} \propto \mathcal{CN}(x; m_2, v_2) \]  \hspace{1cm} (4)

where
\[ v_2 = (1/v - 1/v_1)^{-1} \]
\[ m_2 = v_2 (m/v - m_1/v_1). \] \hspace{1cm} (5)

The above can be extended to multivariate Gaussian functions.

III. AN ALTERNATIVE DERIVATION FOR THE LMMSE DETECTOR

A. System Description

At the transmitter, the information bits are first encoded into a code sequence and interleaved, yielding an interleaved code sequence \( e \). Then \( e \) is broken into a number of length-\( Q \) subsequences \( \{e_n = [e_{n,1}, e_{n,2}, \ldots, e_{n,Q}]^T\} \), and each \( e_n \) is mapped to a symbol \( x_n \in \mathcal{X} \) with a binary labeling map \( \mathcal{M} : \{0,1\}^Q \rightarrow \mathcal{X} \). Here, \( \mathcal{X} = \{\alpha_i, i = 1, 2, \ldots, 2^Q\} \) denotes a \( 2^Q \)-ary symbol alphabet with \( \sum_{i=1}^{2^Q} \alpha_i = 0 \) and \( 2^{-Q} \sum_{i=1}^{2^Q} |\alpha_i|^2 = 1 \), and each element \( \alpha_i \in \mathbb{C} \) corresponds to a binary vector (label) \( s_i = [s_{i,1}, s_{i,2}, \ldots, s_{i,Q}]^T \). The received signal vector \( z \) can be represented as
\[ z = Hx + w \] \hspace{1cm} (7)

where \( H \) denotes the channel matrix, the additive complex white Gaussian noise vector \( w \) has the same PDF as that in model (1), but the transmitted symbol vector \( x = [x_1, x_2, \ldots, x_N]^T \) is not a Gaussian random vector.

Based on \( z \) and the \( a \ priori \) LLRs \( \{L(c_{n,q}), \forall n \text{ and } \forall q\} \) fed back from the decoder, the SISO detector computes the following extrinsic LLR for each code bit \( c_{n,q} \) \[ L^e(c_{n,q}) = \ln \frac{P(c_{n,q} = 0 | z)}{P(c_{n,q} = 1 | z)} - L(c_{n,q}). \] \hspace{1cm} (8)

B. A Concise Derivation for the SISO LMMSE Detector

Let \( X^0_q \) and \( X^1_q \) denote the subset of all \( \alpha_i \in \mathcal{X} \) whose label in position \( q \) (i.e., \( s_{i,q} \)) has the value of 0 and 1, respectively. We can rewrite (8) as [3]-[5]
\[ L^e(c_{n,q}) = \ln \sum_{x_n \in X^0_q} \frac{p(z | e_n = s_i)}{p(z | e_n = s_i)} = \ln \sum_{x_n \in X^0_q} \frac{p(z | e_n = s_i)}{p(z | e_n = s_i)} - L(c_{n,q}) \]
\[ = \ln \sum_{x_n \in X^0_q} \frac{p(z | x_n)}{p(z | x_n)} = \ln \sum_{x_n \in X^0_q} \frac{p(z | x_n)}{p(z | x_n)}. \] \hspace{1cm} (9)

Note that the model (7) also applies in a multiple access system. In this case, \( H \) and \( x \) may be represented as \( H = [H_1, H_2, \ldots, H_K] \) and \( \mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_K^T]^T, \) where \( K \) is the number of users, and \( H_k \) and \( x_k \) denote the channel matrix and the symbol vector of user \( k \), respectively.

In (9), \( P(c_{n,q} = s_{i,q}) \) can be easily calculated based on \( L(c_{n,q}) \) as in [3], and the key issue is how to compute \( p(z | x_n) \).

The \( a \ priori \) mean \( m_n \) and the \( a \ priori \) variance \( v_n \) can be calculated based on the \( a \ priori \) LLRs of the code bits as in [3]. In the following, we pretend that each \( x_n \) is a complex Gaussian variable with PDF
\[ p(x_n) = \mathcal{CN}(x_n; m_n, v_n). \] \hspace{1cm} (10)

Due to the use of the interleaver, \( \{e_n\} \) are approximately independent of each other. Hence \( x \) is a complex random vector with PDF \( \mathcal{CN}(x; m, V) \), where \( m = [m_1, m_2, \ldots, m_N]^T \) and \( V = \text{diag}[v_1, v_2, \ldots, v_N] \). From Section II.A, \( p(x | z) = \mathcal{CN}(x; m^P, V^P) \), so we have
\[ p(x_n | z) = \mathcal{CN}(x_n; m^P_n, v^P_n) \] \hspace{1cm} (11)
where \( m^P_n \) is the \( n \)th entry of \( m^P \), and \( v^P_n \) is the \( n \)th diagonal entry of \( V^P \). To find out \( p(z | x_n) \), we use the following link between \( p(z | x_n) \) and \( p(z | x) \),
\[ p(z | x_n) = \frac{p(z | x_n | z)}{p(z | x_n)} \] \hspace{1cm} (12)

We note that in (12), for a fixed \( z, p(z) \) is a constant, and \( p(z | x_n) \) is a function of \( x_n \). Since \( p(x_n | z) \) and \( p(x_n) \) are two complex Gaussian functions of \( x_n \) (see (10) and (11)), from (4)-(6), we have
\[ p(z | x_n) \propto \mathcal{CN}(x_n; m^e_n, v^e_n) \] \hspace{1cm} (13)

where
\[ v^e_n = (1/v_n^P - 1/v_n)^{-1} \]
\[ m^e_n = v^e_n (m^P_n / v_n^P - m_n / v_n). \] \hspace{1cm} (14)

We call \( m^e_n \) and \( v^e_n \) the extrinsic mean and the extrinsic variance of \( x_n \), respectively. Then (9) can be represented as
\[ L^e(c_{n,q}) = \ln \sum_{\alpha_i \in X^0_q} \exp \left( -\frac{|\alpha_i|^2}{v^e_n} \right) \sum_{q' \neq q} P(c_{n,q'} = s_{i,q'}) \]
\[ = \ln \sum_{\alpha_i \in X^0_q} \exp \left( -\frac{|\alpha_i|^2}{v^e_n} \right) \sum_{q' \neq q} P(c_{n,q'} = s_{i,q'}). \] \hspace{1cm} (16)

Although the above derivation is very different from that in [3], we have the following proposition.

**Proposition 1:** \( L^e(c_{n,q}) \) given in (16) in this letter is equal to that given in (8) in [3]. In other words, they lead to an identical result.

**Proof:** See Appendix.

The operations involved in the proposed SISO LMMSE detector are summarized as follows:

- **Step 1.** With the Gaussian assumption, perform the standard MMSE estimation to find \( m^P \) and \( \{v^P_n\} \) (i.e., the diagonal entries of \( V^P \)).
Step 2. Calculate the extrinsic mean and the extrinsic variance for each symbol using (14) and (15).

Step 3. Calculate the extrinsic LLR for each code bit using (16).

C. Discussions

Compared with the derivation in [3]-[5], our derivation is more straightforward and much simpler. Moreover, it leads to a more concise representation (16) (compared with (8) in [3]) for the LMMSE detector.

If quadrature phase shift keying (QPSK) with Gray mapping is employed, the extrinsic LLRs of the two code bits carried by $x_n$ can be represented as

$$
L^e(c_{n,1}) = 2\sqrt{2} \text{Re} [m_n^e / v_n^e],
$$

$$
L^e(c_{n,2}) = 2\sqrt{2} \text{Im} [m_n^e / v_n^e].
$$

The above derivation also applies to a real system with BPSK modulation. We only need to replace complex Gaussian PDFs with real ones (noting that there will be a factor 2 before $v_n^e$ in (16)). The output extrinsic LLR reduces to

$$
L^e(c_{n,1}) = 2m_n^e / v_n^e
$$

which coincides with the result in [8].

The core of the LMMSE detector (see the three steps) presented in this letter is the standard MMSE estimation. One can implement the MMSE estimation directly using (2) or (3) with complexity $O(N^2)$ per symbol.\(^5\) To reduce the complexity, we can use the sliding window approach similar to [3]-[5] with complexity $O(P^2)$ per symbol ($P$ is the window size), i.e., we can estimate $x_n$ (compute $m_n^e$ and $v_n^e$) one by one using (18) in the Appendix where the matrix inverse involved is calculated recursively. However, performance loss may be incurred if $P$ is small. The following approaches can be employed to efficiently implement the standard MMSE estimator.\(^6\)

- The Kalman smoothing approach [7] or the Gaussian message passing approach as in [8] by formulating the original model (7) into a state-space form. This approach reduces the complexity to $O(L^2)$ per symbol ($L$ is the number of channel taps).

- The cyclic prefixing technique. This technique leads to an $N \times N$ circulant matrix $H$, which can be diagonalized by the discrete Fourier transform matrix $F$, i.e., $H = F^H DF$, where $D$ is a diagonal matrix. Similar to [3], we may approximate the diagonal covariance matrix $V$ to be a scaled identity matrix $\mu I$ where $\mu$ is the average of the diagonal elements of $V$. Based on the above, it is not hard to show that the MMSE estimator can be efficiently implemented using FFT (fast Fourier transform) with complexity $O(\log N)$ per symbol.

\(^5\)We assume that $N$ and $M$ are on the same order. Note that, when (3) is employed, we only need to calculate the diagonal entries of $V_p$ in (3b).

\(^6\)We only focus on the single user case, but the approaches can be extended to the multiuser case.

Appendix

Proof of Proposition 1

Proof: By comparing (16) in this letter and (8) in [3], we only need to prove the following relationship between $\varphi_{n,i}$ in (8) in [3] and the extrinsic mean and extrinsic variance defined in this letter

$$
\varphi_{n,i} = \frac{|\alpha_i - m_n^e|^2}{v_n^e}.
$$

Use $h_n$ to denote the $n$th column of $H$. According to (3a) and (3b),

$$
m_n^p = m_n + v_n h_n^H V_z^{-1} (z - Hm)
$$

$$
v_n^p = v_n - v_n^e h_n^H V_z h_n
$$

then we have

$$
h_n^H V_z^{-1} (z - Hm) = (m_n^p - m_n) / v_n
$$

$$
h_n^H V_z^{-1} h_n = 1 / v_n - v_n^p / v_n^2.
$$

Using (20) and (14), we have

$$
1 - h_n^H V_z^{-1} h_n = v_n - 1 / v_n - 1 / v_n = v_n^e.
$$

By using (19), (20) and (15), it is not hard to show that

$$
\frac{h_n^H V_z^{-1} (z - Hm + m_n h_n)}{h_n^H V_z^{-1} h_n} = v_n \left( \frac{m_n^p}{v_n} - \frac{m_n}{v_n} \right) = m_n^e.
$$

Note that $\varphi_{n,i}$ in (8) in [3] can be represented as

$$
\varphi_{n,i} = \frac{|h_n^H V_z^{-1} (z - Hm + m_n h_n) - \alpha_i h_n^H V_z^{-1} h_n|^2}{|h_n^H V_z^{-1} h_n - v_n (h_n^H V_z^{-1} h_n)|^2}
$$

$$
= \frac{\alpha_i - 1}{h_n^H V_z^{-1} h_n - v_n}
$$

Substituting (21) and (22) into (23), we have (17).

References


