ROBUST ITERATIVE PRUNED-TREE DETECTION AND LDPCC DECODING

by

Xinde Hu

BS, Zhejiang University, 2002

Submitted to the Graduate Faculty of

the The School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2004

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

Xinde Hu

It was defended on

April 9th 2004

and approved by

Heung-no Lee, Assistant Professor, Department of Electrical Engineering

L. F. Chaparro, Associate Professor, Department of Electrical Engineering

M. McCloud, Assistant Professor, Department of Electrical Engineering

Thesis Advisor: Heung-no Lee, Assistant Professor, Department of Electrical Engineering

ROBUST ITERATIVE PRUNED-TREE DETECTION AND LDPCC DECODING

Xinde Hu, M.S.

University of Pittsburgh, 2004

A novel sub-optimal low-complexity equalization and turbo-iterative decoding scheme based on running the sum-product algorithm on an aggressively pruned tree is proposed in this paper for use in a multiple transmit and receive antenna (MIMO) system operating over severe frequency-selective fading inter-symbol interference (ISI) channels. The receiver deals with the issue of signal processing complexity which with a full-search equalization grows with power-law, $M^{N_t\,L}$, where M is the M-ary channel symbol, and N_t is the the number of transmit antennas, and L is the number of delay channel-taps. The sum-product algorithm is applied to the pruned tree which is constructed by two main operations, a sphere list detection and a threshold-based tree search algorithms. At a particular node of the tree, only a number of most probable branches in the tree of hypothetical symbols are expanded and included in the list of candidates; at a particular tree-section, all but some of most probable candidates are pruned. This pruned tree takes the soft input and generates the soft output, and is utilized in the turbo-iterative manner with the decoder of the low-density parity check

code. We obtained the approximated error probability using the pair-wise error calculation averaged over the fading ensemble, and use it to bound our simulation results. Our current simulation results are obtained for MIMO systems up to four transmit and four receive antennas, using 4-QAM symbols. They indicate the proposed receiver performs extremely well. The proposed transceiver system is ideal for a system of higher spectral efficiency with even larger signal constellations. Adopting Hassbi-Vikalo's framework, we provide a method which enables a quick evaluation of the signal processing complexity required in the proposed algorithm at a given set of system parameters, M, N_t, N_r.

Keywords: MAP, turbo-iteration, reduced complexity, joint decoding and equalization, MIMO, LDPC codes, wireless communication.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

Since the Telatar's [7] and Foschini-Gans's [16] landmark works on the capacity of the multi-input and multi-output fading channels, the design of wireless communication system utilizing multiple antennas at both sides, the transmitter and the receiver, became very popular, and a large body of publications with regard to the enabling transceiver that attempts to attain the capacity closely became available in the literature today. As indicated in a very recent paper by Zheng-Tse [1], this additional resource, especially with the availability of the number of transmit antennas, can in fact be utilized in either directions to achieving the spectral efficiency benefit or the diversity-benefit.

The transmitter-receiver pair discussed in this paper, uses a turbo-iterative equalization and decoding scheme, with the powerful low-density parity-check block codes. The transceiver can be used to attain any point in the capacity-diversity trade-off region of the multiple input multiple output (MIMO) system with a relatively simple change of system parameters. Some major features of this enabling transceiver include the capability of handling the growing signal processing complexity while maintaining the performance as close as possible to the theoretical bounds, and the capability of dealing with the frequency-selective channel (due to the multi-path delay-dispersion)

which leads to severe inter-symbol interference. In particular, we focus on the problem of reducing the signal processing complexity consumed at the stage of equalization which generates soft-input and soft-output (SISO) messages. This SISO equalization is combined in turbo-iterative fashion with the graph decoders for the low-density parity-check code.

The delay dispersion is modeled as a finite impulse response (FIR) channel with L Rayleigh fading taps. This channel is depicted well in the right of Fig 1. There are N_t transmit and N_r receive MIMO antenna systems for delay dispersive channel environment.

In [17], a low complexity decoding and equalization scheme based on a novel signal separation and per-antenna equalization receiver utilizing the turbo-iteration is proposed. We noticed however, this per-antenna scheme is still incurring a large amount of computations and comparisons. The complexity measured in terms of a number of states in the ISI-trellis is increasing exponentially fast as the memory of the channel increases. That is, the number of states in the ISI trellis is $M^{N_t \times (L-1)}$, where M is the size of the signal constellation and the number of trellis-edges is $M^{N_t \times L}$. For the proposed low-complexity per-antenna equalization, the number of per-antenna trellis states is only M^{L-1}. However, the complexity in signal separation part is still $O(M^{N_t \times (L-1)})$ – the same as the complexity of the full complexity vector maximum a posteriori (MAP) algorithm [17].

In this paper, we propose a novel reduced complexity tree-search algorithm for MIMO ISI fading channels. Full complexity search algorithms, such as Viterbi-algorithms and BCJR algorithm (MAP algorithm), makes a sequence based decision on a trellis. The common factor of these full-complexity receivers are such that all the possible sequences contribute in

making the final decision of the best possible transmitted sequence. The reduction in signal processing complexity in our case is achieved by maintaining only a small subset of the sequences and then using them for the calculation of the soft decision metric in the sum-product algorithm applied at the equalization step.

The proposed tree-search algorithm utilizes the sphere list detection [5][6] at the stage of expanding a path of a growing tree into the next level of depth. The tree is in general M^{N_t}-ary such that there are M^{N_t} candidates expanded from a single survivor path. Assuming the length of sequence is N_s. The tree construction starts from the first time-epoch, $k = 1$, to the maximum depth of the tree, i.e., $k = N_s$. At each depth of the tree, all survivor paths get expanded into the next level, forming a list of candidates for the next level. Among the list of candidates, a subset of them are selected using a simple threshold detection rule and made available as survivors to the next time-epochs. The rule insures that all but some of the most posterior-probable paths are pruned. The posterior probability is calculated from the product of the likelihood function and the priors. The priors are coming from the decoder. It should be noticed that this tree-search algorithm moves only in the forward direction, and thus it is different from the famous sequential search algorithm like the Fano-algorithm [2].

The proposed tree-search algorithm is rather very similar to the T-algorithm [4]. One of the novel feature of the proposed algorithm is in the use of simple compensation rules in the steps of the sum-product algorithm, and provides a sense of fairness among different survivor paths having different lengths in the pruned tree. The compensation rules are used when generating the soft-output messages, which are calculated from the probability of the paths.

The rest of the paper is organized as follows. Baseband equivalent system descriptions will be given in Chapter 2. In Chapter 3, we will discuss the pair-wise error probability for MIMO fading ISI channels. In Chapter 4, the overall robust equalization-decoding system is introduced. The analysis on pair-wise error probability based on the LDPC code is also given in this section. In Chapter 5, the reduced complexity equalizer using the T-algorithm and the compensation rule is presented. The sphere list detection is discussed in Chapter 6. In Chapter 7, the simulation results for both uncoded and coded transmission are showed and discussed. In Chapter 8, the analysis on system computational complexity is presented. Chapter 9 contains the conclusion.

2.0 BASEBAND EQUIVALENT SYSTEMS

The description on the multi-input multi-output (MIMO) fading inter-symbol interference (ISI) channels will be discussed in this section.

As shown in Fig.1, we use the tapped delay line model to describe the MIMO fading ISI channels where each channel tap is matrix-valued instead of scalar. From one particular transmitter to a particular receiver antenna, the channel is delay-dispersive and thus frequency-selective. This can be modelled as a tapped delay line filter with L channel taps, i.e, $L-1$ channel memories. The overall N_t transmit and N_r receive MIMO ISI channel can then be modelled as the vector-matrix tapped delay line model as shown in the right of Fig. 1.

The l-th delay taps at the k^{th} time-epoch, $\boldsymbol{h}_{k,l}, l = 0, 1..., L-1$, is an $[N_r \times N_t]$ matrix. The N_t channel symbols transmitted can be arranged in the $[N_t \times 1]$ vector $\boldsymbol{x}_k$. The N_r receive symbols can be grouped into the $[N_r \times 1]$ vector $\boldsymbol{r}_k$.

The multi-input/multi-output relationship can be written as:

$$\begin{aligned} \boldsymbol{r}_k &= \sum_{l=0}^{L-1} \boldsymbol{h}_{k,l} \boldsymbol{x}_{k-l} + \boldsymbol{n}_k \\ &= \boldsymbol{y}_k + \mathbf{n}_k, \end{aligned} \tag{2.1}$$

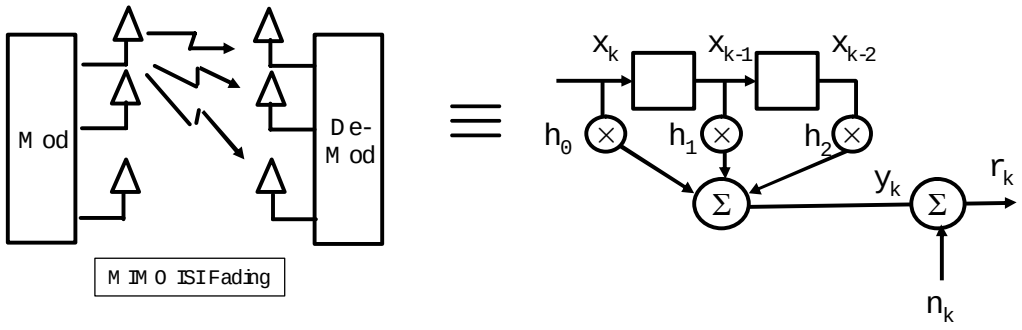

Figure 1: The left figure shows the MIMO RF and antenna system. The right indicates the baseband-equivalent vector-matrix finite impulse response channel model for MIMO system.

where we have defined the following vector variables

$$\boldsymbol{r}_k := \begin{pmatrix} r_k^{(1)} \\ \vdots \\ r_k^{(N_r)} \end{pmatrix}, \boldsymbol{x}_k := \begin{pmatrix} \boldsymbol{x}_k^{(1)} \\ \vdots \\ \boldsymbol{x}_k^{(N_t)} \end{pmatrix}, \mathbf{n}_k := \begin{pmatrix} n_k^{(1)} \\ \vdots \\ n_k^{(N_r)} \end{pmatrix},$$

$$\boldsymbol{h}_{k,l} := \begin{pmatrix} h_{k,l}^{(1,1)} & \cdots & h_{k,l}^{(1,N_t)} \\ \vdots & \ddots & \vdots \\ h_{k,l}^{(N_r,1)} & \cdots & h_{k,l}^{(N_r,N_t)} \end{pmatrix}. \tag{2.2}$$

In the second line in (2.1), $\boldsymbol{y}_k$ is defined to be the clean channel output. In (2.2), $x_k^{(j)}$, $j = 1, 2, ..., N_t$, is the channel symbol at the j^{th} transmitter antenna. The noise $n_k^{(i)}$, $i = 1, 2, ..., N_r$, is a circularly symmetric complex Gaussian (CSCG) variable with zero mean and variance of N_o. In addition, it is independent and identically distributed for each j and k. $h_{k,l}^{(i,j)}$ is the CSCG fading tap with zero mean and a certain variance which can be set

according to multipath power delay profile. The channel $h_{k,l}^{(i,j)}$ is independent and identically distributed for each i and j. Moreover, on the same antenna, the fading coefficients on different fading taps are independent as well. For one particular fading tab, at different time-epoch, the coefficient is time-varying with a maximum Doppler fading rate of f_{dm}. We use the uniform multipath power delay profile and choose the average power of a tap to be $1/L$ for all i and j. Perfect estimation of $\boldsymbol{h}$ is assumed in this model, but it is known only to the receiver part.

We can write the input/output relationship for a block of size N_s by

$$\boldsymbol{r} := \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_{N_s} \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{h}_{1,0} & 0 & 0 & 0 & \cdots \\ \boldsymbol{h}_{2,1} & \boldsymbol{h}_{2,0} & 0 & 0 & \cdots \\ 0 & 0 & \ddots & \ddots & \vdots \\ \boldsymbol{h}_{L,L-1} & \cdots & \boldsymbol{h}_{L,0} & 0 & \cdots \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{h}_{N_s,L-1} & \cdots & \boldsymbol{h}_{N_s,0} \end{pmatrix}}_{:=\boldsymbol{h}} \underbrace{\begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{N_s} \end{pmatrix}}_{:=\boldsymbol{x}} + \underbrace{\begin{pmatrix} \boldsymbol{n}_1 \\ \vdots \\ \boldsymbol{n}_{N_s} \end{pmatrix}}_{:=\boldsymbol{n}}. \tag{2.3}$$

In overall, the block input/output relationship can be written as

$$\boldsymbol{r} = \boldsymbol{h}\boldsymbol{x} + \mathbf{n}, \tag{2.4}$$

where $\boldsymbol{h}$ is the overall channel matrix of size $[N_s N_t \times N_s N_r]$. This description is convenient in describing the algorithm, and finding the theoretical error probability bound for the channels with memory, which will be given in the next section.

3.0 PAIRWISE ERROR PROBABILITY

Under the assumption of perfect channel estimation of $\boldsymbol{h}$, the probability of transmitting codeword $\boldsymbol{x}$ and deciding in favor of another codeword $\tilde{\boldsymbol{x}}$ is approximated using the Chernoff upperbound by

$$P(\boldsymbol{x} \rightarrow \tilde{\boldsymbol{x}} \mid \boldsymbol{h}) \leq \exp(-d^2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) E_s / 4N_0), \tag{3.1}$$

where $d(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is the overall Euclidean distance between $\boldsymbol{y} = \boldsymbol{h}\ \boldsymbol{x}$ and $\tilde{\boldsymbol{y}} = \boldsymbol{h}\ \tilde{\boldsymbol{x}}$. The actual noise is $\boldsymbol{n}$, and $\tilde{\boldsymbol{n}}$ is defined as $\tilde{\boldsymbol{n}} = \tilde{\boldsymbol{r}} - \tilde{\boldsymbol{y}}$. Fig. 2 shows this situation in received signal space.

Fig. 2 shows the Euclidean distance $d_k^{(i)}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = |n_k^{(i)} - \tilde{n}_k^{(i)}|$, on the i-th receiver antenna at a particular time-epoch k. we can express the overall distance $d(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ by accumulating $d_k^{(i)}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ over all k and i, such as

$$\begin{aligned} d^2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) &:= \sum_{k=1}^{N_s} \sum_{i=1}^{N_r} d_k^{(i)}(\boldsymbol{x}, \tilde{\boldsymbol{x}})^2 \\ &= \sum_{k=1}^{N_s} \sum_{i=1}^{N_r} |n_k^{(i)} - \tilde{n}_k^{(i)}|^2, \end{aligned} \tag{3.2}$$

From (2.3), for a particular received signal $r_k^{(i)}$, we can write the channel

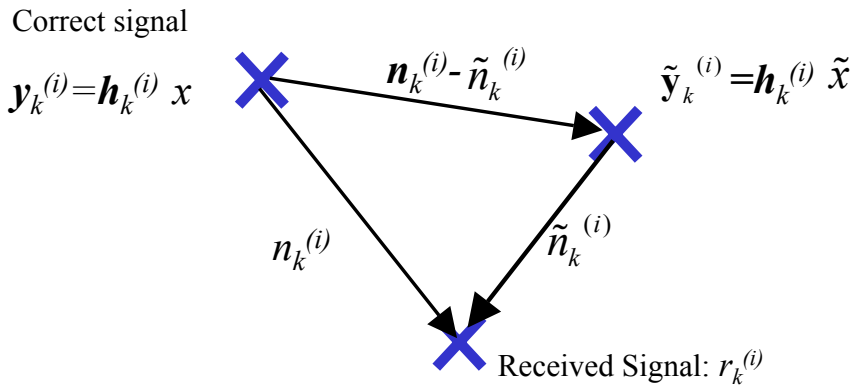

Figure 2: The Different Signal Vectors in the Received Signal Space

input-output relationship as

$$r_k^{(i)} = \underbrace{\left(0 \quad \cdots \quad h_{k,L-1}^{(i,1)} \quad \cdots \quad h_{k,L-1}^{(i,N_t)} \cdots \quad h_{k,0}^{(i,1)} \cdots \quad h_{k,0}^{(i,N_t)} \quad 0 \quad \cdots \quad 0\right)}_{:=\boldsymbol{h}_k^{(i)}} \cdot \boldsymbol{x} + n_k^{(i)}, \tag{3.4}$$

This representation will enable us to analyze the diversity benefit of frequency selective channel in MIMO setting more clearly. Similar equation can be obtained for $\tilde{\boldsymbol{x}}$. Therefore, the distance in (3.2) can be rewritten as:

$$
\begin{aligned}
d^2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) &= \sum_{k=1}^{N_s} \sum_{i=1}^{N_r} |(r_k^{(i)} - \boldsymbol{h}_k^{(i)}\ \boldsymbol{x}) - (r_k^{(i)} - \boldsymbol{h}_k^{(i)}\ \tilde{\boldsymbol{x}})|^2 \\
&= \sum_{k=1}^{N_s} \sum_{i=1}^{N_r} |\boldsymbol{h}_k^{(i)}\ (\boldsymbol{x} - \tilde{\boldsymbol{x}})|^2 \\
&= \sum_{k=1}^{N_s} \sum_{i=1}^{N_r} \boldsymbol{h}_k^{(i)}\ (\boldsymbol{x} - \tilde{\boldsymbol{x}})\ (\boldsymbol{x} - \tilde{\boldsymbol{x}})^*\ (\boldsymbol{h}_k^{(i)})^*.
\end{aligned}
\tag{3.5}
$$

where $(\cdot)^*$ denotes the Hermitian (transpose conjugate) of a matrix or a vector.

As can be seen from (3.4), since only the non-zero part of vector $\boldsymbol{h}_{k,l}$ need to be considered, the part in $\boldsymbol{x}$ needed at the particular k is $(x_{k-L+1}^{(1)} ... x_{k-L+1}^{(N_t)} ... x_k^{(N_t)}) =: \bar{\boldsymbol{x}}$, which is denoted as $\bar{\boldsymbol{x}}$. Thus, (3.5) can be rewritten as:

$$
d^2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sum_{k=1}^{N_s} \sum_{i=1}^{N_r} \bar{\boldsymbol{h}}_k^{(i)}\ (\bar{\boldsymbol{x}} - \bar{\tilde{\boldsymbol{x}}})\ (\bar{\boldsymbol{x}} - \bar{\tilde{\boldsymbol{x}}})^*\ \bar{\boldsymbol{h}}_k^{(i)*} \tag{3.6}
$$

$$
= \sum_{k=1}^{N_s} \sum_{i=1}^{N_r} \bar{\boldsymbol{h}}_k^{(i)}\ \boldsymbol{P}_k^{(i)}\ \bar{\boldsymbol{h}}_k^{(i)*} \tag{3.7}
$$

$$
\tag{3.8}
$$

where $\bar{\boldsymbol{h}}_k^{(i)}$ is the non-zero part of $\boldsymbol{h}_k^{(i)}$. A $[N_t L \times N_t L]$ matrix $\boldsymbol{P}_k^{(i)}$ is defined as $\boldsymbol{P}_k^{(i)} := (\bar{\boldsymbol{x}} - \bar{\tilde{\boldsymbol{x}}}) \cdot (\bar{\boldsymbol{x}} - \bar{\tilde{\boldsymbol{x}}})^*$. Since $\boldsymbol{P}_k^{(i)*} = \boldsymbol{P}_k^{(i)}$, $\boldsymbol{P}_k^{(i)}$ is Hermitian symmetric , there exists a unitary matrix $\boldsymbol{V}_k^{(i)}$ and a real diagonal matrix $\boldsymbol{Q}_k^{(i)}$ such that $\boldsymbol{V}_k^{(i)} \boldsymbol{Q}_k^{(i)} \boldsymbol{V}_k^{(i)*} = \boldsymbol{P}_k^{(i)}$. For the purpose of continuing discussion, it is beneficial to review some properties of $\boldsymbol{Q}_k^{(i)}$ and $\boldsymbol{V}_k^{(i)}$

- $\boldsymbol{V}_k^{(i)}\boldsymbol{V}_k^{(i)*} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. The rows of $\boldsymbol{V}_k^{(i)}$ are a complete basis of an N_tL dimensional space.
- The element of the diagonal matrix $\boldsymbol{Q}_k^{(i)}$ is the eigenvalue of $\boldsymbol{P}_k^{(i)}$, which is denoted as $\lambda_k^{(i)}$ which are all non-negative real. Each $\boldsymbol{P}_k^{(i)}$ has maximum one eigenvalue.
- The matrix $\boldsymbol{P}_k^{(i)}$ and it's eigenvalues are different with respect to different k and i.

Let an $[N_tL \times 1]$ vector $\bar{\boldsymbol{g}}_k^{(i)}$ denote $\bar{\boldsymbol{h}}_k^{(i)}\ \boldsymbol{V}_k^{(i)*}$. Since there are only one non-zero value in the vector, Then we can write:

$$d^2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sum_{k=1}^{N_s}\sum_{i=1}^{N_r} \lambda_k^{(i)} |\bar{g}_k^{(i)(j)}|^2. \tag{3.9}$$

Thus, substituting (3.9) into (3.1), we can obtain

$$P(\boldsymbol{x} \to \tilde{\boldsymbol{x}}|\boldsymbol{h}) \leq \prod_{k=1}^{N_s}\prod_{i=1}^{N_r} \exp\left(-\frac{E_s}{4N_0}\lambda_k^{(i)}|\bar{g}_k^{(i)(j)}|^2\right). \tag{3.10}$$

For the multi-input multi-output ISI fading channel, this equation can be used to approximate the error probability for any given channel matrix $\boldsymbol{h}$. It should be noticed that (3.10) also incorporates the delay diversity taps so that frequency selectivity diversity can be analyzed as well.

Under the assumption of channel matrix, the elements of $\bar{\boldsymbol{h}}_k^{(i)}$ are the circularly symmetric complex Gaussian (CSCG) variables and independent with each other. Since $\bar{\boldsymbol{h}}_k^{(i)}V_k^{(i)*}$ is an unitary transformation of $\bar{\boldsymbol{h}}_k^{(i)}$, the elements of $\bar{\boldsymbol{g}}_k^{(i)}$ are also CSCG random variables and mutually independent. Thus, each $|\bar{g}_k^{(i)(j)}|^2$ is χ^2-distributed random variable with two degrees of

freedom. Given the distribution of $\bar{\boldsymbol{g}}_k^{(i)}$, the ensemble average over $\bar{\boldsymbol{g}}_k^{(i)}$ can be obtained as

$$P_e := P(\boldsymbol{x} \to \tilde{\boldsymbol{x}}) \quad \leq \quad \prod_{k=1}^{N_s} \prod_{i=1}^{N_r} \left[\frac{1}{1 + \frac{E_s}{4N_0} \lambda_k^{(i)}} \right]. \tag{3.11}$$

Remark: We may assume that the eigenvalues of different $\boldsymbol{P}$ matrix are independent with each other. Thus we can write the probability as a product of each individual probabilities.

Comparing (3.11) with the pairwise error probability for MIMO non-ISI fading channels from [8], we can notice that the diversity order is improved by a factor of L when the channel has L delay taps, in addition to the space diversity from N_t transmit antennas and N_r receive antennas. In addition, this result can be applied to either fast fading channel or slow fading channel. However, since fast fading channel provides more diversity benefit than slow fading channel, we present our analysis for fast fading and obtain the system diversity benefit, which then serves as a bound for any practical scheme can achieve. Under the fast fading assumption, the fading coefficients at different time-epochs are assumed to be mutually independent. The probability averaged over N_s and N_r is,

$$P_e \quad \leq \quad \left(\frac{1}{1 + \frac{E_s}{4N_0} \lambda} \right)^{N_s N_r}. \tag{3.12}$$

Recall that the λ_j are the eigenvalues of any matrix $\boldsymbol{P}$, whose elements are the distance between the $\bar{\boldsymbol{x}}$ and $\bar{\tilde{\boldsymbol{x}}}$.

As can be seen from 3.12, the maximum diversity order this system could achieve is dependent upon the product of the number of transmit antenna,

the number of receiver antenna, the number of channel fading taps and the number of time-epochs. When coding is used in the system, any $\boldsymbol{P}$ matrix is formed by the difference matrix between two valid codewords. Therefore, the minimum distance between codewords becomes crucial to the error probability. For given system settings of N_t, N_r, and certain L, a larger distance can lead to an increase in the rank of $\boldsymbol{P}$ matrix, which will increase the diversity benefit of the system and further decrease the error probability. Section IV will continue on the analysis by employing LDPC coding and decoding.

4.0 ROBUST EQUALIZATION–DECODING SYSTEM

The system design on the equalizer and the LDPCC decoder will be discussed in this section.

For the frequency selective channel, an equalizer will be employed to deal with inter-symbol interference. While in the encoding-decoding part, our design is based on Low Density Parity Check Codes in order to achieve near capacity performance. In addition, the turbo iterations between the equalizer and the decoder will exchange the prior information for each signal bit, as shown in Fig. 3.

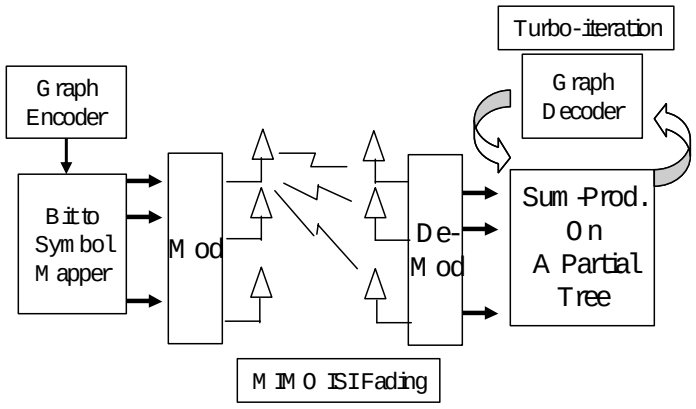

Figure 3: Proposed transceiver for MIMO fading ISI channels

The Gallager's LPDC code is a linear block code defined by a parity-check matrix $\boldsymbol{F}$ where there are n columns, j ones in each column, k ones in each row. The coding rate of the LDPC code R_t satisfies $R_t \leq 1 - j/k$. This parity-check matrix is generated randomly under the weight constraint. Similar to Turbo code, the decoding process of the LDPC code uses iterations to exchange soft information by applying the message passing algorithm [9].

At the transmitter, after the information bits are encoded by LDPC encoder, the modulator will map the coded bit stream into different constellation (2BPSK, 4PSK, 16PSK etc.). Then, the transmitter will execute a series to parallel transformation on the symbol stream and deliver them to N_t different transmitter antennas and send the symbols simultaneously. Assuming no decoding error occurs, a throughput of RMN_t is achieved, where 2^M is the modulation size (for example M = 1 for BPSK). At the receiver, assuming the channel estimation on $\boldsymbol{h}$ is perfect but only known to the receiver, the equalizer takes received signal vector $\boldsymbol{r}_k$ as input. The super iterations (distinct to the inner iteration process inside the LDPC decoder) started at the equalizer. The equalization process begins firstly without prior information and the soft output will be generated then as the extrinsic Log likelihood ratios (LLRs): LLR_{e_k} of each of the LDPC coded bit. The LDPC decoder takes the LLRs as input. It computes as output the a posterior LLRs (LLR_{d_k}) of the coded bits and send them to the equalizer to extract the prior information (LLR_{p_k}).

$$LLR_{p_k} = LLR_{d_k} - LLR_{e_k} \qquad (4.1)$$
$$k = 1...n$$

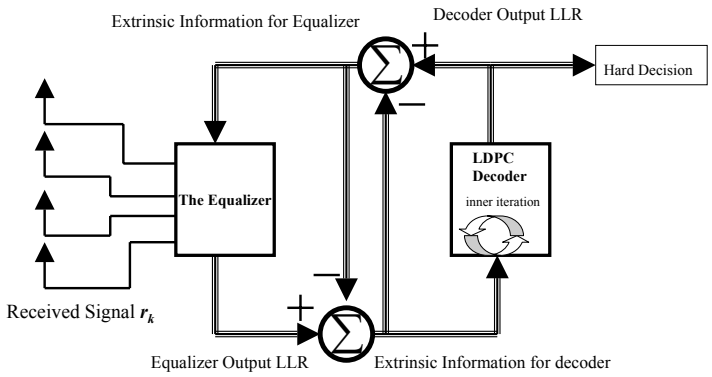

Figure 4: The receiver structure, which employs an equalizer and a soft LDPC decoder, for multiple-antenna system under ISI fading channel.

Similarly, in the second super iteration, updated LLR_{e_k} is delivered to the LDPC decoder as the new prior information for the LDPC decoder. After certain number of iterations, the hard decisions will be made at the output of the decoder. Fig. 4 shows the receiver structure. The detailed design of LDPC encoder and decoder can be found in [9], [11], and [12]. The following sections will focus on the proposed equalizer.

5.0 REDUCED COMPLEXITY EQUALIZER

In this section, we will introduce the novel reduced complexity equalizer design. First of all, the soft-input soft-output (SISO) equalization is based on a pruned tree. The pruning decision is made based on a threshold test on the product of the likelihood and the priors for coded system. Priors will become available when a soft-input soft-output decoder is used at the receiver in a turbo-iterative manner with the proposed SISO equalizer. We use the example of binary modulation and a single transmit and receive antenna for simplicity.

5.1 THE SUM-PRODUCT SOFT-INPUT/SOFT-OUTPUT EQUALIZER

From (2.4), the posteriori probability can be written as

$$\begin{aligned} Pr\{\boldsymbol{x}|\boldsymbol{r}\} &= P(\boldsymbol{x},\boldsymbol{r})P(\boldsymbol{r}) && (5.1)\\ &\propto P(\boldsymbol{r}|\boldsymbol{x})Pr(\boldsymbol{x}) && (5.2)\\ &\propto \prod_{k=1}^{N_s} P(\boldsymbol{r}_k|\boldsymbol{x}_k,\cdots,\boldsymbol{x}_{k-L+1})Pr(\boldsymbol{x}_k) && (5.3) \end{aligned}$$

where (5.1) is due to conditional probability, the second one the Bayes' Theorem, and the third line is due to the assumption that the noise is additive Gaussian white noise. This is the product part of the algorithm.

The likelihoods at each time-epoch k can be calculated with the following way:

$$P(\boldsymbol{r}_k|\boldsymbol{x}_k, \cdots, \boldsymbol{x}_{k-L+1}) \propto \exp(-\frac{1}{N_o}||\boldsymbol{r}_k - \boldsymbol{y}_k||^2) \tag{5.4}$$

$$=: L(\boldsymbol{y}_k), \tag{5.5}$$

where in (5.5) we define the likelihood metric $L(\boldsymbol{y}_k)$ to be the quantity on the left-hand side of (5.4).

The priors $Pr(\boldsymbol{x}_k)$ are initially set to equally likely for all possible $\boldsymbol{x}_k$. After the first super-iteration with the graph decoder, the priors on channel-symbol can be obtained from the extrinsic part of the posteriors which are the outputs of the graph decoder. Thus, the posteriors on the sequences $Pr\{\boldsymbol{x}|\boldsymbol{r}\}$ can be calculated using the likelihoods and priors at each k. Then, the posteriors on a particular time-epoch, the individual posterior on the input-symbol vector $\{\boldsymbol{x}_k\}$, can be obtained by using the summation part of this algorithm which will be described next with an example.

It might be more practical to use the log version of the algorithm which can be obtained from taking the log on both sides of (5.3). Then, the equation of the proportionality becomes

$$\log(Pr\{\boldsymbol{x}|\boldsymbol{r}\}) \propto \sum_{k=1}^{N_s}[\log(L(\boldsymbol{y}_k)) + \log(Pr(\boldsymbol{x}_k))]. \tag{5.6}$$

We note that the log likelihood part is simply the Euclidean distance between the received sequence $\boldsymbol{r}$ and the clean channel output $\boldsymbol{y}$ for a particular candidate $\boldsymbol{x}$. That is, the Euclidean distance $d(\boldsymbol{y}, \boldsymbol{r})$ of a particular

candidate is thus written as

$$d(\boldsymbol{r}, \boldsymbol{y}) := \sum_{k=1}^{N_s} d(\boldsymbol{r}_k, \boldsymbol{y}_k). \tag{5.7}$$

Thus, one is now able to calculate the posterior for a particular clean channel output $\boldsymbol{y}$ which is dependent upon the transmitted input sequence $\boldsymbol{x}$.

5.2 SUMMATION PART

We now use an example depicted in Fig. 5 for the illustration of the summation part of the sum-product algorithm. Note that in this example, we use a binary tree for simplicity. Assume $N_s = 3$ and the full-tree is obtained. There are eight possible sequences, indexed from $q = 0, 1, 2, \ldots, 7$. For full-complexity tree search, the posteriors $P(\boldsymbol{x}|\boldsymbol{r})$ for all eight candidates are calculated. Then the posterior on a particular input event at the epoch k can be obtained by summing all those posteriors having that particular input symbol at k. That is, using our example,

$$Pr\{x_2 = +1|\boldsymbol{r}\} = \sum_{q=0,1,4,5} Pr\{\boldsymbol{x}(q)|\boldsymbol{r}\}, \tag{5.8}$$

$$Pr\{x_2 = -1|\boldsymbol{r}\} = \sum_{q=2,3,6,7} Pr\{\boldsymbol{x}(q)|\boldsymbol{r}\}. \tag{5.9}$$

BCJR algorithm [3] effectively performs the same operation, but on a trellis–instead on a tree. For a full-complexity search, BCJR algorithm is more efficient than the use of the algorithm on a tree, without losing any optimality. For reduced complexity scheme on MIMO settings, however, the use of trellis-approach incurs insurmountable amount of computational

complexity simply due to large numbers of trellis states and edges. For example, with a binary modulation $M = 4$, $N_t = 4$ and $L = 3$, the number of trellis state is $4^8 = 65536$ which is clearly un-manageable.

5.3 PRUNING BY THRESHOLD TEST AND APPLYING THE SUM-PRODUCT ON SURVIVED PATHS

Again referring to Fig. 5, one may notice that the summation algorithm can be applied also to the pruned tree. However, there is one problem dealing with the comparison of the paths having different lengths: Some are pruned earlier than others; then how do we fairly compare them. An earlier pruned path unfairly has an advantage of smaller Euclidean distance simply because it is short in length.

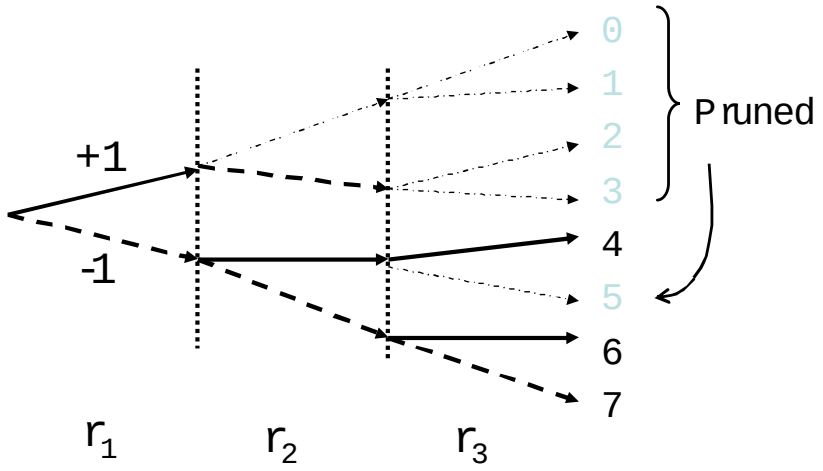

Figure 5: Example of a pruned tree

One solution is to truncating all the sequences in comparison at a same

length. This can be easily achieved by making the threshold based pruning decision while accumulating the Euclidean distances of all the survived paths, starting from the beginning of the tree. This option has a disadvantage since it is not making use of the information available in longer-lived survivors. The solution can be applying a compensation rule for the difference in path lengths, which will be described in the next sub-section.

We now describe the simpler solution which is applying the threshold-based pruning-rule while expanding the tree in a forward direction. Referring again to Fig. 5, we note at $k = 2$, the first path were pruned. From the $k = 1$ epoch to $k = 2$, a survivor path gets expanded into M^{N_t} candidates. Thus, the total candidate in our smaller example is 4, 2 survivors at $k = 1$ and 2 branches out of each survivor. First, we calculate the accumulated metric up to the current exploration depth $k = k_o$ ($k_o = 2$ in this example) for each of the candidates using (5.3)–$\prod_{k=1}^{k_o} L(\boldsymbol{y}_k)Pr(\boldsymbol{x}_k)$–or use the log version of it, (5.6). Then, the rejection rule is as follows:

1. Find the best metric path and set its associated metric value–say B_k.
2. Prune all candidates whose accumulated metric is larger than B_k/T where $0 < T < 1.0$ is a predefined constant

Simmons [4] investigated similar rules based on trellis structure. We extend the idea with the tree structure for more efficient application towards the MIMO fading ISI channels.

5.4 COMPENSATION PART

As part of the proposed transceiver system, the super-iterations will be carried out between the equalizer and decoder by exchanging the extrinsic

information. Therefore, instead of decide only one path in the equalizer tree structure, the soft-output is needed for the Turbo-like super iteration to minimize the errors.

For each early pruned path, a certain amount of Euclidean distance should be added, to complete the calculation of the overall cumulative metric needed in the sum-product algorithm, to the calculation of the overall distance of the path. This is to compensate for the part that are not explored, and its corresponding distance which would have been exactly evaluated only if the path had survived to the end of the tree. In this problem, we seek a simpler solution such that we choose a single compensation C_k per each missing edge. If a path is pruned at $k = j$, for example, then a metric $\sum_{k=j}^{N_s} C_k$ is added to all those paths in which missing edge is starting from $k = j$. This value C_k is the *sufficient* minimum metric such that all pruned paths at the k-th tree section must satisfy this requirement, even though a pruned path may have smaller edge metric than C_k. However, the compensation C_k is varied for different tree-sections. Without giving a proof (due to the limited space provided), we state the following lemmas:

Lemma 1 *Let β_{k-1} and β_k be the minimum forward-cumulative distances at the $k-1$-th and k-th tree sections respectively, and let T' be the corresponding Euclidean distance for the threshold ratio T. Then, a compensation rule satisfying the following inequality*

$$C_k \geq \beta_k - \beta_{k-1} + T' \tag{5.10}$$

at each tree-section is sufficient, such that if any path whose edge metric at k is greater than or equal to the right hand side it always gets rejected by the threshold test.

Lemma 1 readily leads to the following compensation rule, which is summarized in the following Theorem.

Theorem 1 *For a path pruned at $k = j$ where $1 \leq j \leq N_s$, we consider the following compensation rule $\sum_{k=j}^{N_s} C_k$:*

$$\sum_{k=j}^{N_s} C_k \geq \beta_{N_s} - \beta_j + (N_s - j) \times T'. \tag{5.11}$$

This rule is sufficient.

By applying the sum-product tree search algorithm and the compensation rules, the MIMO system is able to handle the case of high modulation size and large number of transmit antennas. Meantime, when we consider the error performance of this system, although it is a sub-optimal approach compared to full trellis search, the bit error rate (BER) is very close to the theoretical bound for uncoded information source, which will be shown in the simulation result section. However, as the number of antennas or the constellation size further increases, even with this design, the complexity still could be forbiddingly high. Again using Fig. 5 as an example, we could see that each survivor path will be expended into M^{N_t} candidates. This number is 2 for the system shown in Fig. 5, but if we have $M = 16$ and $N_t = 4$, the number will be 65536. In T-algorithm, the cumulative metric for each of these candidates need to be computed before the threshold test, which is an unreachable amount of computing for real time communication system. In order to solve this problem, we propose the sphere list detection algorithm to generate a shorter candidate list before the threshold test, and further reduce the computational complexity.

6.0 SPHERE LIST DETECTION

Motivation for sphere list detection (SLD) is to allow a large constellation and a large number of antennas so that the system could enable higher spectral efficiency. The detailed discussion about SLD will be given in this section.

Hochwald and Brink [5] first used the sphere decoding algorithm and shows very promising for the MIMO flat fading channels. For the channel with intersymbol-interference, we develop the list sphere detection rules and combine it with the threshold testing process.

6.1 SLD IN ISI CHANNEL

In the tree-pruning operation, a survived node is expanded and the expanded path becomes the candidates intended for the threshold test. The SLD is applied at the expansion phase of this routine such that each node in the tree is expanded only when they are within the sphere of a pre-chosen radius from the channel output on that branch. Thus, the number of per-node expanded candidates is much smaller than the full list of size M^{N_t}. Note that our channel has L-taps and thus a received signal vector at a particular branch has contribution from up to $L-1$ previous channel symbols. The

previous symbols are stored in the memory of the node. Using them, we can cancel out the contribution of the previous symbols at each node. It is worthwhile to note that this cancellation is exact on the correct path and in-exact on an incorrect path. The consequence of this cancellation is an increased number of candidates in the list on the correct path. On all incorrect paths, however, it leads to an increase in the effective noise level and thus the number of candidates will be dramatically reduced. We note that this is one of the properties very much desired in our reduced complexity scheme.

The result of this cancellation is denoted by $\boldsymbol{r}'_k$:

$$\begin{aligned} \boldsymbol{r}'_k &= \boldsymbol{r}_k - \textstyle\sum_{l=1}^{L-1} h_{k,l}\bar{\boldsymbol{x}}_{k-l} && (6.1) \\ &= \boldsymbol{h}_{k,0}\boldsymbol{x}_k + \boldsymbol{D}_k + \boldsymbol{n}_k. && (6.2) \end{aligned}$$

where $\bar{\boldsymbol{x}}_{k-l}$ is the previous signal on the path, and D_k is the cancellation error. $D_k = 0$ on the correct path.

6.2 SPHERE LIST DETECTION IN THE STAGE OF EXPANDING EACH SURVIVOR PATH

The goal of the sphere list detection is to generate a list of most posterior-probable candidates given the received signal. After the cancellation process, the probability of a particular signal vector $\boldsymbol{x}_k$ being the correct signal is determined by the Euclidean distance, since the noise is Gaussian, i.e.,

$$||\boldsymbol{r}'_k - \boldsymbol{h}_{k,0}\boldsymbol{x}_k||^2 \qquad (6.3)$$

where $\boldsymbol{r}'_k$ is the received signal due only to the current input vector $\boldsymbol{x}_k$. Thus, we use the distance measure to find the list: only those signal vectors with an Euclidean distance less than a certain radius will be kept in the candidate list. However, an exhaustive search can again grow to an unmanageable size, which is not desired. Using the SLD, the list can be obtained in an efficient manner, thanks to Fincke-Pohst algorithm [6]. The process of sphere list detection, as flow-charted in Fig. 6, starts with the unconstrained estimation of $\boldsymbol{x}_k$. Hochwald and Brink [5] used the maximum likelihood (ML) estimator

$$\begin{aligned} \hat{\boldsymbol{x}}_{k,ML} : &= \arg\max \mathrm{P}(\boldsymbol{r}'_k|\boldsymbol{x}_k) \\ &= (\boldsymbol{h}^*_{k,0}\boldsymbol{h}_{k,0})^{-1}\boldsymbol{h}^*_{k,0}\boldsymbol{r}'_k. \end{aligned} \tag{6.4}$$

The estimator requires the $\boldsymbol{h}_l$ matrix to be full rank. In case it is not a full rank matrix, the estimator may encounter a large estimation error due to matrix-inversion operation. This motivates us to devise a regularized estimator. We propose the Minimum Mean Square Estimator (MMSE):

$$\hat{\boldsymbol{x}}_k := \arg\min E\{||\boldsymbol{x}_k - \hat{\boldsymbol{x}_k}||^2\}. \tag{6.5}$$

This can be simplified to be:

$$\hat{\boldsymbol{x}}_k = (\boldsymbol{h}^*_{k,0}\boldsymbol{h}_{k,0} + \boldsymbol{R}_n)^{-1}\boldsymbol{h}^*_{k,0}\boldsymbol{r}'_k, \tag{6.6}$$

where $\boldsymbol{R}_n = N_0\boldsymbol{I}_{N_t\times N_t}$ is the noise covariance matrix. With the MMSE estimator $\hat{\boldsymbol{x}}_k$, the Euclidean distance detection criteria can be written as:

$$\begin{aligned} ||\boldsymbol{r}'_k - \boldsymbol{h}_{k,0}\boldsymbol{x}_k|| = (\boldsymbol{x}_k - \hat{\boldsymbol{x}_k})^*(\boldsymbol{h}^*_{k,0}\boldsymbol{h}_{k,0} + \boldsymbol{R}_n)(\boldsymbol{x}_k - \hat{\boldsymbol{x}_k}) \\ +\boldsymbol{r}'^*_k(I - \boldsymbol{h}_{k,0}(\boldsymbol{h}^*_{k,0}\boldsymbol{h}_{k,0} + \boldsymbol{R}_n)^{-1}\boldsymbol{h}^*_{k,0})\boldsymbol{r}'_k. \end{aligned} \tag{6.7}$$

The second term in (6.7) is a constant for different $\boldsymbol{x}_k$. Thus, the sphere list detection criteria is to find every candidate $\boldsymbol{x}_k$ which satisfied:

$$(\boldsymbol{x}_k - \hat{\boldsymbol{x}_k})^*(\boldsymbol{h}_{k,0}^*\boldsymbol{h}_{k,0} + \boldsymbol{R}_n)(\boldsymbol{x}_k - \hat{\boldsymbol{x}_k}) \leq R^2 \tag{6.8}$$

where R is the sphere radius. In this criteria, we applied the Fincke-Pohst algorithm [6]. As shown in Fig. 6, we use the Cholesky factorization on $(\boldsymbol{h}_{k,0}^*\boldsymbol{h}_{k,0} + \boldsymbol{R}_n)$: $\boldsymbol{h}_{k,0}^*\boldsymbol{h}_{k,0} + \boldsymbol{R}_n = \boldsymbol{U}^*\boldsymbol{U}$, where $\boldsymbol{U}$ is an upper triangular $[N_t \times N_t]$ matrix. Since the $(\boldsymbol{h}_{k,0}^*\boldsymbol{h}_{k,0} + \boldsymbol{R}_n)$ is a positive definite matrix (the sum of two positive definite matrix is still positive definite), the upper triangular matrix $\boldsymbol{U}$ always exists with all diagonal elements being positive real numbers. Therefore, (6.8) can be written as:

$$\begin{aligned}
&(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^*\boldsymbol{U}^*\boldsymbol{U}(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k) \\
&= \sum_{i=i}^{N_t} U_{ii}^2[x_k^i - \hat{x}_k^i + \sum_{j=i+1}^{N_t} \frac{U_{ij}}{U_{ii}}(x_k^j - \hat{x}_k^j)]^2 \\
&\leq R^2
\end{aligned} \tag{6.9}$$

It should be noticed that each term in the sum over i, $i = 1, 2, ...N_t$ is nonnegative. Starting the summation from $i = N_t$ without loss of generality, we also note that as soon as part of the summation exceeds R^2, say at $i = t$, the total sum is definitely is bigger than R^2. Therefore, there is no need to proceed further for all dimensions for $i < t$. Considering at $i = N_t$, for instance, we can choose all candidates for $x_k^{N_t}$ which satisfies:

$$U_{N_tN_t}^2(x_k^{N_t} - \hat{x}_k^{N_t})^2 \leq R^2. \tag{6.10}$$

For each candidate $x_k^{N_t}$, we continue to choose a candidate of $x_k^{N_t-1}$ by again using 6.9. Now we decide on the last two elements of vectors. Similarly,

this process can continue on until $i = 1$. It is possible that no candidate value could be found upon i^{th} dimension. In such cases, the process will continue by going back to the choice of x_k^{i+1}, until some x_k^i was found. The Fig. 6 shows the detailed flow-chart of sphere list detection process.

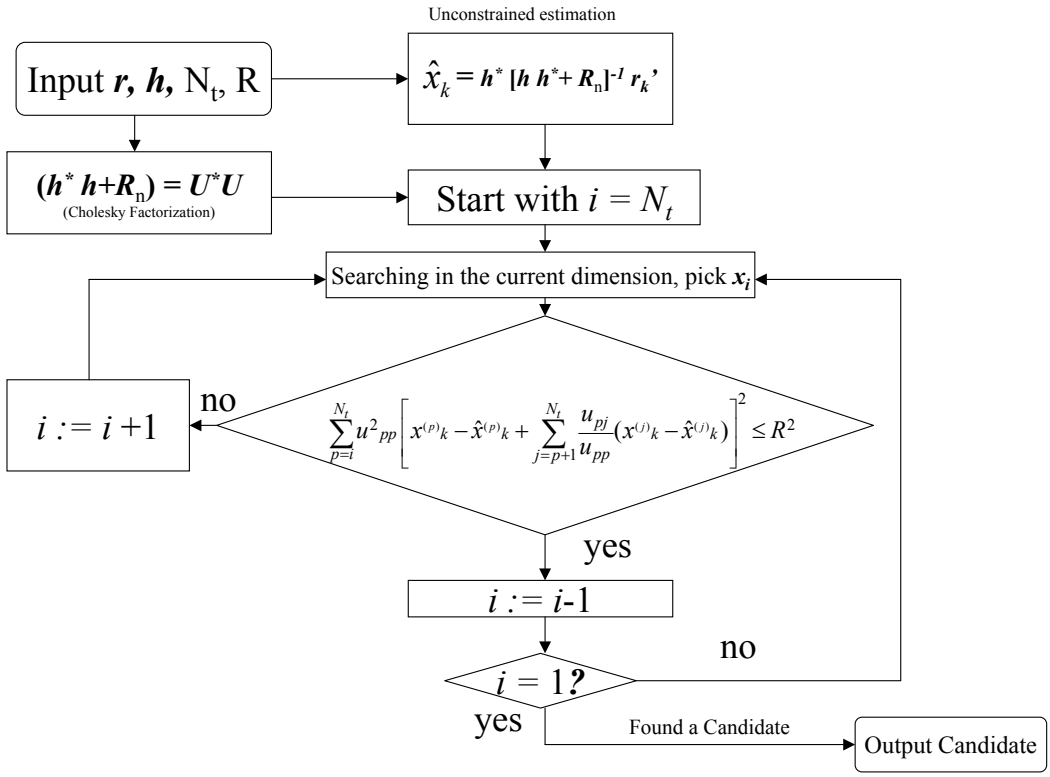

Figure 6: The Procedure of Sphere List Detection.

Upon completing the SLD process, a list of candidates is obtained and extends a survivor to the next time-epoch. Gathering all the candidates from all survivors, we can do the threshold testing on each of them. With compensation, soft output for each bit of coded information will be generated and delivered to the decoder as the prior information.

7.0 SIMULATION RESULTS

In this section, we discuss extensively the computer simulation results.

We use SNR per-information-bit when drawing the curve compared with the theoretical bounds. If SNR_r is defined as Signal-to-Noise Ratio per received channel, then SNR/bit will be:

$$SNR = SNR_r \cdot N_r/[R_t \cdot log_2(M) \cdot N_t]. \tag{7.1}$$

Where R_t is the coding rate, i.e., $R_t = 1/2$ for rate-half LDPC code. In addition, Gray mapping is applied in our simulation on the system.

7.1 UNCODED RESULTS: ($N_T = 4, N_R = 4$)MIMO SYSTEM FOR BPSK/4-QAM

Fig. 7 shows the performance of the equalizer system with BPSK modulation. Without utilizing the sphere list detection, the T-algorithm is applied to this system with uncoded transmission. The BER curve is drawn compared with the Matched Filter Bound (MFB) over a ($N_t = 4, N_r = 4, L = 3$) system [17] with same order of diversity gained. By keeping an average of

less than 10 survivors–a large reduction compared with 256 states in full trellis search– the BER performance is shown to be within 2 dB from the ideal MFB.

The performance under 4-QAM system is shown in Fig. 8. In this case, full complexity trellis has 65536 transitions on each of which a posterior probability is needed to be calculated each time-epoch. By combining the sphere list detection and the T-algorithm, we keep the average length of candidate lists under 20 at the output of SLD, and the average number of survivors around 15, which is comparable to the complexity of BPSK case. Thus, the spectral efficiency is doubled while keeping the computational complexity at the same level. The BER performance is shown to be about 2 dB away from the matched filter bound. In both case, the channel is slow faded with a normalized Doppler fading rate , $f_{dm} \cdot T$, of 1/256, where T is the symbol period.

7.2 CODED RESULTS: ($N_T = 4$,$N_R = 4$) MIMO SYSTEM WITH LDPC CODE

When the equalizer and decoder is combined, the decoder can either take the hard output of equalizer to do the decoding, or utilize the soft information so that the super-iteration could be applied between the two parts. We assume the channel is fast fading channel with a normalized Doppler fading rate , $f_{dm} \cdot T$, of 1/16.

As discussed in Chapter IV, (**??**) shows the pairwise error probability of LDPC coded system. For our setting with a (4096, 4,8) LDPC code, calculating the exact error probability for different SNR is not straightforward

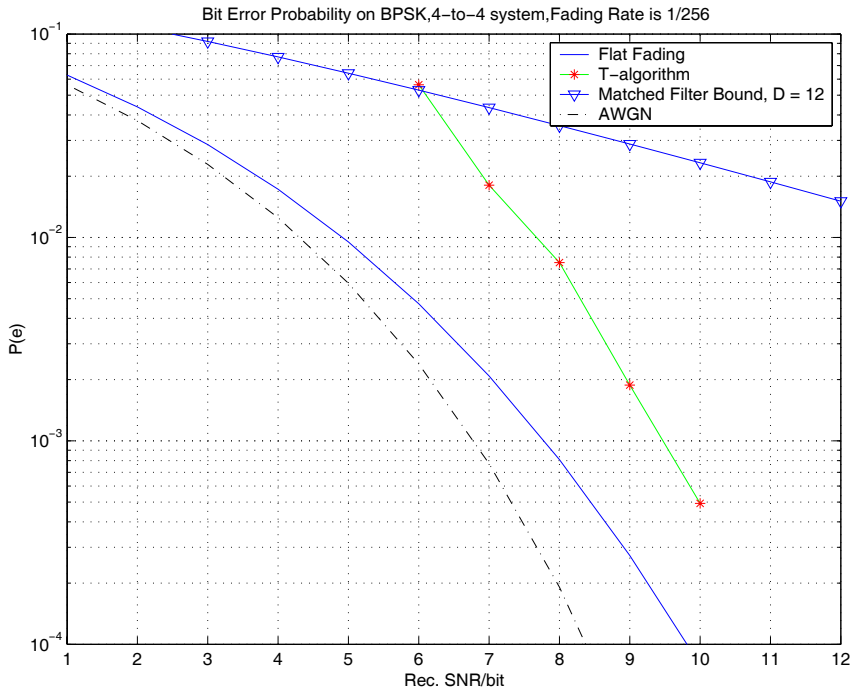

Figure 7: 4×4 system BPSK modulation, without SLD

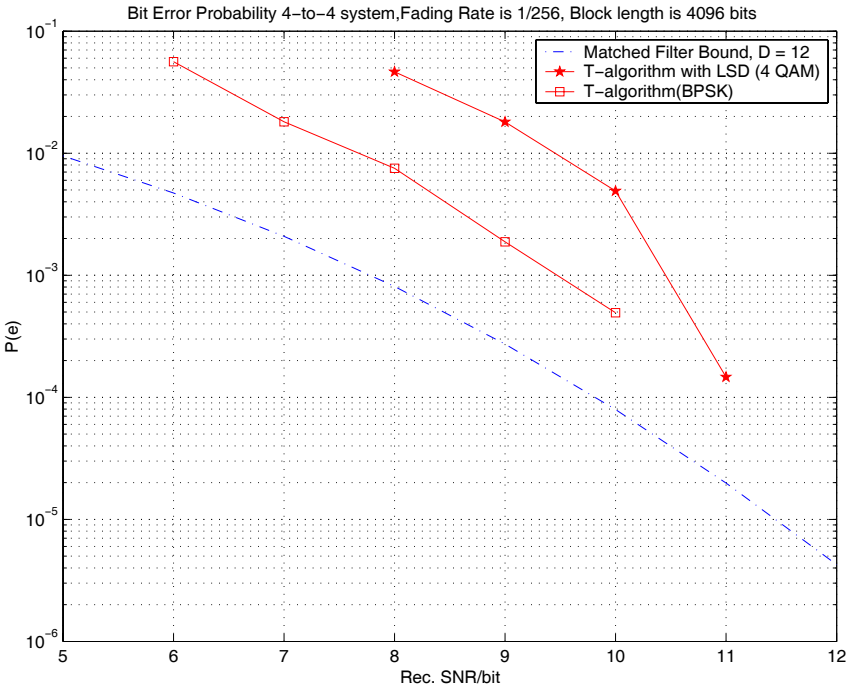

Figure 8: 4×4 system 4-QAM modulation, with SLD

for channels with ISI. However, we could generate the probability by doing ensemble average over a randomly selected code with certain minimum distance. As shown in [9], the parameter σ is a constant for an LDPC code with particular column weight and row weight. For an (4096,4,8) LDPC code, the minimum distance between two codewords, D_{min}, is 257 [9], where $D_{min} = \sigma\ N_s$. Thus, we can obtain the mean bit error probability by doing simulation with the following steps:

Step 1 By doing Bernouli trials with parameter $p = \sigma$, we generate 1000 sequences, and each of them has a length of N_s and a mean weight of D_{min}.

Step 2 For each sequence, we generate $[N_s \times N_r]$ $P_k^{(i)}$ matrices and obtain the corresponding eigenvalues which we use to evaluate (3.12) to get P_e.

Step 3 Perform Step 2 1000 times with different sequences, and find the average of P_e.

Step 4 Calculate bit error probability P_b by

$$P_b = P_e \frac{D_{min}}{N_s} \binom{N_s}{D_{min}} \tag{7.2}$$

as the approximation for bit error probability.

This theoretical error probability can be used to compare with the system error probability, as can be seen in Fig. 10 and Fig. 11.

Fig. 9 shows the overall system performance when the equalizer generates the hard output vectors without applying the compensation rules. The LDPC decoder could reach the error free zone at about 8.5 dB with only one super-iteration. However, the error correcting capacity of LDPC decoder is constrained by the hard output of the equalizer, which further weaken the

efficiency of super-iteration. As shown in following results, employing soft output does help to improve the error correcting performance.

Another observation is that when coding is employed, middle SNR (6-9dB) is most crucial in BER, where the equation-decoding algorithm works less ideal than higher SNR region. This factor leads to a larger performance difference with the theoretical bound compared with the uncoded simulation.

When compensation rule is applied to generate the soft-output, the BER performance is further improved by 1.5 dB, as shown in Fig. 10. Compared with the pairwise error probability bound shown in this figure, the decoding output, which reflects the overall system performance without super-iterations, is about 1 dB away from the theoretical error probability bound obtained by pair-wise error analysis. In addition, with the help of LDPCC decoder output as soft-input to equalizer, the equalizer output error rate is greatly decreased. So does the decoder output. Fig. 11 shows the effect of super-iterations. After 3 super-iterations, the error performance is within 0.8 dB away with the error probability approximation, as shown in the figure. Compared to the full complexity exhaustive search, our approach could reach closely to the theoretical bound, while keeping only averagely 15 survivors each time-epoch. Compared to 65536 transition calculations in full trellis search under similar system settings, we saved more than 99% of computational complexity.

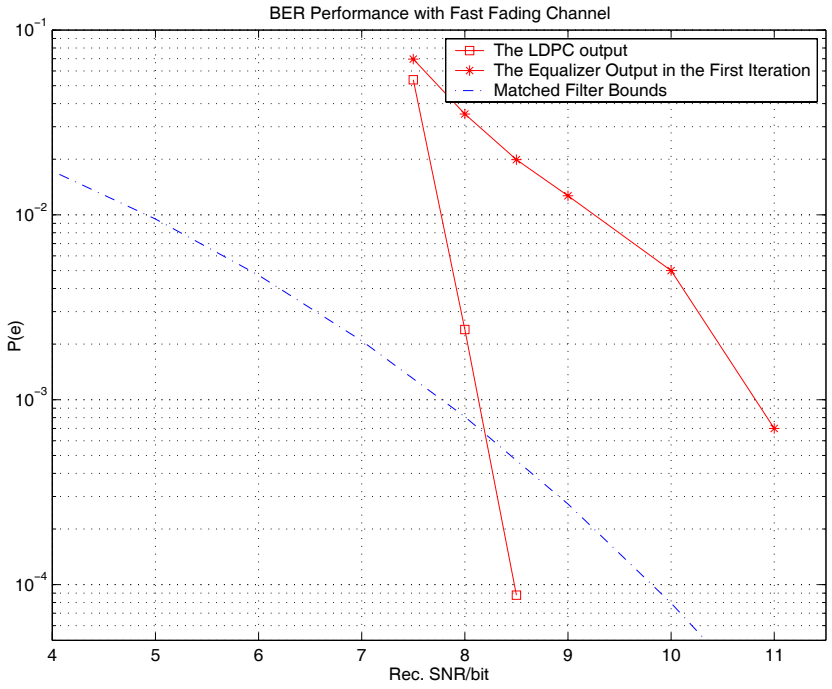

Figure 9: Overall System Performance with Hard Equalizer Output

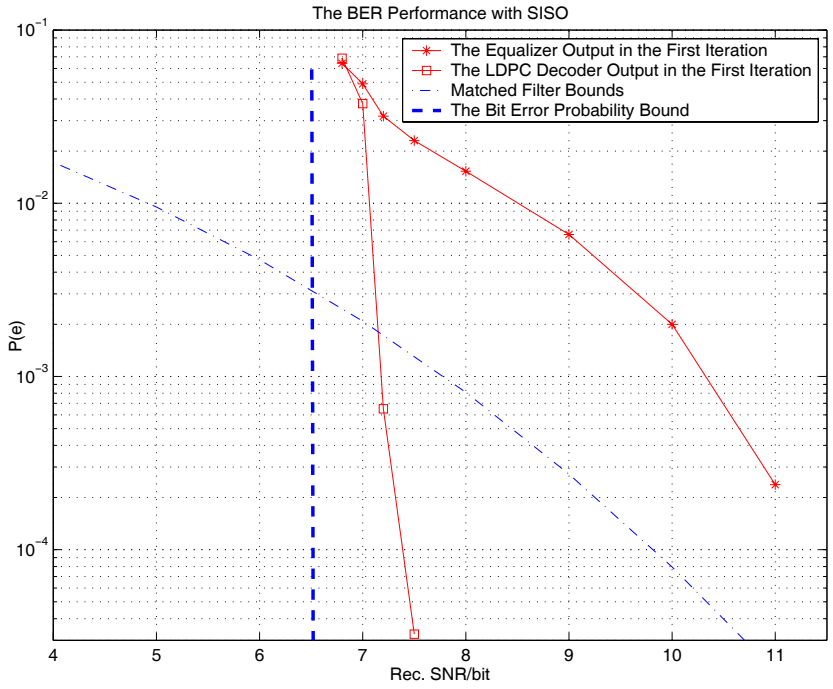

Figure 10: Overall System Performance with Soft Equalizer Output

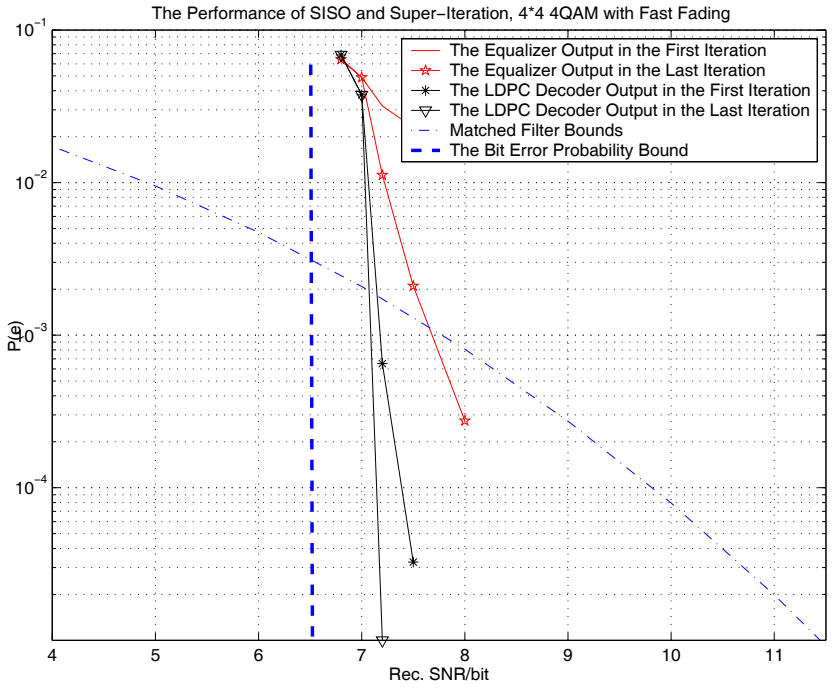

Figure 11: Overall System Performance with 3 Super-Iterations

8.0 SYSTEM COMPUTATIONAL COMPLEXITY ESTIMATION

In this section, the relationship between the computational complexity and system parameters is discussed. The analysis is focused on the sphere list detection, which consumes the dominating parts of computations. In addition, the effects upon complexity as the number of transmit/receive antennas, or modulation size is increased is investigated. The theoretical computational complexity analysis can help estimate the data processing speed and applicability for the proposed system in a particular system setting. For an example, at a particular processing speed, system parameters such as M N_t N_r can be selected in order to reach processing time limit.

In the sphere list detection part, for each survivor, the SLD rules is applied and a list of candidates is generated assuming the path is correct. The number of transmit/receive antennas (N_t/N_r) or modulation size affects the number of computations required. Besides, the predefined search radius R is also a key factor overall. Therefore, the general problem will be:

- Choice of radius R
- The number of operations used for a particular system,

The paper[14] by Vikalo and Hassibi discussed the complexity issue on the MIMO flat fading channel case. We adopt their mathematical framework and apply t to the MIMO frequency-selective fading channel.

Given N_t transmit antennas, without loss of generality, we choose to start the search for qualified candidates from the N_t-th antenna to the first one. Although most points visited will not appear on the final candidates list, each point visited will lead to a certain amount of operations. The overall structure of the algorithm can be represented as a tree, which is shown in Fig. 12.

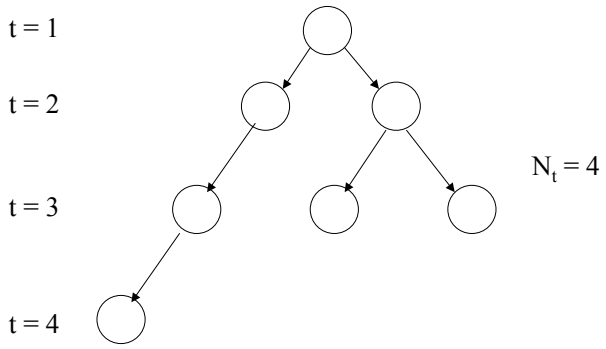

Figure 12: Sample of computation process structure

At a particular level of the tree, the amount of operations used for each point visited is the same; while they are different at different levels of the tree. Therefore, the overall number of computations can be obtained by summing up all those individual sum of computations at different levels. The number of computation per point in the tree can be calculated by

$$f_p(t) = 8t + 17 \tag{8.1}$$

Then, for a system with N_t transmit antennas, N_r receive antennas, and noise variance of N_0, we can write the overall number of computations, denoted as V, as (8.2)

$$V(N_t, N_r, N_0) = \sum_{t=1}^{N_t} (N_p^{(t)})\ f_p(t). \tag{8.2}$$

where $N_p^{(t)}$ is the expected number of points visited in t-th level.

For a particular point $\tilde{\boldsymbol{x}}_k$, the probability that it is visited during the sphere list detection process depends on the Euclidean distance between $\tilde{\boldsymbol{x}}_k$ and the correct signal vector $\boldsymbol{x}_k$. Supposing the point $\tilde{\boldsymbol{x}}_k$ was in the final list, the following equation must hold:

$$||\boldsymbol{r}'_k - \boldsymbol{h}_0^{(k)} \times \tilde{\boldsymbol{x}}_k|| \leq R^2 \tag{8.3}$$

where $\boldsymbol{r}'_k$ is the received signal vector after correct-path cancellation, as described in section VI. A. Since $\boldsymbol{r}'_k = \boldsymbol{h}_{k,0}\boldsymbol{x}_k + \boldsymbol{D}_k + \boldsymbol{n}_k$, the cancellation error $\boldsymbol{D}_k$,i.e., $\boldsymbol{D}_k = 0$ on correct paths can be expressed as

$$\boldsymbol{D}_k = \sum_{l=1}^{L-1} h_{k,l}(\boldsymbol{x}_{k-l} - \bar{\boldsymbol{x}}_{k-l}), \tag{8.4}$$

where $\bar{\boldsymbol{x}}_{k-l}$ is the previous signal on the path. Recall that each element of $\boldsymbol{h}_{k,l}$ is CSCG variable, thus each element of $\boldsymbol{D}_k$ can be viewed as a CSCG variable with zero mean as well. Thus, (8.3) can be expressed as

$$||\boldsymbol{n}_k + \boldsymbol{D}_k + h_0(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)|| \leq R^2. \tag{8.5}$$

Considering the vector of $\boldsymbol{\varepsilon}_k := \boldsymbol{n}_k + \boldsymbol{D}_k$, each element of $\boldsymbol{\varepsilon}_k$ is CSCG variables with zero mean and thus can be viewed as noise with larger variance, denoted as N_e. In medium to high SNR region, which most of our attention

lies on, erroneous path tends to have the minimum difference with the correct path. Therefore, within the list of survivors leading different paths at any time-epoch, we may assume that there is the correct path and all the other incorrect paths are due to minimum distance error events for an estimation in complexity analysis. Now, we can estimate the overall computations $V(N_t, N_r, N_0)$ as

$$\begin{aligned} & V(N_t, N_r, \sigma^2) \\ & = \sum_{t=1}^{N_t} f_p(t) \sum_{l=0}^{\infty} \gamma(\frac{R^2}{2(N_e + l)}, \frac{N_r - N_t + t}{2})\, g_t^{(l)}, \end{aligned}$$

where $\gamma(\cdot, \cdot)$ is the incomplete Gamma function in the χ^2 distribution, which represents the probability of a particular point having a distance less than R. $g_t^{(l)}$ denotes the number of constellation points in the t^{th} hyper-space with a distance of l, in which the neighboring points have the unit distance. According to Eular's idea addressed in [13], for 16QAM system for an example, $g_t^{(l)}$ can be calculated by

$$g_t^{(l)} = \sum_{\lambda} \frac{1}{2^t} \binom{t}{l} M_{t,N_t}^{\lambda}$$

where M_{t,N_t}^{λ} is the coefficient of x^{λ} in this polynomial

$$(1 + x + x^4 + x^9)^m (1 + 2x + x^4)^{k-m} \tag{8.6}$$

Following this result, we can evaluate the amount of computation used for systems with different M, N_t,N_r and different search radius R. Fig. 13 and Fig. 14 show the results for 4-QAM and 16QAM constellations respectively with different R.

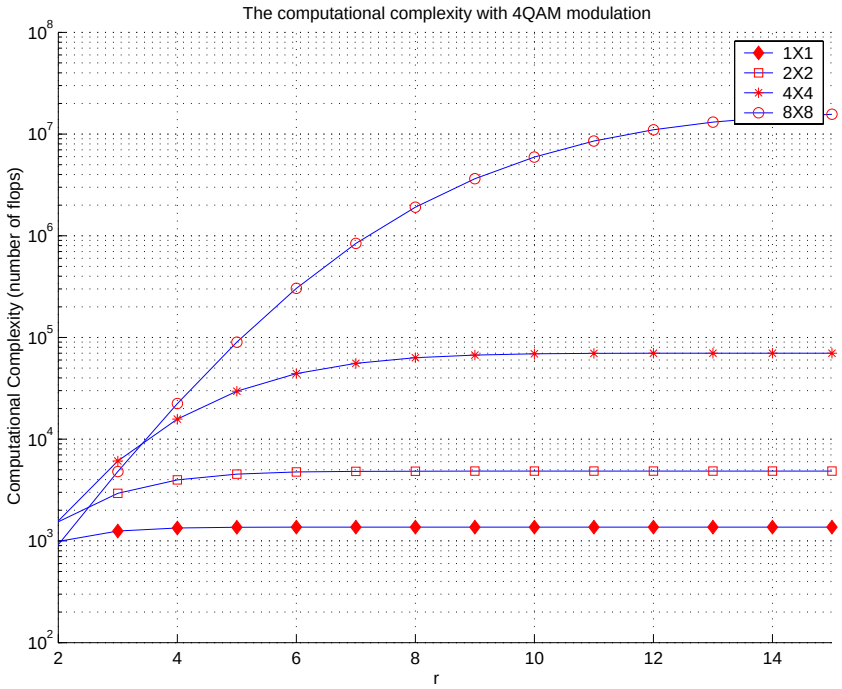

Figure 13: The complexity with 4-QAM modulation, the SNR per transmit antenna is 10 dB. The graph shows the general trend as the search radius increases

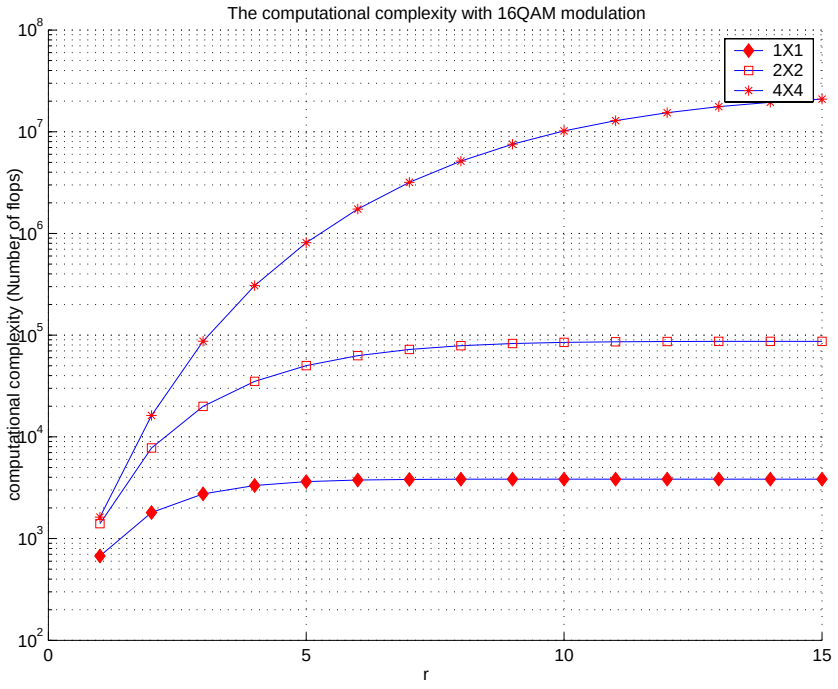

Figure 14: The complexity with 16QAM modulation

In Fig. 13, the number of computations used in systems with different N_t and N_r is shown. For the same search radius R, the computational complexity increases as the number of transmitter and receiver antennas increases as expected. In addition, for a particular system, increasing R will lead to larger computational complexity as well. When the radius R reaches a certain limit, all constellation points are included in the sphere. After that, further increase on R will not increase the computational complexity. This is observed in the figures. Moreover, increasing the constellation size has similar effect as increasing N_t or N_r, as Fig. 14 shows.

The analysis in complexity offers a way to estimate the amount of computation, which can be used to estimate the system processing time and decide the applicability even before actual implementation.(For example, for a system with $N_t = 8$, $N_r = 8$, and 4-QAM, given a chip with a processing speed of 10^6, the search radius should be around 6 or 7 in order to achieve the rate of 16 bits/sec. Even for this reduced complexity design,a system with large constellation size or large amount of parallel antennas may still require unmanageable size of computing to maintain an reliable communication. Thus, this kind of estimation is quite helpful for system implementation.

9.0 CONCLUSION

We have proposed a novel transmitter-receiver system design with a new reduced complexity tree-search based soft-input soft-output equalization method which can be combined in to the graph-decoder in turbo-iterative manner. The simple but novel compensation rule allows the generation of soft-output messages and contributes to a significant reduction in the bit error rates. In overall, we have presented an enabling transmitter and receiver pair which may be useful in deriving up the spectral efficiency dramatically, into the region of tens of bits/sec/Hz, employing bigger signal constellations and more number of transmit antennas. The increase in the signal processing complexity is handled by the proposed procedure of reduced complexity schemes. The quality performance of the proposed receiver is insured with the use of turbo-iterative decoding and equalization steps, and the use of powerful block code. We notice that from the pair-wise error analysis, the low-density parity-check codes posses a large minimum Hamming distance and thus achieve very high order of signal diversity from all dimensions, including space, time, and frequency. Furthermore, based on the pair-wise error probability, we provide an analytical framework which enables us to calculate an upper bound on the overall error probability, ensemble-averaged over the fading channel. The bounds show that the simulation results of the

low-complexity scheme deviates from the bounds only within a few dB. Finally, a delineation of different signal processing complexities upon choosing different system parameters is presented. Future work includes the study of finding the direct trade-off relationship of the signal-processing complexity to the probability of errors and to the capacity of the channels.

APPENDIX

In this section, we offer the detailed proof of the compensation rule. First, define

$$\hat{\alpha}^{m'}_{x^i_{(k)}} = \sum_1^k \hat{\gamma}^{m'}_{x^i_{(k)}}, \tag{.1}$$

where $\hat{\gamma}^{m'}_{x^i_{(k)}}$ is the Euclidean distance in a particular time section. The amount of compensation for each time section will be $\hat{C}$. For those surviving-to-end paths, $\hat{\beta}$ will be the cumulative metric backward for each transition inside each path

$$\hat{\beta}^{m'}_{x^i_{(k)}} = \sum_{j=k+1}^{N} \hat{\gamma}^{m',m}_{x^i_{(j)}}. \tag{.2}$$

For those leaves, one compensation will be assigned for each time section until the end of the codeword, as indicated in the figure. The more compensation, the more distance with the best path.

As part of the reduced complexity algorithm, the compensation will not involve much complexity in computation. So the $\hat{C}$ will be uniform for same time section.

$$\hat{\beta}_{x_{(k)}} = \sum_{j=k+1}^{N} \hat{C}_j \tag{.3}$$

The compensation principle is

- $\hat{C}$ should satisfy that the additional distance will be larger than the worst survivor in this time section.
- $\hat{C}$ should small enough to get them fairly compete with survivors.

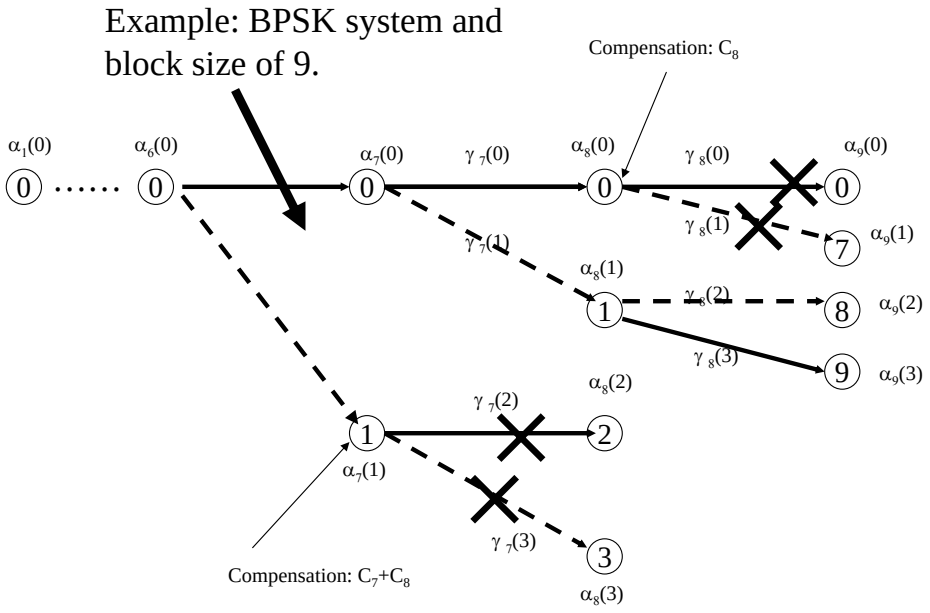

Figure 15: The example with block length of 9.

An example is illustrated in the Fig. 15. The analysis is concentrate on the last 3 time sections out of block length of 9.

In this example, α is the cumulative metric of γ:

$$\begin{aligned} \alpha_9(0) &= \alpha_8(0) + \gamma_8(0) \\ \alpha_9(2) &= \alpha_8(0) + \gamma_8(1) \\ &\dots \end{aligned} \tag{.4}$$

In time section 7.8.9, we define

$$\begin{aligned} \sigma &= \min_i(\alpha_9(i)) \\ \lambda &= \min_i(\alpha_8(i)) \\ \theta &= \min_i(\alpha_7(i)). \end{aligned} \tag{.5}$$

Survived pathes satisfy

$$\begin{aligned} \alpha_8(0), \alpha_8(1) &< \lambda + T' \\ \alpha_9(2), \alpha_9(3) &< \sigma + T'. \end{aligned} \tag{.6}$$

where T' is the constant term in (.6).

$\hat{C}_7$ and $\hat{C}_8$ represents the compensation for nodes ended in 7^{th} and 8^{th} time section. Therefore:

$$\begin{aligned} \alpha_8(0) + \hat{C}_8 &> \sigma + T' \\ \alpha_7(1) + \hat{C}_7 &> \lambda + T' \\ \alpha_7(1) + \hat{C}_7 + \hat{C}_8 &> \sigma + T', \end{aligned} \tag{.7}$$

Given these requirement, we consider the worst situation, which means

the compensation should be greater than the worst survivor

$$\begin{aligned} \alpha_8(1) &= \sigma \\ \alpha_8(0) &= \sigma \\ \alpha_9(3) &= \lambda + T'. \end{aligned} \tag{.8}$$

This situation indicates the best survivor's leaves all get pruned in the next time section. Therefore

$$\alpha_8(1) + \hat{C}_8 > \alpha_9(3) = \alpha_8(1) + \lambda - \sigma + T'. \tag{.9}$$

That is:

$$\hat{C}_8 > \lambda - \sigma + T'. \tag{.10}$$

For the 7^{th} time epoch

$$\begin{aligned} \hat{\beta}_{x_{(7)}} &= \hat{C}_8 + \hat{C}_7 \\ &= \lambda - \sigma + T' + \sigma - \theta + T' \\ &= \lambda - \theta + 2T'. \end{aligned} \tag{.11}$$

Generally, the compensation will be

$$\begin{aligned} \hat{\beta}_{x_{(k)}} &= \sum_{j=k}^{N} \hat{C}_k \\ &= \min_{m}(\alpha_N(m)) - \min_{m'}(\alpha_j(m')) + (N-k) \cdot T', \end{aligned} \tag{.12}$$

where N is the block length, and k is the time epoch this path ends

This formula gives the bound of legal compensation, but this amount neglect the differences among pruned paths. Unlike assume all pruned paths

have the near-surviving parameter, another factor $F \geq 1$ employed will be even more helpful:

$$\hat{\beta}_{x_{(k)}} = F \cdot \min_m(\alpha_N(m)) - \min_{m'}(\alpha_j(m')) + (N-k) \cdot T'. \tag{.13}$$

BIBLIOGRAPHY

[1] Lizhong Zheng, David N. C. Tse, "Diversity and Multiplexing: A Fundamental Tradeoff in Multiple-Antenna Channels" *IEEE Trans. Info. Theory*, vol. 49, No. 5 pp. 1073-96, May 2003.

[2] R. Fano, "A heuristic discussion of probabilistic decoding" *IEEE Trans. Info. Theory*, vol. 9, No. 2 pp. 64-74, Apr. 1963.

[3] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Info. Theory*, vol. 20, pp. 284–7, Mar. 1974.

[4] S.J. Simmons, "Breadth-first trellis decoding adaptive effort," *IEEE Trans. Commun.*, vol. 38, pp. 3–12, Jan 1990.

[5] Bertrand M.Hochwald , Stephan ten Brink, "Achieving Near-Capacity On a Multi-antenna Channel," *IEEE Trans. Commun.*, vol. 51, pp. 389–399, Mar. 2003.

[6] U. Fincke and M.. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,", *Mathematics of computation*, vol.44, pp. 463-471, April 1985.

[7] I.E.Telatar, "Capacity of multi-antenna Gaussian channels," *Euro Trans. Telecom.*, vol. 10, pp. 585–595, Nov. 1999.

[8] V. Tarokh, N. Seshadri, A. R. Calderbank, "Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction," *IEEE Trans. Commun.*, vol. 44, pp. 744–765, Mar. 1998.

[9] R. G. Gallager, "Low-Density Parity-Check Codes," *Ph. D dissertation*, MIT, 1963.

[10] R. G. Gallager, "Low density parity check codes," *IRE Trans. Infor. Theory.*, vol. 8, pp. 21-28, 1962.

[11] D. J. C. MacKay and R. M. Neal, Near shannon limit performance of low density parity check codes, *Electron. Lett.*, vol. 33, pp. 457458, Mar. 1997.

[12] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, Design of capacity-approaching irregular low-density parity-check codes, *IEEE Trans. Inform. Theory,*, vol. 47, pp. 619637, Feb. 2001.

[13] Godfrey H. Hardy *Ramanujan Twelve Lectures* , Chelsea Publishing, 1940

[14] Babak Hassibi; Haris Vikalo *On Sphere Decoding Algorithm. I. Expected Complexity*, Technical Report, 2003

[15] Babak Hassibi; Haris Vikalo *On Sphere Decoding Algorithm. II. Generalizations, Second-order Statistics and Applications to Communications*, Technical Report, 2003

[16] G. Foschini and M. Gans "On limits of wireless communications in a fading environment when using multiple antennas" *Wireless Personal Communications*, vol. 6, pp. 311-335, Mar. 1998.

[17] Vivek Gulati and Heung-no Lee, *A low-complexity iterative per-antenna map equalizer for MIMO frequency-selecive fading channels,*, Proc. of IEEE Globecom-2002, vol. 2, pp 1118–1123, 2002.

[18] J. G. Proakis, Digital Communications, McGraw-Hill, New York, 3rd edition, pp. 345-356 1995.