Paper ID # 901480.PDF

# DETECTOR DESIGN FOR HALF-DUPLEX RELAY NETWORKS IN ISI CHANNELS

Yanjie Peng, Andrew G. Klein and Xinming Huang
Worcester Polytechnic Institute
Department of Electrical and Computer Engineering
Worcester, MA

August 14, 2009

## ABSTRACT

*Recently relay networks have attracted a lot of attention for their potential to increase spatial diversity. However, limited attention has been paid to practical detector design and implementation issues. In this paper, we investigate the design of maximum-likelihood detectors for half-duplex amplify-and-forward relay networks in intersymbol interference (ISI) channels. In particular, we study the case when the relay period is long, which results in an effective channel impulse response that is sparse and periodically time-varying. The proposed design provides a good performance/complexity tradeoff.*

## INTRODUCTION

Multipath fading is one of the chief impairments to reliable communication in wireless networks [1]. Historically, such fading has been combated by using time and frequency diversity techniques. Recently, cooperative diversity [2, 3] and relay networks [4] have attracted a lot of attention for their ability to exploit increased spatial diversity available at distributed antennas on other nodes in the system. By intelligent cooperation among nodes in the network which may only have a single antenna, a virtual multiple antenna system can be formed. Indeed, information theoretic results demonstrate that some of the loss associated with using single-antennas can be recuperated by using intelligent cooperation among distributed nodes [5, 6].

While communication via half-duplex relays and/or user cooperation has seen a lot of active research interest in recent years, most of the existing work has largely come from the information theory and coding communities, and with a few exceptions (e.g. [7],[8]) very little research has yet been conducted into the practical detector design and implementation issues. As we know from [8], the maximum-likelihood (ML) detector for half-duplex relays in frequency selective channels can be realized with a whitening filter and a maximum likelihood sequence estimator (MLSE), i.e. a modified Viterbi detector. However, the computational complexity of this detector increases exponentially with respect to the effective channel length. Particularly, the implementation becomes infeasible when the relay period is long.

In this paper, we investigate the design of ML detectors with long relay periods encountered in practice. While a variety of forwarding protocols have been previously proposed, we will consider amplify-and-forward (AF) for its simplicity and reduced implementation cost, and also we will consider the optimistic case where the receiver has perfect channel knowledge. After introducing the system model for the case of AF relays in ISI channels, we will briefly present the ML detector realization based on the Viterbi algorithm (VA). Then we will show that the long relay period produces time spreads which make the effective channel become sparse. Considering this unique property, we will propose a novel detection scheme using the Multitrellis Viterbi Algorithm (MVA) [9]. Finally, we conclude with a complexity analysis and performance simulations demonstrating the superior performance/complexity tradeoff.

## SYSTEM MODEL

The three-node relay system model is shown in Figure 1. Both source and relay can be considered as mobile users, and each has only one antenna. However, the relay can receive the “overheard” information when the source

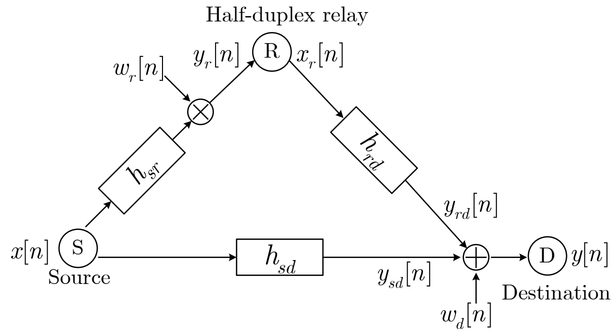


Figure 1: System model

transmits data to the destination, and can then forward it to the destination. Since the channels from source and relay to destination are statistically independent, the three-node cooperative communication scheme effectively forms spatial diversity. We assume that a source transmits a continuous stream of data to a destination, and an AF relay assists the source by amplifying and forwarding the data to the destination.

Relays are mainly two types – full duplex relays that can transmit and receive simultaneously, and half-duplex relays that can either transmit or receive in any time slot. Since full duplex relays are difficult to implement due to self interference if both transmit and receive operations are in the same band, half-duplex is considered more practical for cooperative communication systems. In our system model, we define the half-duplex relay period as $P$ and assume that the relay receives for $P$ symbol periods, and then transmits for $P$ symbol periods. The relay repeats these two tasks alternately.

The source sends the signal $\mathbf{x} = [x[0], x[1], \dots x[N-1]]^T \in \mathbb{C}^N$, where $N$ is the number of transmitted symbols. Each channel is modeled as a linear time-invariant finite impulse response (FIR) filter together with complex additive white Gaussian noise (AWGN). We assume that the effect of pulse shaping is included in the channels. The source-destination, source-relay, and relay-destination channel impulse responses are denoted by $\mathbf{h}_{sd}$, $\mathbf{h}_{sr}$, $\mathbf{h}_{rd}$, respectively, and they have corresponding channel lengths $L_{sd}$, $L_{sr}$ and $L_{rd}$ (e.g. $\mathbf{h}_{sd} = [h_{sd}[0], h_{sd}[1], \dots, h_{sd}[L_{sd}-1]]^T$). The signals $\mathbf{w}_r$ and $\mathbf{w}_d$ are AWGN at the relay and the destination with variance $\sigma_r^2$ and $\sigma_d^2$, respectively.

The destination receives the superposition of the two signals from the source and the relay. The received signal can be expressed as

$$\mathbf{y} = \mathbf{y}_{sd} + \mathbf{y}_{rd} + \mathbf{w}_d \tag{1}$$

where $\mathbf{y}_{sd} \in \mathbb{C}^{N+L_{sd}-1}$ is the contribution from the source and $\mathbf{y}_{rd} \in \mathbb{C}^{N+L_{sr}+L_{rd}-2}$ is the contribution from the relay.

We first consider the source-destination link. The contribution from source to destination is written as

$$\mathbf{y}_{sd} = \mathbf{H}_{sd}\mathbf{x} \tag{2}$$

where $\mathbf{H}_{sd} \in \mathbb{C}^{(N+L_{sd}-1)\times N}$ is the complex Tœplitz channel convolution matrix defined by $[\mathbf{H}_{sd}]_{i+1,j+1} = h_{sd}[i-j]$, $0 \le i, j$ and $0 \le i-j \le L_{sd}-1$, i.e.

$$\mathbf{H}_{sd} = \begin{bmatrix} h_{sd}[0] & 0 & & \\ h_{sd}[1] & h_{sd}[0] & \cdots & \\ \vdots & h_{sd}[1] & & \\ h_{sd}[L_{sd}-1] & \vdots & & \\ 0 & h_{sd}[L_{sd}-1] & & \cdots \\ & & \vdots & \ddots \end{bmatrix}.$$

For the source-relay-destination link, the corresponding contribution is given by

$$\mathbf{y}_r = \mathbf{H}_{sr}\mathbf{x} + \mathbf{w}_r \tag{3}$$

$$\mathbf{x}_r = \mathbf{\Gamma}\mathbf{y}_r \tag{4}$$

$$\mathbf{y}_{rd} = \mathbf{H}_{rd}\mathbf{x}_r. \tag{5}$$

The Tœplitz channel matrices $\mathbf{H}_{rd} \in \mathbb{C}^{(N+L_{sr}+L_{rd}-2)\times(N+L_{sr}-1)}$ and $\mathbf{H}_{sr} \in \mathbb{C}^{(N+L_{sr}-1)\times N}$ are defined in the same way as $\mathbf{H}_{sd}$, $\mathbf{y}_r \in \mathbb{C}^{N+L_{sr}-1}$ is the signal received by the relay, $\mathbf{x}_r \in \mathbb{C}^{N+L_{sr}-1}$ is the signal transmitted from the relay, and $\mathbf{\Gamma} \in \mathbb{C}^{(N+L_{sr}-1)\times(N+L_{sr}-1)}$ denotes the relay matrix. Note that for the matrix dimensions to be compatible, we require that $L_{sd} = L_{sr} + L_{rd} - 1$; if this is not satisfied, we can append zeros to the appropriate matrix without loss of generality. The function of $\mathbf{\Gamma}$ is to impose the half-duplex constraint by selecting groups of $P$ symbols from $\mathbf{y}_r$ (receiving), scaling these symbols by a factor $\beta$ (amplifying), and then delaying the scaled symbols of $\mathbf{y}_r$ for transmission in the next $P$ symbol block (forwarding), recursively. For example, when $P = 2$,

$$\mathbf{\Gamma} = \beta \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 1 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & \\ \vdots & & & & & & \ddots \end{bmatrix}. \tag{6}$$

Thus $\mathbf{\Gamma}$ is given by

$$\mathbf{\Gamma} = \beta\, \mathbf{I}_{\left(\frac{N+L_{sr-1}}{2P}\right)} \otimes \begin{bmatrix} \mathbf{0}_{P\times P} & \mathbf{0}_{P\times P} \\ \mathbf{I}_{P\times P} & \mathbf{0}_{P\times P} \end{bmatrix} \qquad (7)$$

where $\otimes$ denotes Kronecker product. Here we implicitly require that $N+L_{sr}-1$ is divisible by $2P$. From (3)−(5), we have

$$\mathbf{y}_{rd} = \mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{H}_{sr}\mathbf{x} + \mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{w}_r. \qquad (8)$$

From (1), (2) and (8), the received signal at the destination is expressed as

$$\begin{aligned} \mathbf{y} &= (\mathbf{H}_{sd} + \mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{H}_{sr})\mathbf{x} + \mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{w}_r + \mathbf{w}_d \\ &= \mathbf{H}_{\mathit{eff}}\mathbf{x} + \mathbf{w}_{\mathit{eff}} \end{aligned} \qquad (9)$$

where $\mathbf{w}_{\mathit{eff}} \sim \mathcal{CN}(\mathbf{0}, \sigma_d^2\mathbf{I} + \sigma_r^2\mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{\Gamma}^H\mathbf{H}_{rd}^H)$. Note that $\mathbf{w}_{\mathit{eff}}$ is colored, not white, since the AWGN on the source-relay link get amplified and forwarded over the relay-destination channel which introduces coloring of the noise.

From (6) and (7), the relay matrix $\mathbf{\Gamma}$ has a repetitive structure with a period of $2P$. Accordingly, the channel matrix $\mathbf{H}_{\mathit{eff}}$ shows the same structure as $\mathbf{\Gamma}$. Consequently $\mathbf{H}_{\mathit{eff}}$ is a periodically time-varying channel which consists of $2P$ sets of different channels. In the following section, we will briefly introduce the ML detector to estimate transmitted symbols at the receiver end for the ISI channel model prescribed above.

## MAXIMUM LIKELIHOOD DETECTOR

Assuming that receivers can acquire perfect channel knowledge, MLSE can be employed by searching for the minimum Euclidean distance between observed signal and any given transmitted signals. The Viterbi Algorithm (VA) [10] is an efficient technique for solving the minimum distance problem, and its implementation has been investigated extensively [11, 12, 13, 14, 15]. A Viterbi-based ML detector for half-duplex relay networks was proposed in [8], and here we review its operation in the case of periodically time-varying effective channels with colored noise. The detector shown in Figure 2 consists of a whitening filter and a modified Viterbi detector.

Before detection, a whitening filter $\mathbf{G}^{-1}$ is first employed to whiten the noise. $\mathbf{G}^{-1}$ is obtained by factoring the noise covariance matrix as:

$$\mathbf{G}\mathbf{G}^H = \sigma_d^2\mathbf{I} + \sigma_r^2\mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{\Gamma}^H\mathbf{H}_{rd}^H. \qquad (10)$$

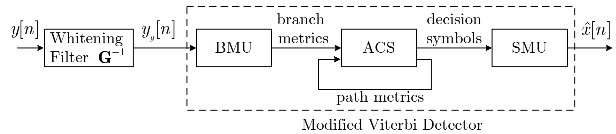


Figure 2: Block diagram of the ML detector

where $\mathbf{G}$ is the Cholesky factorization of the covariance. Then, by filtering the received signal with $\mathbf{G}^{-1}$, the noise becomes whitened since the covariance of the filtered noise $\mathbf{G}^{-1}\mathbf{w}_{\mathit{eff}}$ is given by $E\left[(\mathbf{G}^{-1}\mathbf{w}_{\mathit{eff}})(\mathbf{G}^{-1}\mathbf{w}_{\mathit{eff}})^H\right] = \mathbf{G}^{-1}E[\mathbf{w}_{\mathit{eff}}\mathbf{w}_{\mathit{eff}}^H](\mathbf{G}^{-1})^H = \mathbf{I}$. Ignoring end effects (or, equivalently, taking the block length $N \to \infty$), $\mathbf{G}^{-1}$ follows the same repetitive structure as $\mathbf{\Gamma}$, and thus also shows periodically time-varying property. It naturally defines $2P$ sets of coefficients denoted as $\mathbf{g}_0$, $\mathbf{g}_1$, … $\mathbf{g}_{2P-1}$. Without loss of generality, letting the start time be 0, coefficients adopted by the whitening filter at certain time instant $n$ are given by

$$\mathbf{g}_n = \mathbf{g}_{\mathrm{mod}(n,2P)} \in \{\mathbf{g}_0, \mathbf{g}_1, ..., \mathbf{g}_{2P-1}\}, \quad n = 0, 1, ... \qquad (11)$$

where $\mathrm{mod}(n, 2P)$ is the modulus reminder.

After applying the whitening filter, the received symbols become

$$\mathbf{y}_g = \mathbf{G}^{-1}\mathbf{H}_{\mathit{eff}}\mathbf{x} + \mathbf{G}^{-1}\mathbf{w}_{\mathit{eff}}. \qquad (12)$$

Note that the effective channel matrix $\mathbf{G}^{-1}\mathbf{H}_{\mathit{eff}}$ maintains the periodically time-varying property due to the similar structures of $\mathbf{G}^{-1}$ and $\mathbf{H}_{\mathit{eff}}$. The structure of the effective channel matrix $\mathbf{G}^{-1}\mathbf{H}_{\mathit{eff}}$ is given as

$$\begin{bmatrix} & \vdots & & & & & \\ \ldots & h_0[L-1] & \ldots & h_0[0] & 0 & \ldots & \\ & 0 & h_1[L-1] & \ldots & h_1[0] & 0 & \\ & & & \vdots & \vdots & \vdots & \\ & & & h_{2P-1}[L-1] & \ldots & h_{2P-1}[1] & \ldots \\ & & & 0 & h_0[L-1] & \ldots & \\ & & & & \vdots & & \end{bmatrix}$$

It defines $2P$ sets of effective channel coefficients as $\mathbf{h}_0$, $\mathbf{h}_1$, ..., $\mathbf{h}_{2P-1}$, where $\mathbf{h}_0 = [h_0[0], h_0[1], \ldots h_0[L-1]]^T$, $\mathbf{h}_1 = [h_1[0], h_1[1], ..., h_1[L-1]]^T$,..., and $L$ is the effective channel length.

At time instant $n$, the corresponding channel coefficients are

$$\mathbf{h}_n = \mathbf{h}_{\mathrm{mod}(n,2P)} \in \{\mathbf{h}_0, \mathbf{h}_1, ...\mathbf{h}_{2P-1}\} \quad n = 0, 1, ... \qquad (13)$$

The output signal at the time $n$ is given by

$$s[n] = \sum_{i=0}^{L-1} h_n[i]x[n-i], \quad n = 0, 1, ... \qquad (14)$$

In the branch metric unit (BMU), the branch metrics along the trellis path are not only related to state transitions but also the current time instant. The branch metric calculation is modified as

$$\lambda_{i,j}[n] = |y_g[n] - s_{i,j}[n]|^2, \quad n = 0, 1, ... \tag{15}$$

where $y_g[n]$ is the symbol from the whitening filter at the time instant $n$, $s_{i,j}[n]$ and $\lambda_{i,j}[n]$ is the output signal and the branch metric from state $i$ to state $j$ at the instant $n$, respectively.

The add-compare-select (ACS) unit recursively computes path metrics and decision bits,

$$\Lambda_j[n] = \min_i [\Lambda_i[n-1] + \lambda_{i,j}[n]] \quad n = 0, 1, ... \tag{16}$$

where $\Lambda_j[n]$ denotes the path metric at state $j$ at the instant $n$ and $i$ corresponds to the previous state of $j$. The path metric for each state is updated for the next iteration, and the decision indicating the survivor path for state $j$ are recorded and retrieved from the the survivor-path memory unit (SMU) in order to estimate the transmitted symbols along the final survivor path.

Similar to the traditional MLSE, the implementation cost of the ML detector for relay networks increases exponentially with respect to the effective channel length. With respect to the received signal, the length of channel impulse response follows the larger channel length between the source-destination link and source-relay-destination link. Then the length may be extended by the whitening filter. Hence, we have

$$L \geq \max(L_{sd},\, L_{sr} + L_{rd} + P - 1). \tag{17}$$

It suggests that the constraint length for the MLSE can be very large under a practical situation. For example, when the relay period $P$ is large, use of the optimal detector becomes impractical. Next, we will present a MVA-based ML detector which can efficiently address this problem with negligible performance loss.

## MVA-BASED MAXIMUM LIKELIHOOD DETECTOR

Consider the practical effect of the relay period $P$. Assume a simplified relay network without ISI channels, i.e. $L_{\{sd,sr,rd\}} = 1$. When the relay is in the receiving period, the destination can only receive the signal from the source. The channel impulse response at the destination is shown in Figure 3a. After $P$ time instants, the relay begins to forward the copy of the signal to the destination. Then the channel impulse response is shown in Figure 3b. We can observe that the relay actually introduces ISI even in the non-ISI channels. Also, the large $P$ will increase the number of zero coefficients of the channel impulse response, which makes the effective channels become sparse.

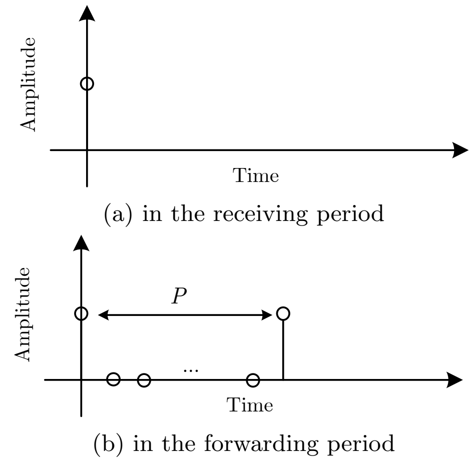


Figure 3: Channel impulse response in different relay periods

Viterbi algorithms for sparse channels have been investigated by several independent researchers. The parallel trellis Viterbi algorithm (PTVA) proposed in [16] reformulated the original single trellis into a set of independent trellises. These independent trellises operate in parallel and have less overall complexity than a single trellis. However, PTVA requires that the channel have equi-spaced coefficients, which usually cannot be satisfied in practice. Although a generalized PTVA is given to deal with general sparse channels, its performance loss is remarkable if the channel is not close to the equi-spaced structure. The multitrellis Viterbi algorithm (MVA) is proposed in [9]. The complexity does not depend on the channel impulse response length but only on the number of nonzero coefficients. However, this algorithm aims at fixed sparse channel. In order to process the sparse time-varying channels for relay networks, the MVA is modified and incorporated in our MVA-based ML detector.

To illustrate the operation of the MVA, let us consider an example. Assume $\mathbf{h}_n \in \{\mathbf{h}_0, \mathbf{h}_1, ...\mathbf{h}_{2P-1}\}$ has few nonzero coefficients, for example $h_n[i] \neq 0$ for $i = 0, K, L-1,\ 0 < K < L-1$. The output signal

Table 1: The dependencies between $x[0]$ and related output signal

| Related output signal | Symbols to be recorded |
|---|---|
| $s[0] = f(\mathbf{h}_0, x[0], x[-K], x[-L+1])$ | $x[0]$ |
| $s[K] = f(\mathbf{h}_K, x[K], x[0], x[K-L+1])$ | $x[0], x[K]$ |
| $s[L-1] = f(\mathbf{h}_{L-1}, x[L-1], x[L-1-K], x[0])$ | $x[0], x[K], x[L-1]$ |
| $s[2K] = f(\mathbf{h}_{2K}, x[2K], x[K], x[2K-L+1])$ | $x[K], x[L-1], x[2K]$ |
| $s[K+L-1] = f(\mathbf{h}_{K+L-1}, x[K+L-1], x[L-1], x[K])$ | $x[K], x[L-1], x[2K], x[K+L-1]$ |
| $s[2L-2] = f(\mathbf{h}_{2L-2}, x[2L-2], x[2L-2-K], x[L-1])$ | $x[L-1], x[2K], x[K+L-1], x[2L-2]$ |
| $s[3K] = f(\mathbf{h}_{3K}, x[3K], x[2K], x[3K-L+1])$ | $x[2K], x[K+L-1], x[2L-2]$ |
| $s[2K+L-1] = f(\mathbf{h}_{2K+L-1}, x[2K+L-1], x[K+L-1], x[2K])$ | $x[2K], x[K+L-1], x[2L-2]$ |
| $s[K+2L-2] = f(\mathbf{h}_{K+2L-2}, x[K+2L-2], x[2L-2], x[K+L-1])$ | $x[K+L-1], x[2L-2]$ |

at the time $n$ is given by

$$s[n] = h_n[0]x[n] + h_n[K]x[n-K] + h_n[L-1]x[n-L+1].$$

When $x[0]$ is under estimation, we can see that $x[0]$ is needed in $s[0]$, $s[K]$, $s[L-1]$. Next, in $s[K]$, a new symbol $x[K]$ appears and it is also needed in $s[L-1]$ and $s[2K]$. In this way, we record all the output signals and symbols related with $x[0]$ in Table 1. Note that some output signals and symbols is not needed when $x[0]$ is under detection, for example, if $K \neq 1$, there is no need to record $s[1]$, since $x[1]$, $x[1-K]$, $x[2-L]$ in $s[1]$ do not affect the estimation of $x[0]$. With the traceback length $L_{tb} = 3(L-1)$, for example, the estimation on $x[0]$ depends by a non-instantaneous relationship on $x[K]$, $x[L-1]$, $x[2K]$, $x[K+L-1]$, $x[2L-2]$, assuming that $x[n]$, $n < 0$ are known.

Notice that some related symbols that appear only once in the first column, e.g. $x[L-1-K]$, do not need to be recorded. Its value can be determined by an instant decision given by

$$\hat{x}[L-1-K] = \underset{\tilde{x}[L-1-K]}{\arg\min} |y_g[L-1] - \tilde{s}[L-1]|$$

where $\tilde{s}[L-1] = f(\mathbf{h}_{L-1}, x[L-1], \tilde{x}[L-1-K], x[0])|$, and $x[L-1]$ and $x[0]$ are known for a given state.

When two or more symbols are determined by the instant decision, e.g. $x[3K]$ and $x[3K-L+1]$ in $s[3K]$, the estimation is given by

$$\{\hat{x}[3K], \hat{x}[3K-L+1]\} = \underset{\tilde{x}[3K], \tilde{x}[3K-L+1]}{\arg\min} |y_g[3K] - \tilde{s}[3K]|$$

where $\tilde{s}[3K] = f(\mathbf{h}_{3K}, \tilde{x}[3K], x[2K], \tilde{x}[3K-L+1])$, and $x[2K]$ is known for a given state.

Table 2: State definition ($x[0]$ under estimation)

| Related time instant | State definition |
|---|---|
| $0$ | $[x[0]]$ |
| $K$ | $[x[K], x[0]]$ |
| $L-1$ | $[x[L-1], x[K]]$ |
| $2K$ | $[x[2K], x[L-1], x[K]]$ |
| $K+L-1$ | $[x[K+L-1], x[2K], x[L-1]]$ |
| $2L-2$ | $[x[2L-2], x[L-1+K], x[2K]]$ |
| $3K$ | $[x[2L-2], x[L-1+K], x[2K]]$ |
| $2K+L-1$ | $[x[2L-2], x[L-1+K], x[2K]]$ |
| $K+2L-2$ | $[x[2L-2], x[L-1+K], x[2K]]$ |

The definition of state depends only on the related time instant. By the list of related symbols, the state definition is derived and given in Table 2. Note that the state definition excludes the symbols assumed to be known, i.e. $x[-K]$, $x[-L+1]$, $x[K-L+1]$, and the symbols that can be determined by the instant decision.

From Table 1 and 2, it is observed that the corresponding trellis shrinks in two dimensions, which leads to a significant reduction in computational complexity. Furthermore, the process of traceback is faster, since for some instant given the current state, the previous state can be obtained immediately without survivor path decision. There are two categories for these instant traceback. First, the previous state definition is a subset of the current state definition. For example, the state at instant $2K$ is defined as $[x[2K], x[L-1], x[K]]$, while the previous state $[x[L-1], x[K]]$ at the instant $L-1$ can be obtained from the current state without the help of the survivor path decision. Second, the state definition is the same for the current and previous state. For example, considering the instants $K+2L-1$, $2K+L-1$, $3K$ and $2L-2$, we can bypass the traceback from $K+2L-2$ to $2L-2$. Once the start state at the instant $K+2L-2$

is available, we can begin to traceback from the instant $2L-2$ at the same state.

When the following symbols are under detection, the structure of the trellis remains the same, except that the branch metric calculation for the first several instants are slightly different, since the initial symbols (e.g. $x[-K]$, $x[-L+1]$, $x[K-L+1]$ for $x[0]$) have been estimated. The available estimated symbols will be used in the calculation of output signal $s[n]$ when needed.

Due to the reduced-size trellis, the detector can be realized by utilizing $L_{tb}$ trellises working in parallel to increase the throughput. The received signals $y_g[n]$ are filled in the $L_{tb}$ trellises sequentially. At the instant $L_{tb}-1$, the first trellis is full and $x[0]$ is estimated. At the instant $L_{tb}$, $y_g[L_{tb}]$ is ready to fill in the first trellis, and also, $x[1]$ is available from the second trellis. Notice that these trellis are similar in structure, however, the channel coefficients used in the branch metric calculation are not the same at different instant. For example, in the first trellis, $\mathbf{h}_0$, $\mathbf{h}_K$, $\mathbf{h}_{L-1}$, ...,$\mathbf{h}_{K+2L-2}$ are used in sequence for each step, while in the second trellis, $\mathbf{h}_1$, $\mathbf{h}_{K+1}$, $\mathbf{h}_L$, ...,$\mathbf{h}_{K+2L-1}$ are used for each step. The corresponding channel coefficients for the received signals $y_g[n]$ are $\mathbf{h}_{\text{mod}(n,2P)} \in \{\mathbf{h}_0, \mathbf{h}_1, ...\mathbf{h}_{2P-1}\}$.

For general Viterbi detectors, complex multiplications dominate the computational cost. There are $M^{L-1}$ states and each state corresponds to $M$ complex multiplications for branch metric calculation. Then the total computational cost is $M^L$. For the proposed MVA-based detector in the sparse channel under consideration, there are $M^3$states at most. The computational cost for each trellis is $M^4$. Thus the total computational cost is $L_{tb} \cdot M^4$. Survivor path decisions are recorded in SMU which consumes significant amount of power. The memory cost for general detector is $k \cdot L_{tb} \cdot \log_2 M \cdot M^{L-1}$. $k$ is the number of memory banks, and its values depends on the traceback scheme used, for example $k = 4$ for 2-pointer traceback. For the proposed detector, the survivor path decision only need to be recorded for 3 instants in each trellis. The total memory cost is $3 \cdot L_{tb} \cdot \log_2 M \cdot M^3$. The comparison on implementation cost between the general detector and the proposed detector is shown in Table 3.

## NUMERICAL RESULTS

We simulate the symbol error rate (SER) performance of the proposed detector under QPSK modulation. The

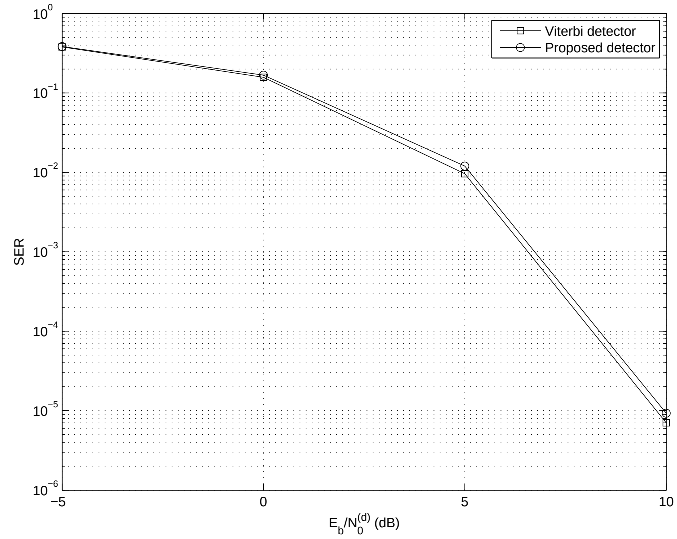


Figure 4: Performance comparison between the general detector and the proposed detector

source signal is i.i.d unit power symbols $x[n] = \{\pm 1, \pm i\}$. We assume fixed ISI channels, and the channels are $\mathbf{h}_{sd} = 1$, $\mathbf{h}_{sr} = [0.58, -0.58 + 0.58i]^T$ and $\mathbf{h}_{rd} = 1$. $E_b/N_o^{(d)}$ and $E_b/N_o^{(r)}$ denote bit-energy-to-noise ratio for the destination and the relay, respectively. Here we assume that $E_b/N_o^{(r)} = E_b/N_o^{(d)} + 10dB$. The relay period $P = 4$, and the transmitted signal from the relay has unit power. The effective channel length $L = 6$. Each effective channel $\mathbf{h}_n \in \{\mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_7\}$ has 3 non-zero coefficients $(\mathbf{h}_n[0], \mathbf{h}_n[4], \mathbf{h}_n[5])$.

The performance of the proposed detector is given in Figure 4. It is shown that performance loss from the MVA-based detector is negligible. From Table 3, the proposed detector reduces the computational cost and memory cost by 6.25% and 95.3%, respectively (assuming $k = 4$).

## CONCLUSION

In this paper, we investigated the design of maximum-likelihood detectors for half-duplex relay networks in intersymbol interference channels. In particular, we studies the case when the relay period is long, so that the half-duplex relay switches between receive and transmit modes infrequently. We showed that the resulting effective channel impulse response in this case exhibits a sparse, periodically time-varying characteristic. We then considered the use of the Multitrellis Viterbi Algorithm [9]. Complexity analysis and numerical result

Table 3: Comparison on implementation cost between the general detector and the proposed detector

| | General detector | Proposed detector |
|---|---|---|
| Computational cost | $M^L$ | $L_{tb}M^4$ |
| Memory cost | $k \cdot L_{tb} \cdot \log_2 M{\cdot}M^{L-1}$ | $3 \cdot L_{tb} \cdot \log_2 M{\cdot}M^3$ |

showed that the proposed detector reduced the computational cost and memory cost significantly with negligible performance loss. Future work will focus on the VLSI implementation of the proposed detector.

# References


[1] J. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2000.

[2] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity, Parts I & II," *IEEE Trans. Commun.*, vol. 51, Nov. 2003.

[3] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[4] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inf. Theory*, vol. 51, pp. 3037–3063, Sep. 2005.

[5] A. Sendonaris, E. Erkip, and B. Aazhang, "Increasing uplink capacity via user cooperation diversity," Aug. 1998, p. 156.

[6] J. N. Laneman, G. W. Wornell, and D. N. C. Tse, "An efficient protocol for realizing cooperative diversity in wireless networks," Jun. 2001, p. 294.

[7] P. Murphy, A. Sabharwal, and B. Aazhang, "Building a cooperative communications system," *IEEE Trans. Inf. Theory*, submitted for publication.

[8] Y. Peng, A. Klein, and X. Huang, "Design of a maximum-likelihood detector for cooperative communications in intersymbol interference channels," in *Proceedings of the 19th ACM Great Lakes symposium on VLSI.* ACM New York, NY, USA, May 2009, pp. 429–432.

[9] N. Benvenuto and R. Marchesani, "The Viterbi algorithm for sparse channels," *IEEE Trans. Commun.*, vol. 44, no. 3, pp. 287–289, Mar. 1996.

[10] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, pp. 260–269, Apr. 1967.

[11] M. Kamuf, V. Owall, and J. B. Anderson, "Optimization and implementation of a Viterbi decoder under flexibility constraints," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 8, pp. 2411–2422, Sep. 2008.

[12] Y. Gang, A. T. Erdogan, and T. Arslan, "An efficient pre-traceback architecture for the Viterbi decoder targeting wireless communication applications," *IEEE Trans. Circuits Syst. I*, vol. 53, no. 9, pp. 1918–1927, Sep. 2006.

[13] G. Feygin and P. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 425–429, Mar. 1993.

[14] R. Cypher and C. Shung, "Generalized trace-back techniques for survivor memory management in the Viterbi algorithm," *The Journal of VLSI Signal Processing*, vol. 5, no. 1, pp. 85–94, Jan. 1993.

[15] P. J. Black and T. H. Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1877–1885, Dec. 1992.

[16] N. C. McGinty, R. A. Kennedy, and P. Hocher, "Parallel trellis Viterbi algorithm for sparse channels," *IEEE Commun. Lett.*, vol. 2, no. 5, pp. 143–145, May 1998.