

PARAMETER ESTIMATION AND TRACKING IN
PHYSICAL LAYER NETWORK CODING

A Thesis

by

MANISH JAIN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2011

Major Subject: Electrical Engineering

PARAMETER ESTIMATION AND TRACKING IN
PHYSICAL LAYER NETWORK CODING

A Thesis

by

MANISH JAIN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Scott L. Miller Alex Sprintson
Committee Members,	Krishna Narayanan Natarajan Gautam
Head of Department,	Costas N. Georghiades

May 2011

Major Subject: Electrical Engineering

ABSTRACT

Parameter Estimation and Tracking in Physical Layer

Network Coding. (May 2011)

Manish Jain, B. Tech., Indian Institute of Technology, Roorkee, India

Co-Chairs of Advisory Committee: Dr. Scott L. Miller
Dr. Alex Sprintson

Recently, there has been a growing interest in improving the performance of the wireless relay networks through the use of Physical Layer Network Coding (PLNC) techniques. The physical layer network coding technique allows two terminals to transmit simultaneously to a relay node and decode the modulo-2 sum of the transmitted bits at the relay. This technique considerably improves performance over Digital Network Coding technique.

In this thesis, we will present an algorithm for joint decoding of the modulo-2 sum of bits transmitted from two unsynchronized transmitters at the relay. We shall also address the problems that arise when boundaries of the signals do not align with each other and when the channel parameters are slowly varying and are unknown to the receiver at the relay node. Our approach will first jointly estimate the timing offsets and fading gains of both signals using a known pilot sequence sent by both transmitters in the beginning of the packet and then perform Maximum Likelihood detection of data using a state-based Viterbi decoding scheme that takes into account the timing offsets between the interfering signals. We shall present an algorithm for simultaneously tracking the amplitude and phase of slowly varying wireless channel that will work in conjunction our Maximum Likelihood detection algorithm. Finally, we shall provide extension of our receiver to support antenna diversity.

Our results show that the proposed detection algorithm works reasonably well, even with the assumption of timing misalignment. We also demonstrate that the performance of the algorithm is not degraded by amplitude and/or phase mismatch between the users. We further show that the performance of the channel tracking algorithm is close to the ideal case i.e. when the channel estimates are perfectly known. Finally, we demonstrate the performance boost provided by the receiver antenna diversity.

To my family, friends and professors

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisors Dr. Scott Miller and Dr. Alex Sprintson for introducing me to research in the area of distributed computing, for being so patient and supportive, especially in the not so good times, for the detailed feedback on my reports and progress on the research work which helped me significantly through my progress in this thesis. This thesis was not possible without your guidance and support.

I would also like to thank the members of my advisory committee, Dr. Krishna Narayanan and Dr. Natarajan Gautam, for their guidance in research and the intriguing questions about my work. Thanks to the staff in the Department of Electrical Engineering for making my academic life here so convenient and to faculty for providing me with the important knowledge needed for research. I would like to thank Dr. Henry Pfister who attended my defense exam substituting for Dr. Krishna Narayanan in his absence, and also for allowing me to run simulations using his server resources.

A special thanks to my friends in the whole "Ganga Gang", Richeek Arya, and Monika Vishwakarma for their support and encouragement which made this work possible. Finally, I am thankful to my parents, both my brothers and sisters-in-laws, my cute little nephew and neice for their love, encouragement and their support.

TABLE OF CONTENTS

CHAPTER	Page	
I	INTRODUCTION	1
	A. Relay Network	1
	1. Standard Transmission	2
	2. Digital Network Coding (DNC)	3
	3. Physical Layer Network Coding (PLNC)	4
	B. Scenario and Assumptions	5
	C. Description of Thesis Chapters	7
II	BACKGROUND AND RELATED WORK	8
	A. Background	8
	1. Wireless Signals	8
	2. Modeling of Wireless Channels	10
	3. Simulation of Wireless Channels	13
	B. Previous Work	14
	1. PLNC with Amplify-and-Forward	14
	2. PLNC with Decode-and-Forward	17
	3. PLNC with Non-Coherent Detection	19
	C. Goals and Motivation	19
III	SYSTEM OVERVIEW, RECEIVER STRUCTURE AND AL- GORITHMS	22
	A. System Model	22
	B. Receiver	22
	1. Algorithm	23
	2. Simulation Results	27
	C. Joint Estimation of Amplitude, Phase and Timing	29
	1. Algorithm	30
	2. Simulation Results	33
	D. Amplitude and Phase Tracking for a Slowly Varying Channel	34
	1. Tracking Algorithm for Single User Case	35
	2. Tracking Algorithm for Double User Case	36
	3. Implementation and Results	38
	E. Receiver Diversity	42

CHAPTER	Page
IV CONCLUSION AND FUTURE WORK	45
REFERENCES	46
VITA	48

LIST OF FIGURES

FIGURE		Page
1	A relay network	1
2	Standard transmission technique	2
3	Digital network coding technique	3
4	Physical layer network coding technique	5
5	Transmitter	8
6	Square pulse	9
7	Multipath fading	10
8	Bathtub type PSD	12
9	Generating bathtub type PSD	13
10	Matched filter output for $a_k = +1$ and $b_k = -1$	24
11	Overlap of pulses $p(t - kT_s)$ and $p(t - lT_s - \tau)$	25
12	Trellis used in MLSE receiver	25
13	Receiver structure	27
14	MLSE performance in terms of probability of error vs SNR (in dB) for different values of τ for $h_A = h_B=1$ (i.e. phases of two signals align with each other)	28
15	MLSE performance in terms of probability of error vs τ for different values of SNR for $h_A = h_B=1$ (i.e. phases of two signals align with each other).	29
16	MLSE performance in terms of probability of error vs phase of second user for SNR = 7 dB, $\tau = 0.2T_s$, $h_A = 1$ and $ h_B = 1$	30
17	Unsynchronized matched filter output for $a_k = +1$ and $b_k = -1$	31

FIGURE		Page
18	MSE in calculating complex fade values for different values of n (for pilot sequence of length 50 bits)	34
19	MSE in calculating timing offsets for different lengths of pilot sequence ($n = 3$)	35
20	Impact of filtering	39
21	Amplitude and phase tracking	41
22	Performance of tracking algorithm	42
23	Performance of receiver with diversity	44

CHAPTER I

INTRODUCTION

A. Relay Network

A relay network refers to a broad class of topologies (commonly used in wireless networks), where the source and destination are interconnected by means of one or more nodes. In such a network the source and destination cannot communicate to each other directly due to their limited wireless ranges or some obstruction, hence the need for intermediate node(s) to relay [1, 2]. It is also possible that the nodes are in wireless ranges on each other, but the channel conditions between the two nodes do not permit reliable data transmission. Wireless relay networks have recently attracted a significant attention from the research community. In this thesis, we focus on a special but important case of a 3-node relay network. Figure 1 depicts a relay network where the two transmitter nodes N_1 and N_3 are not within wireless range of each other and are interconnected by means of a relay N_2 . In order to communicate, N_1 and N_3 need to exchange data packets through the relay node. Let us discuss some of the methods for reliable communication between N_1 and N_3 through the relay node N_2 .

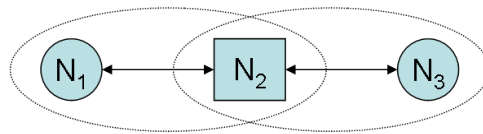


Fig. 1.: A relay network

1. Standard Transmission

The standard (half-duplex) transmission technique is the simplest way to utilize a relay. In this technique, N_1 transmits its data packet to relay N_2 and N_2 forwards it to N_3 . Similarly, the relay N_2 forwards the data packet sent by N_3 , to N_1 . In this way, data packets are exchanged between the nodes N_1 and N_3 . This would require four transmission slots as shown in Figure 2.

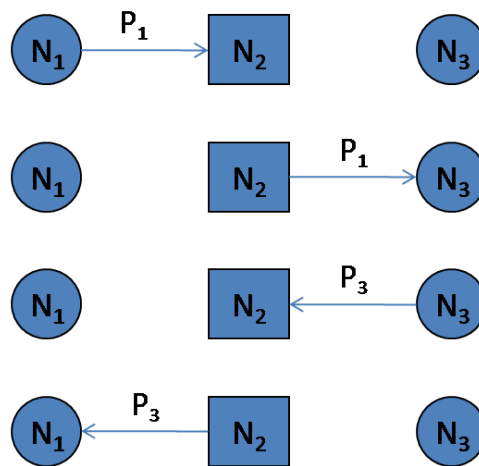


Fig. 2.: Standard transmission technique

It is worth noting that with this approach, each transmission slot involves reception of signals without any interference. Hence, the receiver has a simpler structure. However, it will be shown in the subsequent sections that the same communication can be done in 3 time slots which is achieved with an increase in some complexity at receivers (either the relay node, the transmitting nodes or both).

2. Digital Network Coding (DNC)

The throughput of a relay network can be increased by employing digital network coding (DNC) [3, 4]. DNC utilizes the broadcast property of wireless channels which allows the relay node to simultaneously transmit a data packet to both N_1 and N_3 . In this approach, the transmitter nodes first send their data packets P_1 and P_3 , respectively, to the relay node. The relay node then decodes the bits corresponding to both the packets and broadcasts a re-modulated packet resulting from a linear combination of the received data bits. Bitwise exclusive-OR is an example of such a linear combination. Each transmitter buffers the packet it sent and hence, can deduce (from the linear combination) the data bits transmitted by the other node. The overall communication takes 3 time slots as shown in Figure 3, hence improving the throughput of the wireless network by 25% over the standard transmission method.

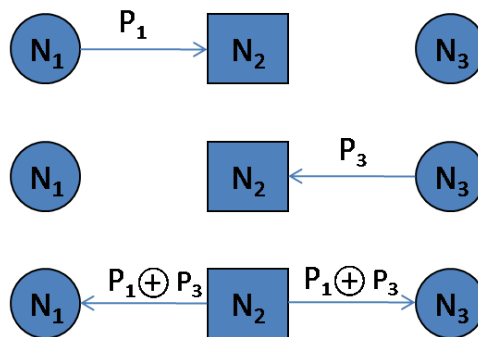


Fig. 3.: Digital network coding technique

The trade-off here is between the throughput (or the number of slots used for communication) and the complexity of the relay node. The relay node would now decode both the data packets and evaluate the modulo-2 sum or the result from the exclusive OR operation of the data packets and then, forms a data packet resulting

from modulation of the resultant data bits. It is worth noting here that the complexity at the receiver is also accompanied by a small delay (due to the exclusive-OR operation of P_1 and P_3), and thus, the gain in throughputs are slightly lesser than 25%.

3. Physical Layer Network Coding (PLNC)

The throughput of a relay network can be further increased by using Physical Layer Network Coding (PLNC). This technique takes advantage of the mixing properties of the wireless signals. In this technique the transmitter nodes would simultaneously transmit to the relay node which would receive the analog sum or real addition of the two signals. The received signal can also be seen as the sum of two sinusoids with different phases, amplitudes and time delays. The relay can then either broadcast the received signal after amplification (called *amplify-and-forward technique*[5]), or can decode the modulo-2 sum of the bits from the received signal and then broadcast a signal constructed from the modulo-2 sum of the bits (called *decode-and-forward technique*[6]). Regardless of the technique used, PLNC provides a throughput improvement of 33% over DNC and 50% improvement over the standard transmission method, as shown in Figure 4. The significant gains in throughput over standard technique and DNC are achieved by introducing substantial complexity at the relay node. The increase in complexity, however, still permits reliable transmission if the modulo-2 sum is accurately estimated at the relay. The performance analysis of such a system is done in [7]. The complexity introduced in the reception of signals at relay node causes an increase in decoding delay, resulting in reduction of the throughput gains by a small amount. The whole system, however, shows drastic improvements over the standard transmission techniques and DNC.

There are many techniques introduced to utilize the aspect of PLNC like amplify-and-forward, decode-and-forward, denoise-and-forward etc. The amplify-and-forward

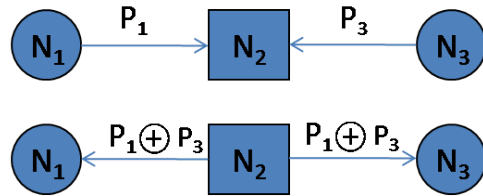


Fig. 4.: Physical layer network coding technique

technique involves amplification of the signal received at the relay and broadcasting the received sum to the transmitter nodes. This means that all the decoding and signal processing has to be performed by the transmitter nodes instead of the relay, hence lowering the computational load on the relay. On the other hand, the *decode-and-forward* technique involves the decoding and processing to be done at the relay. The relay would decode the modulo-2 sum of the bits from the received signal and then broadcast the signal generated from the modulo-2 sum of the bits, thus making it similar to DNC.

B. Scenario and Assumptions

In this thesis work, it is assumed that the relay network divides time into fixed length slots and data is transmitted in packets that will fit into a time slot. Each data packet consists of a preamble of bits known to both the transmitter and receiver followed by the information carrying data bits which would obviously be known to the transmitter but not to the receiver. In a time slotted relay network, each node would need to synchronize its internal clock to some common reference so that their packet transmissions will align with the appropriate time slot and not overlap (interfere) with the transmissions of other nodes in adjacent time slots. However, due to varying propagation delays between the transmitting nodes and the relay, it may be difficult

for the transmitting nodes to synchronize their transmissions to the point where the two packets received at the relay node align in time to a precision that is significantly smaller than a data symbol interval. One of the main contributions of this thesis is to examine the consequences of having the two transmitting nodes imprecisely aligned and also to present algorithms for dealing with the detection of the data in such a situation.

In order to motivate the set of assumptions used in this work, consider a scenario where two mobile nodes, N_1 and N_3 , are communicating through a fixed relay node, N_2 . Furthermore suppose that the distance between the mobile nodes and the relay can be as much as a few kilometers so that variations in the propagation delays between the mobile nodes and the relay node may be as much as about 10 microseconds. We could then transmit data at rates up to something on the order of 100kbps without encountering propagation delays of more than a single bit interval. Further, if we assume that the time slots are on the order of a few tens of milliseconds, then each data packet may contain a few thousand data bits. Finally, fading rates encountered by the system may be as much as perhaps a hundred Hz or possibly slightly higher depending on what frequency band is used. In any event, we would expect the fading rate to be around 3 orders of magnitude smaller than the data rate.

Based on the generic scenario described in the previous paragraph, we make the following assumptions about the nature of the relay network. When the two mobile nodes are simultaneously transmitting data packets to the relay node, they attempt to align their transmissions as best as they can but are only able to synchronize their timings to within +/- one symbol interval. No attempt is made to synchronize their transmissions at the carrier level. That is, there is an arbitrary and independent phase offset on each of the received signals at the relay. Each signal experiences fading which is modeled using the traditional Rayleigh fading model with a "bathtub" type spectral

density commonly used in land mobile environments. The normalized Doppler rate of the fading is assumed to be around 0.1%. Thus, for a data packet of thousands of bits, the fading may change substantially throughout the course of a packet and hence cannot be treated as constant. The timing offsets of the two users signal, however, will vary at a much slower rate and can reasonably be treated as constant throughout the full duration of a data packet.

C. Description of Thesis Chapters

Chapter II describes the related work and gives a background overview, which will help in better understanding of the thesis. The proposed algorithms and receiver architecture is described in Chapter III, along with the performance simulation and results. It also involves the discussion of results and further modifications and improvements. The conclusion and future work possibility is presented in Chapter IV.

CHAPTER II

BACKGROUND AND RELATED WORK

A. Background

Before delving into the main thesis topic, let us review the background material which will prove essential for understanding the thesis.

1. Wireless Signals

In this section, we are going to discuss how signals are modulated and sent through the wireless channel. A simple transmitter in a communication system is shown in Figure 5. For the simplicity of understanding, let us take the example of a

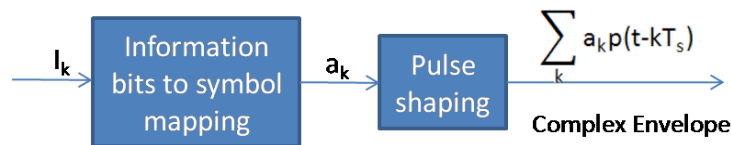


Fig. 5.: Transmitter

digital communication system where a square pulse is used:

$$p(t) = \begin{cases} \frac{1}{\sqrt{T_s}} & \text{if } 0 \leq t < T_s \\ 0 & \text{otherwise,} \end{cases}$$

where T_s is the symbol duration. The pulse shape is shown in Figure 6. Typically, a square wave would waste bandwidth. For the same reason, spectrally efficient pulse shapes like RRC (Root Raised Cosine) pulses are used in practical scenarios. Let us assume that information bits to be transmitted are given by I_k , using a Binary Phase Shift Keying (BPSK) modulation format. For a BPSK modulation format, the

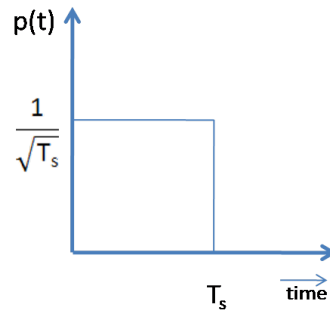


Fig. 6.: Square pulse

symbols transmitted are ± 1 . Hence, a mapping is required from the information bits I_k to the BPSK symbols a_k . The standard mapping is that $I_k = 0$ maps to $a_k = -1$ and $I_k = 1$ maps to $a_k = +1$. The BPSK symbols are then filtered through the pulse shaping to form the complex envelope of the signal. It is worth noting that the actual transmission is done at a very high carrier frequency, and would look like a sinusoid. The actual signal transmitted is a passband version of this signal which is obtained by modulating the baseband signal by a very high carrier frequency. It is just for the ease of analysis, that we consider only the complex envelope or the baseband version of the signal.

At receiver, the received signal is correlated with the a pulse that is matched to the pulse shape used at the transmitter i.e. the conjugated time reversed version of pulse shape used at transmitter. The output is sampled then, at the appropriate time instants such that the Signal to Noise Ratio (SNR) is maximized. An appropriate symbol detection technique is required then, depending upon the type of noise added, modulation format and other fading parameters to decode the information bits.

2. Modeling of Wireless Channels

A wireless channel is usually modeled by multiple paths between the transmitter and receiver. The receiver may receive the signal directly through a line of sight path or a reflected, refracted and diffracted version of it. An example of such a scenario is shown in Figure 7. Each path may lead to same signal being received with multiple delays. Each path can have possibly different attenuation (and phase shift), which may also vary with time. The relative phases of different paths may align with each other and add constructively or may cancel each other, resulting in fading.

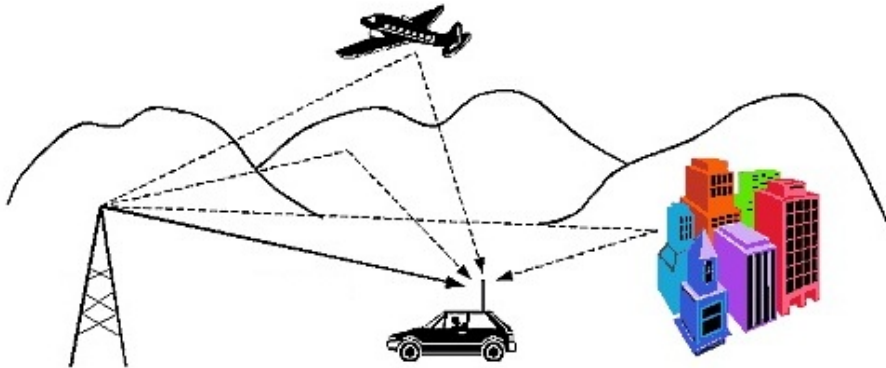


Fig. 7.: Multipath fading

Suppose the signal transmitted has the complex envelope $g_s(t)$. The receiver receives

$$g_r(t) = \sum_k \rho_k e^{j\theta_k} g_s(t - \tau_k) + n(t) \quad (2.1)$$

where ρ_k , θ_k and τ_k are the attenuation, phase shift and delay of the k^{th} path, respectively and $n(t)$ is Additive White Gaussian Noise (AWGN). If the maximum relative delay between the paths (called the delay spread of the channel) is less than the symbol duration, then the channel is not dispersive and may not cause signif-

icant Inter-symbol Interference (ISI). With this assumption, Equation (2.1) can be simplified to

$$g_r(t) = g_s(t) \sum_k \rho_k e^{j\theta_k} + n(t). \quad (2.2)$$

By defining $h_k = x_k + jy_k = \rho_k e^{j\theta_k}$, Equations (2.2) can be re-written as:

$$g_r(t) = g_s(t) \sum_k (x_k + jy_k) + n(t) \quad (2.3)$$

Now, if no path has substantially smaller attenuation than other paths i.e. there is no line of sight path communication possible and if the number of paths is large, then by applying the central limit theorem for x_k and y_k , we can deduce that $x = \sum_k x_k$ and $y = \sum_k y_k$ can be approximated as Gaussian random variables. The received signal can then be written as:

$$g_r(t) = hg_s(t) + n(t)$$

where $h = x + jy = \rho e^{j\theta}$ is a zero-mean complex Gaussian random variable. It is worth noting that x and y are real Gaussian random variables and their distribution is given by:

$$\begin{aligned} f_{x,y}(x,y) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \\ f_\rho(\rho) &= \frac{\rho}{\sigma^2} \exp\left(-\frac{\rho^2}{2\sigma^2}\right) && \forall \rho > 0 \\ f_\theta(\theta) &= \frac{1}{2\pi}, && 0 \leq \theta < 2\pi \end{aligned}$$

This type of fading is known as Rayleigh fading and is typically encountered in land mobile channels in urban areas, where light of sight is rare.

Now let us concentrate on the effects of motion of the transmitter or receiver.

As a direct consequence of the Doppler effect, the received signal will be shifted in frequency relative to the transmitted frequency when the receiver is in motion. The amount of Doppler shift will depend upon the relative motion and the direction of arrival of the received signal. Thus, different angles of arrival will cause the signal to be received at different frequencies around the carrier frequency. It has been observed

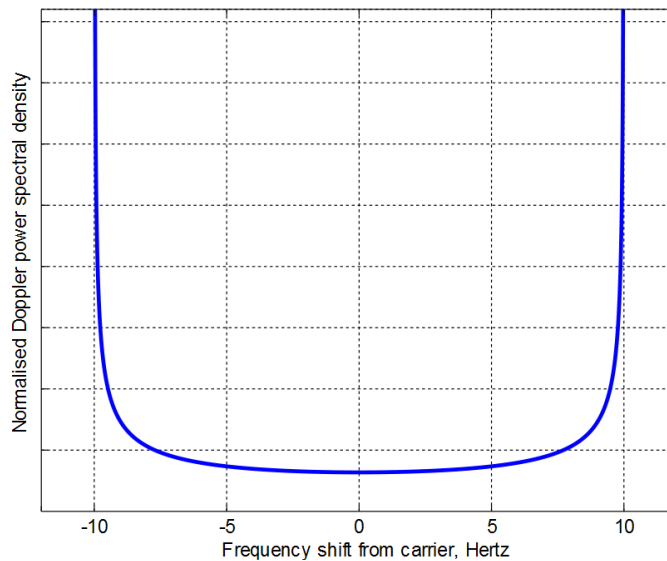


Fig. 8.: Bathtub type PSD

that because of this phenomenon, the Power Spectral Density (PSD) of the received signal looks like a "bathtub", as a function of frequency, as shown in Figure 8. The derivation of this bathtub type spectral density is given in [8]. The maximum Doppler spread is determined by speed of motion of the receiver.

Therefore, a wireless channel gains are modeled as a complex Gaussian random variables and the PSD of the signal has the form

$$S_{XX}(f) = \frac{P_R}{\pi\sqrt{1 - (f/f_d)^2}}, \quad |f| < f_d$$

where f_d is the maximum Doppler frequency and P_R is the total receiver power.

3. Simulation of Wireless Channels

There are many ways to simulate the fade values as modeled in the previous section. We are going to focus on one of them, which we will refer to as the Frequency Domain Method. For the simulations in this thesis involving generation of wireless channel parameters, this method has been used.

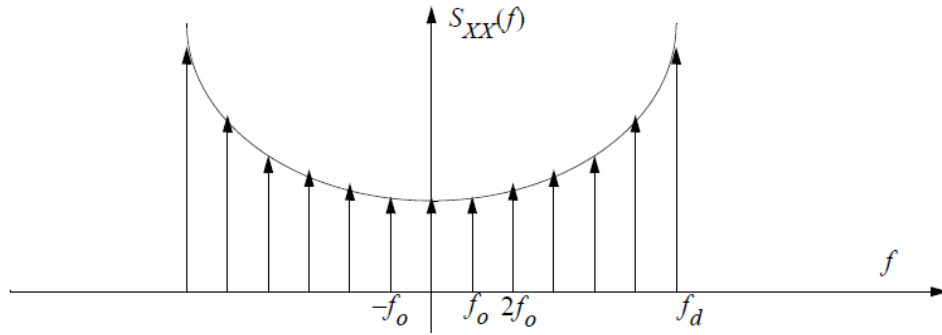


Fig. 9.: Generating bathtub type PSD

Let us say, we are required to generate a realization (say $x(t)$) of this random process over some time interval $(0, T)$. A periodic signal $\hat{x}(t)$ can be produced by repeating $x(t)$ every T seconds. Since, $\hat{x}(t)$ is periodic, it has a Fourier Series representation

$$\hat{x}(t) = \sum_k X_k e^{jk\omega_o t}, \quad \omega_o = \frac{2\pi}{T}$$

The line spectrum is then given by

$$S_{\hat{x}, \hat{x}}(f) = \sum_k \sigma_k^2 \delta(f - kf_o), \quad \sigma_k^2 = E[|X_k|^2]$$

where $\delta(f)$ is the Kronecker delta function. Now, the σ_k can be chosen to shape the PSD into any form and the $X(k)$ can be generated as complex Gaussian random variables.

For generating a "bathtub" type spectral density, we want $\sigma_k \propto \frac{1}{\sqrt{f_d^2 - (kf_o)^2}}$. This

procedure is shown in Figure 9. The proportionality constant can be chosen such that the random process is normalized to have unit power. Taking the DFT (Discrete Fourier Transform) of $X(k)$, shall produce the required time domain random process.

Thus, the above mentioned method allows us to simulate a Rayleigh Fading channel for any desired maximum Doppler frequency.

B. Previous Work

The application of PLNC to the problem of relay networks has seen many techniques introduced providing substantial gains in throughput. Primarily, there are two techniques proposed for PLNC: *decode-and-forward* and *amplify-and-forward*. Let us discuss some of the previous work done with these techniques.

1. PLNC with Amplify-and-Forward

In this section, we will discuss the method used in [5]. The paper considers the reception of two an MSK (Minimum shift keying) modulated signals at the relay. MSK is a binary frequency shift keying, in which waveforms corresponding to the two symbols are orthogonal to each other [9]. Essentially in the absence of noise, the phase of the baseband signal changes by $\pm 90^\circ$ depending upon whether the bit transmitted for the corresponding time interval is 1 or 0. In other words, we can say that the bit information is stored in the phase differences between the beginning and end of symbol duration.

The baseband signal received at the relay (as stated in [5]) :

$$\begin{aligned}
 y[n] &= s_A[n] + s_B[n] \\
 &= h_A e^{i\theta_s[n]} + h_B e^{i\phi_s[n]} \\
 &= |h_A| e^{i(\theta_s[n] + \angle h_A)} + |h_B| e^{i(\phi_s[n] + \angle h_B)},
 \end{aligned}$$

where $\theta_s[n]$ and $\phi_s[n]$ are the phases of the signals transmitted and h_A , h_B are the channel parameters for the two nodes respectively. As stated above, the bit information is stored in the phase $\theta_s[n]$ and $\phi_s[n]$ for MSK modulation scheme. For a single user case, a parameter r can be defined as :

$$\begin{aligned}
 r &= \frac{s_A[n+1]}{s_A[n]} \\
 &= \frac{h_A e^{i\theta_s[n+1]}}{h_A e^{i\theta_s[n]}} \\
 &= e^{i(\theta_s[n+1] - \theta_s[n])}.
 \end{aligned}$$

Assuming the channel phase does not change dominantly across consecutive symbols, the phase of r is proportional to the phase difference $\theta_s[n+1] - \theta_s[n]$. The n^{th} bit is 1 if this quantity is greater than 1, or else the the bit is zero. The general rule of decoding in presence of noise for such a scheme is that positive phase differences map to "1" and negative phase differences map to "0". Hence, the phase estimation for the channel is not required.

Now coming back to the two user case, the relay forwards the signal $y[n]$ to the nodes. Since we need only the successive phase differences $\Delta\theta$ and $\Delta\phi$ to decode the data bits transmitted by each user, the algorithm tries to find the pair $(\Delta\theta, \Delta\phi)$ which could have resulted in

$$y[n] = |h_A| e^{i(\theta_s[n] + \angle h_A)} + |h_B| e^{i(\phi_s[n] + \angle h_B)}. \quad (2.4)$$

where $y[n]$ is the n^{th} sample of the received signal. It is important to note here that the $\angle h_A$, $\angle h_B$ are not significant because the data information is stored in the phase differences. Hence, without loss of generality, we can assume $\theta'_s[n] = \theta_s[n] + \angle h_A$ and $\phi'_s[n] = \phi_s[n] + \angle h_B$, and re-write Equation (2.4) as:

$$y[n] = |h_A|e^{i\theta'_s[n]} + |h_B|e^{i\phi'_s[n]}. \quad (2.5)$$

Using simple aspects of complex algebra, it can be shown that the possible values of $\theta'_s[n]$ and $\phi'_s[n]$ satisfying Equation(2.5), given $y[n]$, $|h_A|$ and $|h_B|$ are:

$$\theta'_s[n] = \arg(y[n](|h_A| + |h_B|D \pm i|h_B|\sqrt{1 - D^2})) \quad (2.6)$$

$$\phi'_s[n] = \arg(y[n](|h_B| + |h_A|D \mp i|h_A|\sqrt{1 - D^2})), \quad (2.7)$$

where, $D = \frac{|y[n]|^2 - |h_A|^2 - |h_B|^2}{2|h_A h_B|}$, $|y[n]|$ is the norm of complex number $y[n]$ and \arg is the angle of a complex number.

Hence, the algorithm first provides an estimation technique for A and B. It is worth noting that A and B are just amplitudes of the channel gains, and there is no attempt made to estimate the phase change provided by the channel. For the n^{th} sample, using A, B, $y[n]$ and Equations (2.6) and (2.7), two possible pairs of $(\theta_s[n], \phi_s[n])$ are estimated. Two pairs are similarly found for $n + 1^{th}$ sample, which results in four possible combinations for the pair $(\Delta\theta, \Delta\phi)$ for the n^{th} bit. At N_1 , the data packet P_1 is buffered and hence, N_1 has the knowledge of exact phase difference $\Delta\theta_1$ for each bit. The pair for which $\Delta\theta_1$ is closest to $\Delta\theta$ is selected, and the corresponding $\Delta\phi$ provides the knowledge of data bits sent by N_3 .

The paper assumes that there is symbol boundary level synchronization (i.e. the symbol boundaries align) but addresses the issue that arises due to symbol level misalignment i.e, n^{th} bit of N_1 may interfere with m^{th} bit of N_3 , where $m \neq n$. A method

is also proposed to differentiate between the interference-free and interfered signal. The research work in this paper has the following limitations:

- Symbol boundary level synchronization is assumed. The paper does not propose a solution to counter the case when the boundaries of both users' signals are not aligned with each other.
- The algorithm is strongly dependent on the characteristics of MSK. The paper does not provide a method to generalize the algorithm for practical modulation formats like QPSK, QAM etc. The spectral efficiency is compromised because of the use of MSK.
- It is assumed that the channel is assumed to be fairly constant throughout each transmission period.

2. PLNC with Decode-and-Forward

The authors in [6] have proposed a technique based on a decode-and-forward solution. This research work assumes the lack of symbol boundary level synchronization which is not addressed in previous work. The receiver in this paper uses a Viterbi decoder which also accounts for the timing offset that arises due to lack of symbol boundary level synchronization.

The work involves transmission of BPSK modulated signals from both the receivers using a square pulse. The timing offset and the channel gains are not known at the receiver. The algorithm starts with choosing an arbitrary value of gains, phases and timing offset. Let us assume that the channel gains are h_A and h_B , and the relative timing offset is τ . The received baseband signal is passed through a matched filter and sampled at time instants $t = kT_s$ and $t = kT_s + \tau$. The matched filter output is denoted as $r_p(t)$. The proposed algorithm then involves computation of an

appropriate metric for implementing a Maximum Likelihood Sequence Detector receiver involving a trellis search using a Viterbi algorithm. The trellis involves 4 states, each corresponding to $\{a_k, b_k\} = \{\pm 1, \pm 1\}$. The branch metric for transition from state $\{a_k, b_k\}$ in k^{th} stage to state $\{a_{k+1}, b_{k+1}\}$ in $k + 1^{th}$ stage, as evaluated in the paper is given by

$$B(a_{k+1}, b_{k+1}, a_k, b_k) = \left| r_p((k+1)T_s)y_{k+1}^a - \left(h_A T_s a_{k+1} + \tau h_B (b_k - b_{k+1}) + h_B T_s b_{k+1} \right) \right|^2 + \left| r_p(kT_s = \tau) - \left(h_B T_s b_k + \tau h_A (a_{k+1} - a_k) + h_A T_s a_k \right) \right|^2 \quad (2.8)$$

It is important to note that the metric minimized by the trellis search depends upon α , β and τ . Since, the fading gains for both the users and the relative timing offset is not known, the receiver iterates over certain discrete values of timing offset (τ) and gains (h_A and h_B) and evaluates the overall metric as a function of these parameters. It is important to note that the survivor sequence may be different for different values of these unknown parameters. Hence, the survivor sequence corresponding to the minimum branch metric is chosen for a particular set of $\{h_A, h_B, \tau\}$. As a result, the branch metric computed is a function of the timing offset and gains of both the users and needs to be minimized with respect to $\{h_A, h_B, \tau\}$ as well as the data sequences a_k and b_k . The minimum value of the metric corresponds to the estimate of the values and the corresponding survivor sequence is the sequence of $\{a_k, b_k\}$. It is worth noting here that the errors in decoding individual sequences $\{a_k\}$ and $\{b_k\}$ may be much higher. However, we are interested in the decoding of the modulo-2 sum of the two sequences i.e. $\{a_k \oplus b_k\}$.

The iteration over discrete values makes the algorithm computationally inefficient as the Viterbi algorithm, which already involves complex calculations, is repeated for

discrete values of phase and timing offset. Another drawback of this algorithm is that the discrete values chosen for phase and timing offset cause more quantization errors and may further worsen the decoding performance.

Both the research works mentioned above, however, assume that the channels are static over time. The work in [6] is based on the pre-knowledge of the fading gain of one of the users, and iterates the Viterbi algorithm for discrete values of the channel phase of the second user (assuming the amplitude to be unity).

3. PLNC with Non-Coherent Detection

Let us discuss one of the papers [10] whose authors have accounted for the fact that the channel state information is unknown at the relay. The algorithm presented in [10] is based on the decode and forward technique. They have assumed that the receiver does not know the phases and amplitudes of the channel gain. It is important to note that they do not attempt to estimate these channel parameters. Instead, they have proposed a non-coherent reception technique at the relay. The problem in this kind of method is that that the receiver suffers because of the losses inherent in non-coherent detection. However, the non-coherent detection presented will account for the fact that channel is varying as a symbol by symbol detector is proposed.

C. Goals and Motivation

As discussed in the previous sections, none of the research done in this area aims at estimation and tracking of the channel parameters, and simultaneously lift the assumption of synchronization of the signals of the users. The authors in [11] and [12] have done work in discovering the higher order modulation formats, suitable for the application of PLNC. Some of the research work like [13] involve the design of

channel coding to support PLNC.

With all the assumptions of previous work in mind, we propose a general framework which lifts all the above mentioned assumptions. Therefore the aim of our research work is:

- To enhance the parameter estimation and tracking in physical layer network coding.
- To provide a generic algorithm with a framework that can be easily extended for all modulation formats. Although we will be focusing on a simple modulation format (like BPSK), we will not exploit any special property inherent to the modulation format, and will allow the usage of other modulation formats by simple extensions to our framework.
- To lift the assumption of symbol boundary level synchronization and provide an algorithm to estimate the timing offsets between two signals of both users. We also aim to provide an algorithm whose estimation mechanism detects symbols in the presence of this timing offset. Since the relative timing offset between both users may be a fraction of a symbol duration, a symbol by symbol detector might not be efficient. Therefore, we will focus on the Maximum Likelihood Sequence Estimation of data.
- To provide an algorithm to compute the channel information before the decoding is started, so as to implement the detector in a coherent fashion and remove unnecessary losses inherent in non-coherent detection. This is particularly important as detection with modulation formats like QPSK, QAM etc. is highly dependent on the phase information. Hence, a non-coherent detector or a coherent detector with wrong phase information will result in huge losses when

dealing with these modulation formats.

- To account for a slowly varying channel and provide an algorithm which tracks the fading gains of both the channels, along with the estimation of bits. The assumption of constant channel is not valid when we are dealing with wireless networks. Also, phase information is very important for decoding higher order modulation formats like QPSK, QAM etc and hence, even small changes in the phase (and amplitude) must be tracked.

CHAPTER III

SYSTEM OVERVIEW, RECEIVER STRUCTURE AND ALGORITHMS

A. System Model

Suppose that the users, N_1 and N_3 , transmit their signals using a BPSK modulation format. Let us assume, for simplicity, that rectangular pulse shaping of symbol duration T_s is used. Let a_k and b_k be the bits sent by the two users, respectively. The signal received at the relay will then be of the form

$$y(t) = s_A(t - \tau_1) + s_B(t - \tau_2) + n(t),$$

where τ_1 and τ_2 are the time delays of each user and $s_A(t)$ and $s_B(t)$ are the contributions from each of the users, i.e.

$$s_A(t) = \sum_k h_A(t) a_k p(t - kT_s)$$

$$s_B(t) = \sum_k h_B(t) b_k p(t - kT_s),$$

where $h_A(t)$ and $h_B(t)$ are the complex Gaussian fading processes associated with the two transmitted signals, and $n(t)$ is Additive White Gaussian Noise (AWGN).

B. Receiver

In this section, we present the receiver structure which will be used to detect the modulo-2 sum of the two users data streams assuming that the various signal parameters are known. That is, for now we assume that h_A , h_B , τ_1 and τ_2 are known. The fading parameters and timing offsets are assumed constant throughout the length of the packet. Also, without loss of generality and for simplicity of notation, we assume that $\tau_1 = 0$ and $\tau_2 = \tau$. In the next sub-section, we will lift this assumption

and address how these parameter values can be found. The performance results and its comparison with a previous algorithm are also shown.

1. Algorithm

The received signal is passed through a matched filter which is equivalent to convolving the unknown signal with a conjugated time-reversed version of the pulse shape. In the case of square pulses, this is the same as convolving with $p(t)$. The signal output after matched filtering is given as

$$r_p(t) = \int_{-\infty}^{\infty} p(u)r(t-u)du.$$

Since the two signals are offset in time, a symbol-by-symbol detector will be sub-optimal. Rather, we employ a maximum likelihood sequence estimator (MLSE) which uses the Viterbi algorithm to search through an appropriate trellis diagram.

In order to develop the form of the trellis search, the following metric needs to be minimized:

$$\begin{aligned} \gamma &= \int \left| y(t) - s_a(t) - s_b(t - \tau) \right|^2 dt \\ &= \int \left(|y(t)|^2 + |s_a(t)|^2 + |s_b(t - \tau)|^2 - 2Re\{y(t)s_a^*(t) + y(t)s_b^*(t - \tau) - s_a(t)s_b^*(t - \tau)\} \right) dt \\ &= \int \left(|y(t)|^2 + |s_a(t)|^2 + |s_b(t - \tau)|^2 \right) dt - 2Re\left\{ \sum_k a_k^* h_A^* \underbrace{\int (y(t)p(t - kT_s) dt)}_{r_p(kT_s)} \right\} \\ &\quad - 2Re\left\{ \sum_k h_B^* b_k^* \underbrace{\int (y(t)p(t - kT_s - \tau)^* dt)}_{r_p(kT_s + \tau)} \right\} \\ &\quad + 2Re\left\{ \sum_k \sum_l h_A^* h_B a_k^* b_l \int (p(t - kT_s)p(t - lT_s - \tau)) dt \right\} \end{aligned}$$

The first term is constant for all k . Hence, the constant terms can be removed

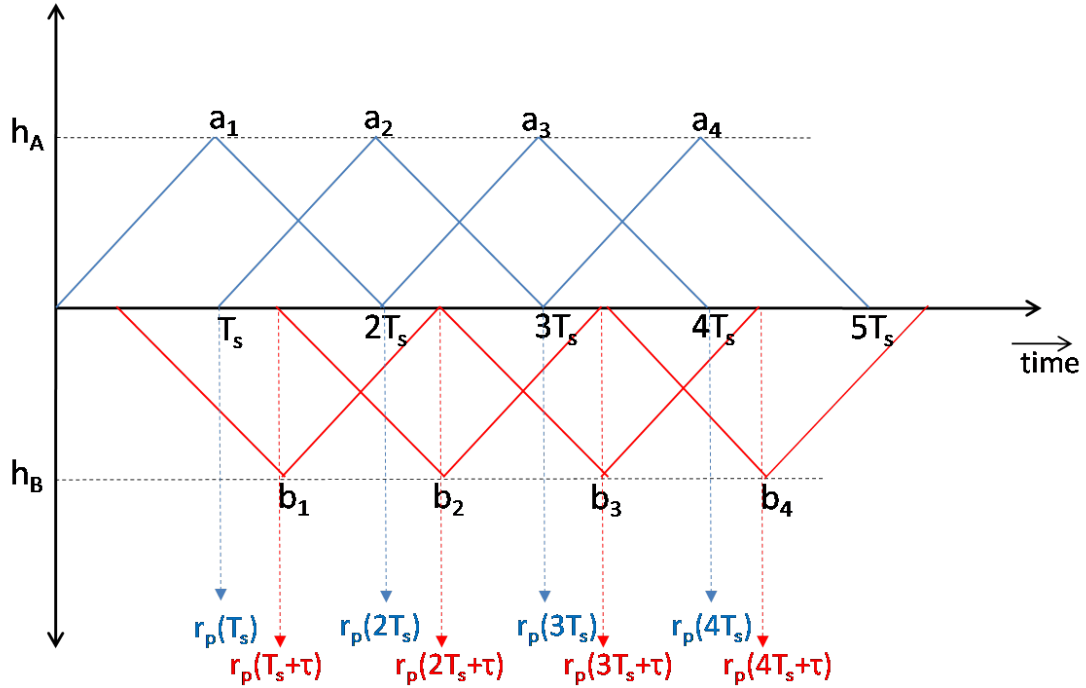


Fig. 10.: Matched filter output for $a_k = +1$ and $b_k = -1$

from the metric to be minimized. Now, the second and third terms can be calculated easily from the matched filter output as shown in Figure 10.

In order to simplify the fourth term, we look at the overlap of pulses $p(t - kT_s)$ and $p(t - lT_s - \tau)$. This is shown in Figure 11. So, the fourth term reduces to:

$$2\text{Re}\left\{\sum_k h_A^* h_B a_k^* (b_k(T_s - \tau) + b_{k-1}\tau)\right\}$$

Hence, the overall metric to minimize reduces to:

$$\gamma = \sum_k \text{Re}\left\{-a_k^* h_A^* r_p(kT_s) - h_B^* b_k^* r_p(kT_s + \tau) + h_A^* h_B a_k^* (b_k(T_s - \tau) + b_{k-1}\tau)\right\}. \quad (3.1)$$

The problem of minimizing this metric can be seen as a Maximum Likelihood

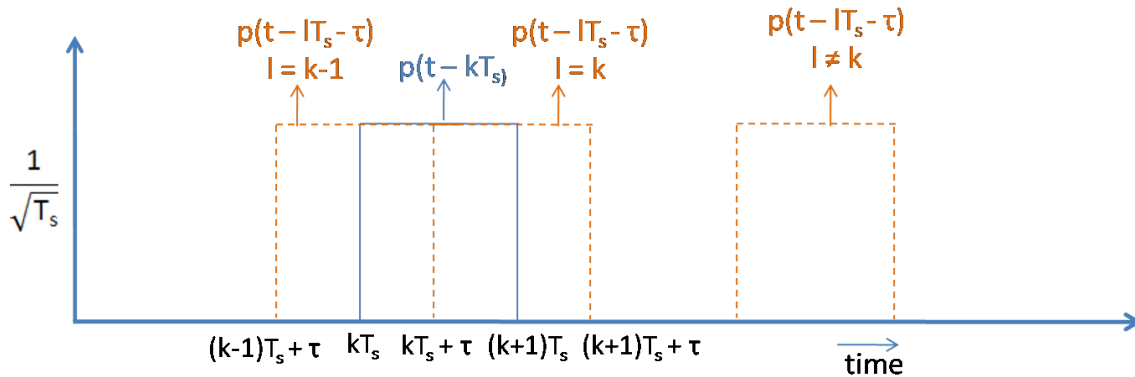


Fig. 11.: Overlap of pulses $p(t - kT_s)$ and $p(t - lT_s - \tau)$

Sequence Detection. In other words, we need to find the sequences a_k and b_k that minimize the overall metric γ . If the length of the sequences is N bits, a brute force search for this minimization problem would result in $2^N \times 2^N$ comparisons, making the search procedure not feasible.

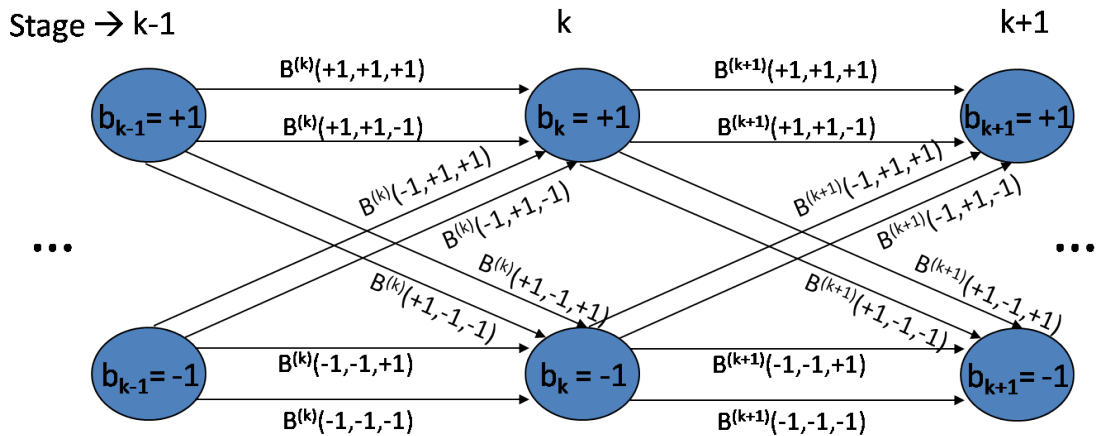


Fig. 12.: Trellis used in MLSE receiver

This problem can also be viewed as a trellis search as shown in Figure 12, where the

branch metric is :

$$B^{(k)}(b_{k-1}, b_k, a_k) = \text{Re} \left\{ -a_k^* h_A^* r_p(kT_s) - h_B^* b_k^* r_p(kT_s + \tau) + h_A^* h_B a_k^* (b_k(T_s - \tau) + b_{k-1}\tau) \right\} \quad (3.2)$$

Each symbol can be seen as a stage of the trellis. The trellis for this problem involves only two states ($b_k = \pm 1$), as shown in the figure. It is worth noting that there is a hidden state (a_k) involved in this trellis, resulting in two more transitions between the states of consecutive stages. Thus, each state is involved with a total of 4 transitions.

A feasible solution to this minimization problem is given by the Viterbi Algorithm. At the k^{th} stage, all transitions that would lead to a particular state are analyzed and the one with least cumulative cost (or the cumulative branch metric i.e. $\sum B^{(k)}(b_{k-1}, b_k, a_k)$) until k^{th} stage is chosen as the "survivor" for that state. The idea of the Viterbi Algorithm is that only the survivor branches to a particular state can be extended, in order to find the sequence (or path) with minimum cost. After finding the survivors for each state at the k^{th} stage, the same procedure is repeated for the $k + 1^{\text{th}}$ stage and so on, until the end of the trellis is reached. At the end of the trellis, the sequence with the minimum overall cost is chosen and is traced back to recover the whole sequence. It is important to note here that in this problem, we decode the sequences a_k and b_k , but it is highly likely that the decoding results in interchanging of some of the bits of these sequences. As discussed above, we only need to decode the modulo-2 sum of the bits, and therefore, we need to decode $\{a_k \oplus b_k\}$. Therefore, we form the modulo-2 sum of the data sequences.

A block diagram of the receiver structure is shown in Figure 13. The received signal is match filtered and sampled twice per symbol interval to produce the sufficient statistics $r_p(kT_s)$ and $r_p(kT_s + \tau)$. These statistics are then used to compute the branch

metrics of the trellis diagram and the optimal path through the trellis diagram is computed.

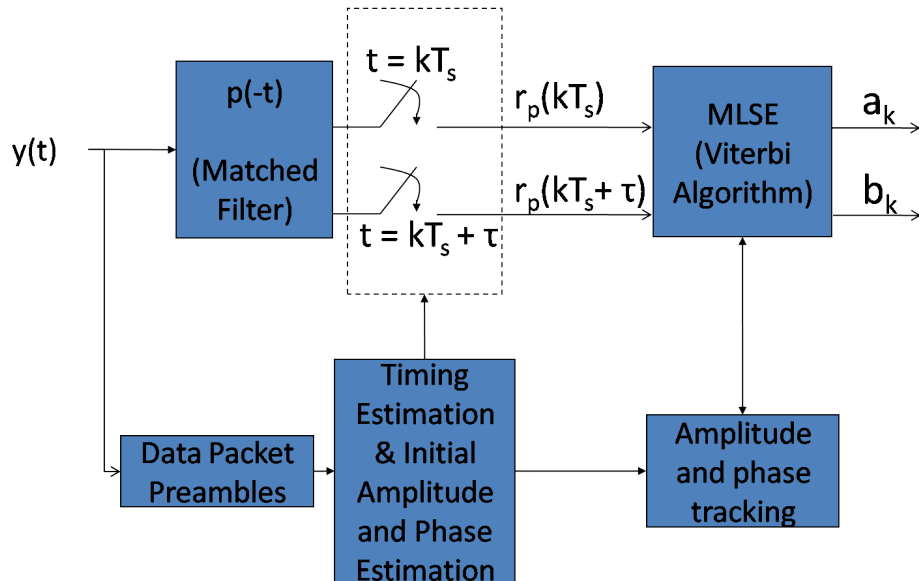


Fig. 13.: Receiver structure

2. Simulation Results

The fading channel gains are treated as constant throughout the length of a packet and are fixed to $h_A=1$ and $h_B=1$ in these simulation results. In the subsequent sections, we will lift this assumption and also provide an algorithm for their initial estimation. Figure 14 shows the detection performance of the proposed algorithm and a comparison to an algorithm suggested in [6]. It is worth noting that we are aiming at decoding only the modulo-2 sum of the bit sequences, and hence, the probability of error corresponds to the modulo-2 sum, and not the individual sequences. While the performance of our detector is slightly better it should also be pointed out that our receiver is also significantly less complex as the detector in [6]

required a search through a 16-state trellis while ours requires only two states. Note also from Figures 14 and 15 that the performance of the receiver is worst when the two signals are misaligned by exactly one half of a bit interval. While this slight degradation due to misalignment is unavoidable, the sensitivity is not nearly as great as reported in [6]. Figure 16 shows the robustness of the algorithm to the relative phase difference of the two users.

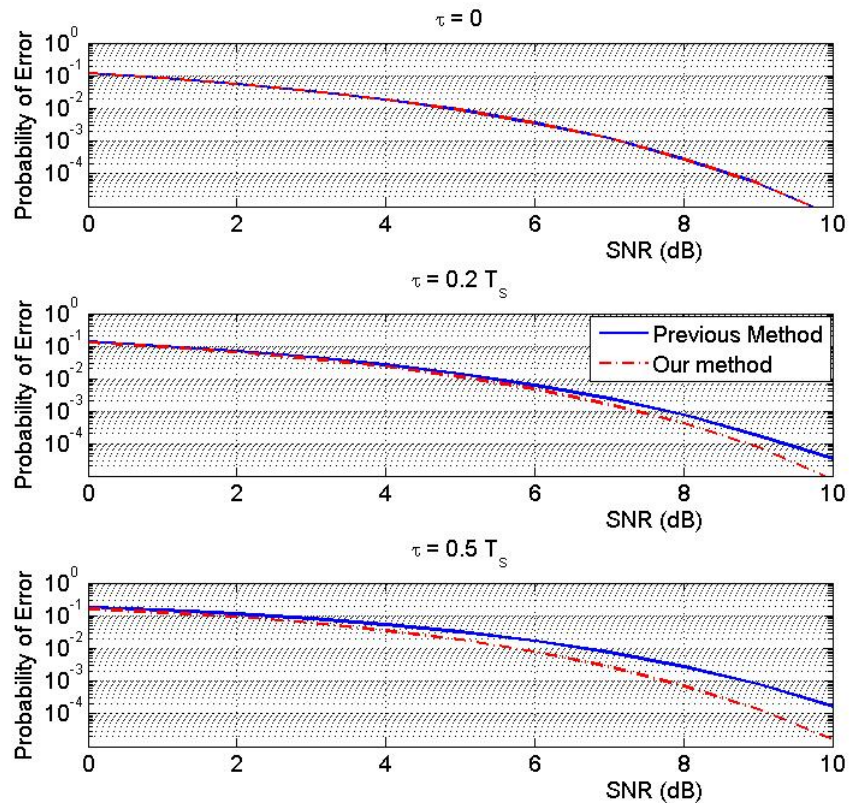


Fig. 14.: MLSE performance in terms of probability of error vs SNR (in dB) for different values of τ for $h_A = h_B = 1$ (i.e. phases of two signals align with each other)

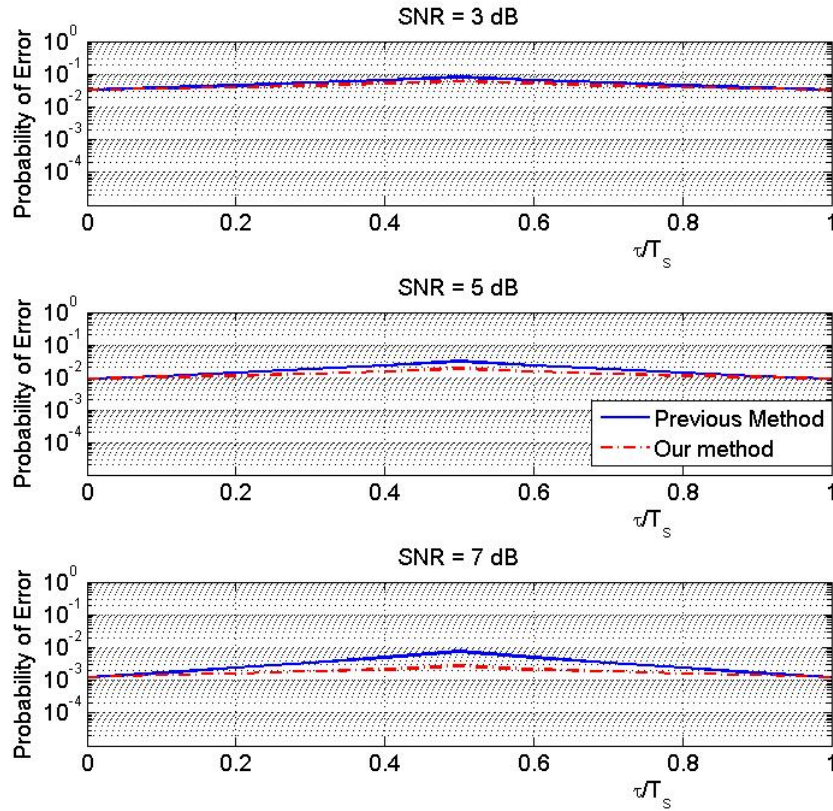


Fig. 15.: MLSE performance in terms of probability of error vs τ for different values of SNR for $h_A = h_B = 1$ (i.e. phases of two signals align with each other).

C. Joint Estimation of Amplitude, Phase and Timing

In the previous section, we assumed that the fading gains and timing offsets were constant throughout the transmission time and were known beforehand. In this section, we investigate how to estimate these parameters. As illustrated in Figure 13, initial estimates of the signal timings and fading parameters are found using the preambles of the data packets. Thus, during the task of estimating signal parameters, the data bits are taken to be known.

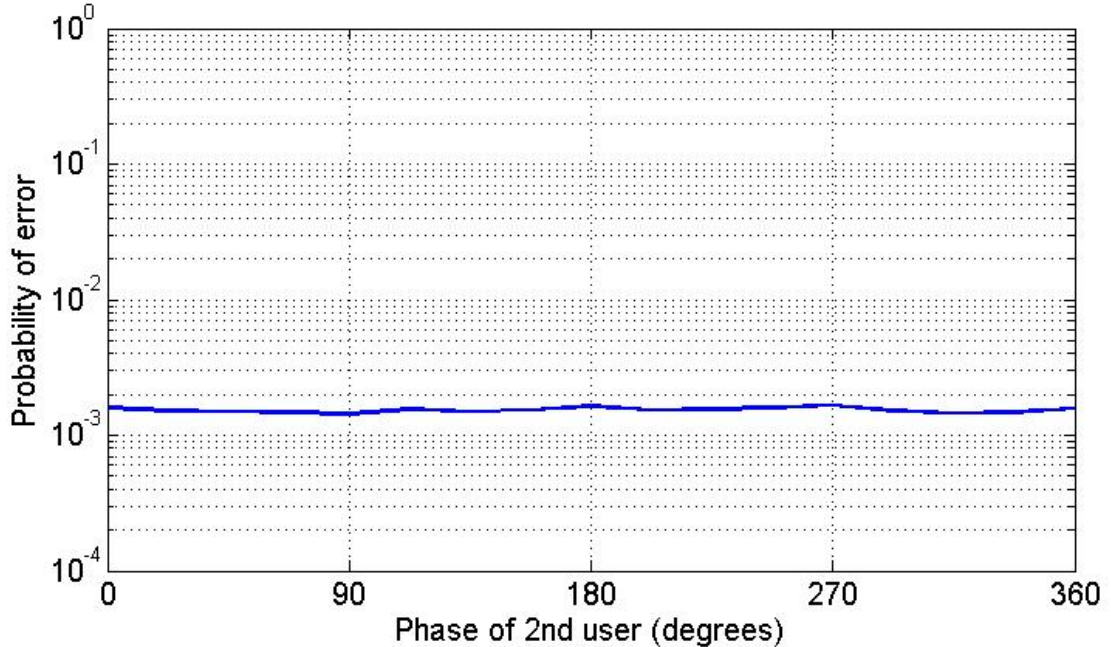


Fig. 16.: MLSE performance in terms of probability of error vs phase of second user for SNR = 7 dB, $\tau = 0.2T_s$, $h_A = 1$ and $|h_B| = 1$

1. Algorithm

We first sample the match filter output at an arbitrary time offset and a frequency of once per symbol interval to produce the samples

$$y_k = \frac{h_A}{T_s}(a_k T_s + \tau_1(a_{k-1} - a_k)) + \frac{h_B}{T_s}(b_k T_s + \tau_2(b_{k-1} - b_k)) + n_k, \quad (3.3)$$

This is also shown in Figure 17. We look at the problem of estimating the complex fading parameters, h_A and h_B , assuming the timings, τ_1 and τ_2 are known. Towards that end and in order to simplify notation, define $X_k = a_k + \frac{\tau_1}{T_s}(a_{k-1} - a_k)$ and $Y_k = b_k + \frac{\tau_2}{T_s}(b_{k-1} - b_k)$ in which case Equation (3.3) reduces to

$$y_k = h_A X_k + h_B Y_k + n_k \quad (3.4)$$

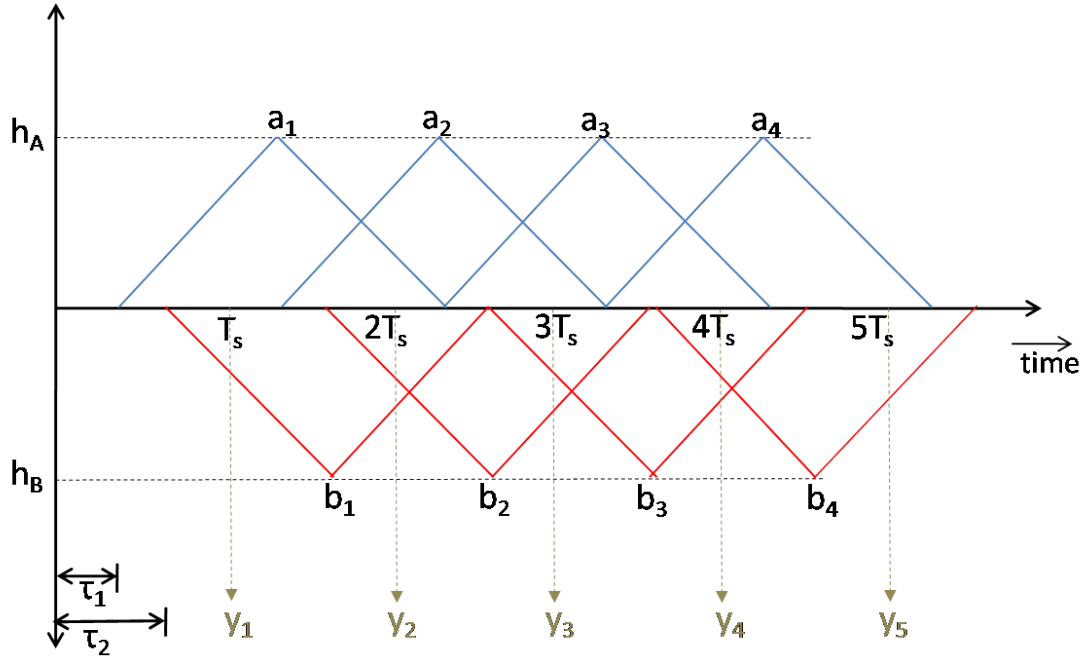


Fig. 17.: Unsynchronized matched filter output for $a_k = +1$ and $b_k = -1$

Our estimate of the fading parameters will be the values of h_A and h_B which minimize

$$\gamma = \sum_k \left\| y_k - h_A X_k - h_B Y_k \right\|^2. \quad (3.5)$$

Differentiating with respect to the real and imaginary parts of h_A and h_B and setting the derivative equal to zero results in a set of two complex Equations (four real Equations) that are linear in h_A and h_B

$$h_A \sum_k X_k^2 + h_B \sum_k X_k Y_k = \sum_k X_k y_k \quad (3.6)$$

$$h_A \sum_k X_k Y_k + h_B \sum_k Y_k^2 = \sum_k Y_k y_k \quad (3.7)$$

The fading parameter estimates are then found by solving this set of linear Equations for h_A and h_B . The solution is given by the following matrix Equation:

$$\begin{bmatrix} h_A \\ h_B \end{bmatrix} = \begin{bmatrix} \sum_k X_k^2 & \sum_k X_k Y_k \\ \sum_k X_k Y_k & \sum_k Y_k^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_k X_k y_k \\ \sum_k Y_k y_k \end{bmatrix}$$

Estimates for the timing offsets τ_1 and τ_2 given that the fading parameters h_A and h_B are known can be found in a similar manner. Define $z_k = y_k - h_A a_k - h_B b_k$, $U_k = \frac{h_A}{T_s}(a_{k-1} - a_k)$ and $V_k = \frac{h_B}{T_s}(b_{k-1} - b_k)$. Then Equation (3.3) can be rewritten as

$$\gamma = \sum_k \left\| z_k - \tau_1 U_k - \tau_2 V_k \right\|^2. \quad (3.8)$$

The values of τ_1 and τ_2 which minimize this expression are the solutions to the linear Equations

$$\tau_1 \sum_k |U_k|^2 + \tau_2 \sum_k \text{Re}[U_k^* V_k] = \sum_k \text{Re}[U_k^* z_k] \quad (3.9)$$

$$\tau_1 \sum_k \text{Re}[U_k V_k^*] + \tau_2 \sum_k |V_k|^2 = \sum_k \text{Re}[V_k^* z_k]. \quad (3.10)$$

The timing estimates can thus be found by solving the following matrix Equation:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \sum_k |U_k|^2 & \sum_k \text{Re}[U_k^* V_k] \\ \sum_k \text{Re}[U_k V_k^*] & \sum_k \sum_k |V_k|^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_k \text{Re}[U_k^* z_k] \\ \sum_k \text{Re}[V_k^* z_k] \end{bmatrix}$$

At this point we have seen that it is straightforward to estimate h_A and h_B if τ_1 and τ_2 are known and requires only the solution of a set of two linear Equations. Likewise, estimating τ_1 and τ_2 when h_A and h_B are known also only requires the solution of two linear Equations. However, if all four quantities are unknown, the resulting set of four Equations ((3.6), (3.7), (3.9) and (3.10)) are not linear in h_A , h_B , τ_1 and τ_2 , and hence a closed form expression is difficult to obtain. We propose the following iterative algorithm:

- Take an initial guess of τ_1 and τ_2 .

- Estimate h_A and h_B using τ_1 and τ_2 found in Step 1 and Equations (3.6) and (3.7).
- Estimate τ_1 and τ_2 using h_A and h_B determined in Step 2 and Equations (3.9) and (3.10).
- Iterate Step 2 and 3 until the solution converges to a particular set of values. This usually takes 3-5 iterations to converge.

The point of convergence of this iterative procedure depends upon the initial guess chosen. Sometimes, because of a poorly chosen initial guess the solution may not converge or converge to a wrong solution. Hence, we refine the above mentioned method. Take n initial guesses and find the solution corresponding to each guess and then choose the one which has the minimum value of γ , given by (3.5).

2. Simulation Results

Figure 18 shows the effectiveness of the timing and fading estimation algorithm for different values of n . The timing offsets and phase of the channels were randomly chosen with a uniform distribution. The amplitudes of both the channels were fixed to unity.

It was observed that $n = 3$ improves the performance of the algorithm drastically over the performance with $n = 1$ i.e. with single guess. This justifies our assumption that the wrong initial guess may lead us to a wrong solution. However, changing from $n = 3$ to $n = 5$ did not improve the performance much further. Hence, $n = 3$ seems to have the optimum trade-off between performance and complexity of the receiver.

In addition, we also evaluated the number of bits in the pilot sequence required to estimate the channel parameters and timing offsets. Figure 19 shows the variation

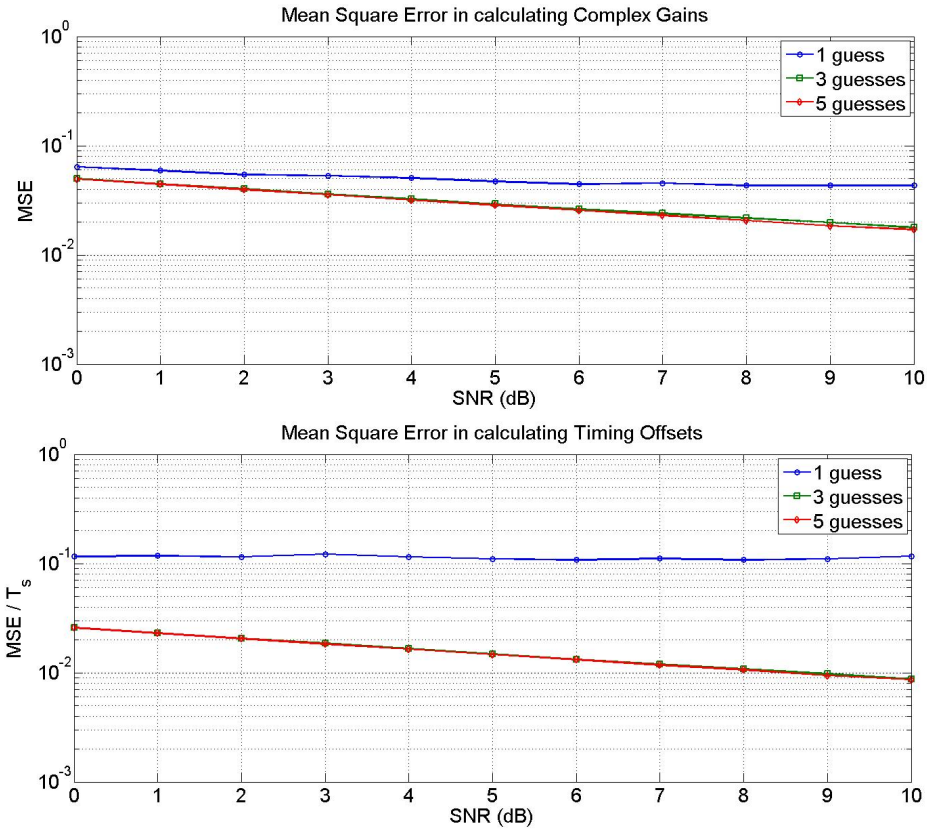


Fig. 18.: MSE in calculating complex fade values for different values of n (for pilot sequence of length 50 bits)

of performance of this algorithm with respect to the length of pilot sequence used. After a detailed exercise, we chose use a pilot sequence of length 50 bits.

D. Amplitude and Phase Tracking for a Slowly Varying Channel

As discussed previously, it may not be reasonable to assume that the fading encountered is constant for the entire duration of a data packet. As a result, we must make an effort to track the independently time-varying fading processes experienced by the two users. Since our receiver already employs an MLSE, we propose to use a

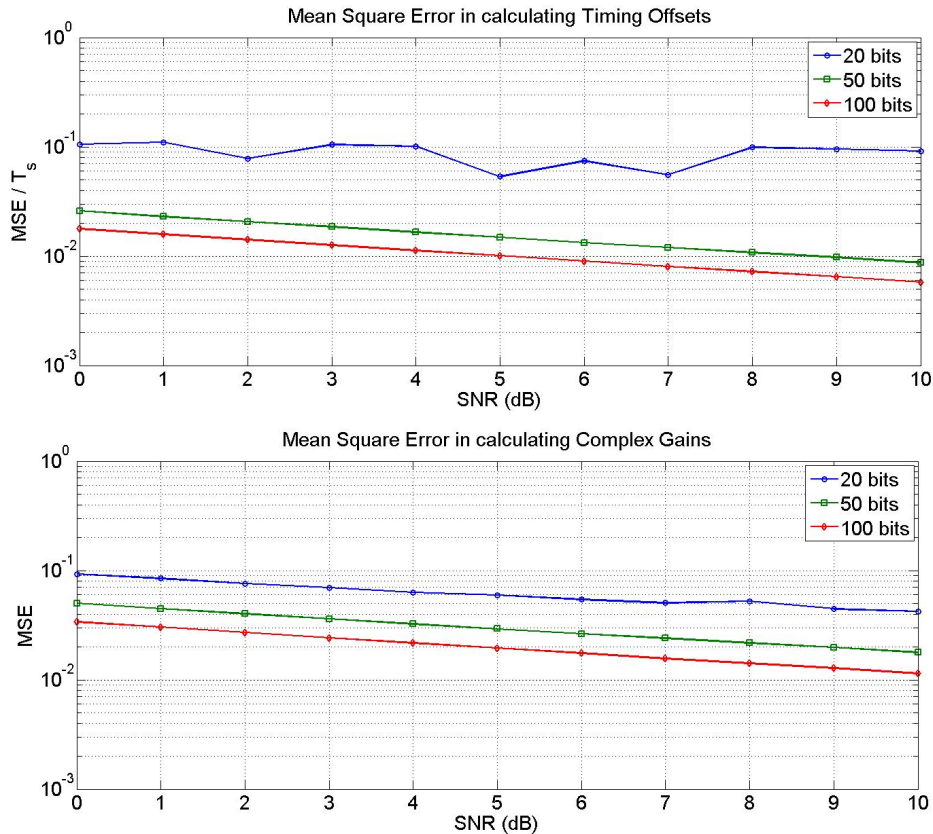


Fig. 19.: MSE in calculating timing offsets for different lengths of pilot sequence ($n = 3$)

per-survivor processing (PSP) approach [14] to estimate the fading processes which will work in conjunction with the Viterbi algorithm used for detecting the data bits.

1. Tracking Algorithm for Single User Case

The authors in the paper [14] have used PSP algorithm to track a time varying linear channel introducing Inter-Symbol Interference (ISI) to the signal. In this section, a brief overview of idea used in the PSP algorithm for a single user case is presented.

Suppose $\{a_n\}_{n=0}^k$ are the symbols transmitted and channel is not ideal. The channel introduces ISI into the signal at the symbol level, which is given by $\{f_n\}_{n=1}^L$ such that the symbol level output of the channel is $y_k = \sum_{n=0}^L f_n a_{k-n}$. Now given the observation sequence $\{y_n\}_{n=0}^k$, the joint MS estimate of data sequence and channel impulse response is obtained by maximizing the likelihood function over $\{a_n\}_{n=0}^k$ and $\{f_n\}_{n=0}^L$ i.e.

$$\max_{\{a_n\}_{n=0}^k} \max_{\{f_n\}_{n=0}^L} \left(\sum_{n=0}^k \left| y_n - \sum_{i=0}^L f_i a_{n-i} \right|^2 \right) \quad (3.11)$$

The authors in [14] have proposed an approximation to the same maximization problem based on PSP, in which the channel impulse response can be derived by maximizing over $\{f_n\}_{n=0}^L$ for each survivor sequence. The survivor metrics represent the value of the likelihood function for the associated survivor sequences, in which the maximization over the channel parameters has already been performed. Associated with each survivor sequence, we have now an estimate of the channel impulse response. Consequently, the survivor sequence with the largest metric corresponds to the desired joint estimate of the channel parameters and data sequences.

The maximization of the likelihood function with respect to the channel vector for a given survivor sequence may be performed by a Recursive Least Square (RLS) algorithm. Hence, the algorithm works in conjunction with the Viterbi algorithm. After each step of the Viterbi algorithm, the channel impulse response is updated for all the transitions that extend the survivors. This is just a brief overview of the idea and the details of this implementation are shown in [14].

2. Tracking Algorithm for Double User Case

In this section, we use the general framework presented in [14] and extend it to the double user case. In the treatment that follows, we ignore the correlation between the

noise components contributing to y_{ak} and y_{bk} respectively. However, results show that the performance of our tracking algorithm is good even while ignoring this correlation. Using a similar framework as proposed in [14], we denote the error at the k th step of the Viterbi algorithm as:

$$e_{1k}(s_k, s_{k+1}) = r_p(kT_s) - [h_A(k) \ h_B(k) \ h_B(k-1)] \cdot [a_k T_s \ b_k(T_s - \tau) \ b_{k-1}\tau]^T \quad (3.12)$$

The fading gains (and timing offsets) are estimated using a pilot sequence at the beginning of the data packet. The Viterbi algorithm is initiated with the knowledge of these gains for the first symbol. One step of the Viterbi algorithm is performed by maximizing over the possible data sequences as explained in the previous section. For each transition $s_k \rightarrow s_{k+1}$ that extends the survivor sequence, we find the error $e_{k1}(s_k, s_{k+1})$ given by Equation (3.12). The problem of updating the channel gains for a given survivor sequence can be solved using Recursive Least Squares (RLS) algorithm. Using the error corresponding to each survivor, the Kalman gain vectors, correlation matrices, and channel gains are updated according to:

$$\begin{aligned} k_{k+1} &= \frac{P_k(s_k) a_k(s_k, s_{k+1})}{w + a_k(s_k, s_{k+1})^T P_k(s_k) a_k(s_k, s_{k+1})^*} \\ P_{k+1} &= \frac{1}{w} [P_k(s_k) - k_{k+1} a_k(s_k, s_{k+1})^T P_k(s_k)] \\ h_A(k+1) &= h_A(k) + [k_{k+1}(s_{k+1}) e_{1k}(s_k, s_{k+1})]_1 \end{aligned} \quad (3.13)$$

where $[\]_1$ denotes the first element of a column vector and w is the weighing factor to limit the memory of algorithm in order to track the slowly varying signals. It is worth noting that Equation (3.13) gives a 3-element column vector. Each term of this vector updates the corresponding element in $([h_A(k) \ h_B(k) \ h_B(k-1)]^T)$.

However, we use only the first element of this vector which corresponds to h_A ,

the complex fading gain for the channel corresponding to the first user. Similarly, we can track the fading gain of second user by finding similar expressions and using the matched filter output at $t = kT_s + \tau$.

3. Implementation and Results

The above mentioned algorithm was implemented and simulations were run to test the performance. The channel fade values were generated from the Rayleigh fading model with a bathtub type power spectral density with a normalized Doppler rate of 0.001 (as explained in Chapter II Section A3). The relative timing offset was set to $0.2T_s$ for the following results.

The tracking algorithm explained above works well but the estimation is somewhat noisy. The performance of this tracking algorithm can be further improved by filtering the estimated fading gains with a low pass filter. It was observed that a single-pole IIR filter gives a very good trade off between complexity and performance. Hence, we used an IIR filter with one pole to filter the fading gains. The filter coefficient can be adjusted according to the frequency of variation of the channel (i.e. the fading Doppler frequency). Since we are concerned here with tracking of slowly varying signals (with normalized Doppler rates around 0.001), the filter coefficient should be chosen to be very small. After running some simulations with different filter coefficient, we chose to use a filter coefficient of 0.05. The need for filtering is demonstrated in Figure 20, which shows that the filtered estimates are much less noisier than the unfiltered ones.

The tracking algorithm can be summarized as follows:

- Step1: Initial fading gains and timing offsets are estimated using the algorithm explained in Section C. It is assumed that the channel is slowly varying and the

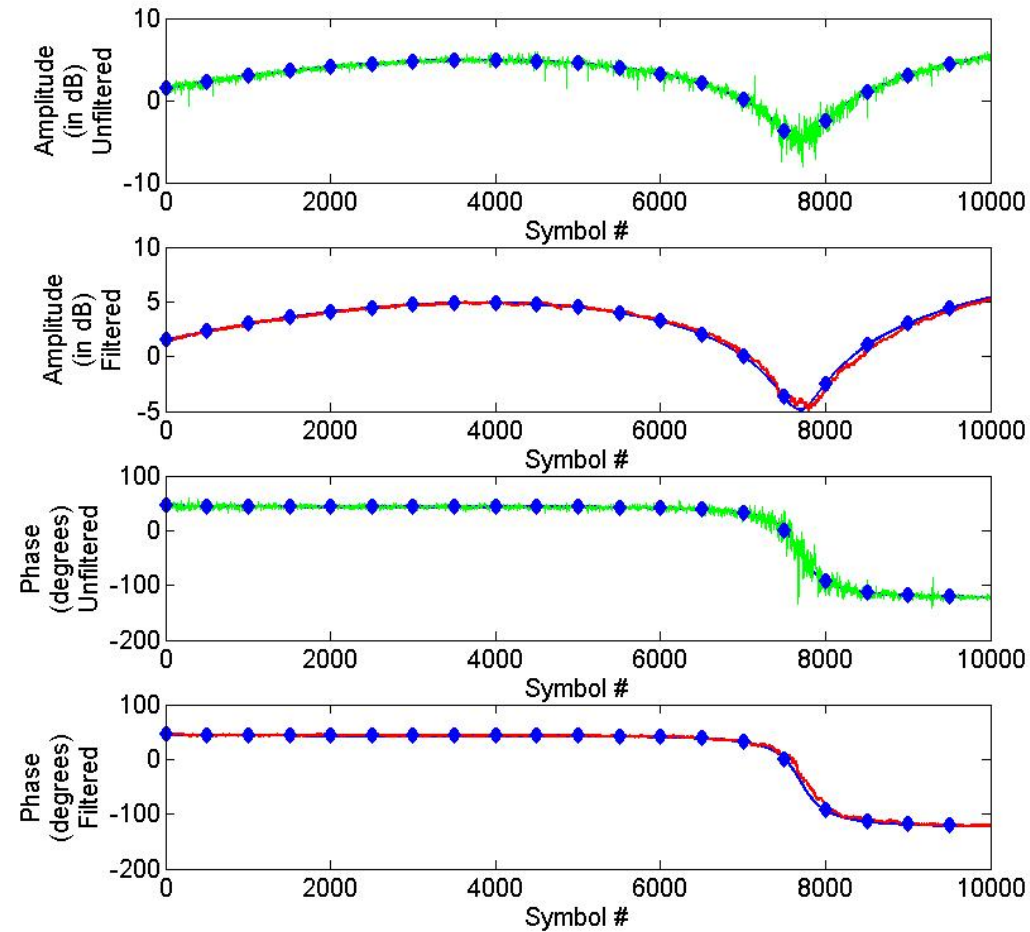


Fig. 20.: Impact of filtering

fading gains are relatively constant throughout the preamble or known symbol sequence.

- Step2: MLSE decoder is initialized with these fading values in the beginning and error metric given by (3.12) is calculated. The error metric is then used to update the Kalman gain vectors, correlation matrices and fading gains using Equations (3.13).
- Step3: The fading gains are filtered using an one pole IIR filter whose coefficient

is set to 0.05 (for very slowly varying channel).

- Step4: Repeat from Step 2 with the next stage of trellis and using the updated fading gains.

This tracking algorithm was applied to the receiver architecture and the results were encouraging. However, as with most attempts to track fading in a decision directed manner, the algorithm had a tendency to lose lock on the signal during deep fades. This is illustrated in Figure 21 where a typical realization of the tracking process is shown. When the signal enters a deep fade, the received signal-to-noise ratio (SNR) momentarily falls very low resulting in a burst of often incorrect estimated data bits. Since our estimates of the fading process rely on the detected data bits, these estimates become corrupted and we lose track of the fading process. When the signal emerges from the deep fade, the system will recover and resume reliable detection of the data bits as well as reasonable tracking of the fading process, but it is common for the algorithm to incur a cycle slip and emerge from the fade locked out of phase on the fading process by 180 degrees. From that point on, the polarity of the detected data bits will also be inverted resulting in most detected bits being in error for the remainder of the packet (or until another fade event occurs). This problem can easily be fixed through the use of differential encoding and decoding of the data bits. It should be emphasized that we are not using differential detection (DPSK) here but rather coherent detection of differentially encoded BPSK.

The only drawback of using differential encoding is that it slightly increases the decoded bit error rate as random errors tend to occur in pairs when differential encoding is used. However, this does provide us a means of robustness to the phase slips described above.

The performance of the algorithm, with and without differential encoding is

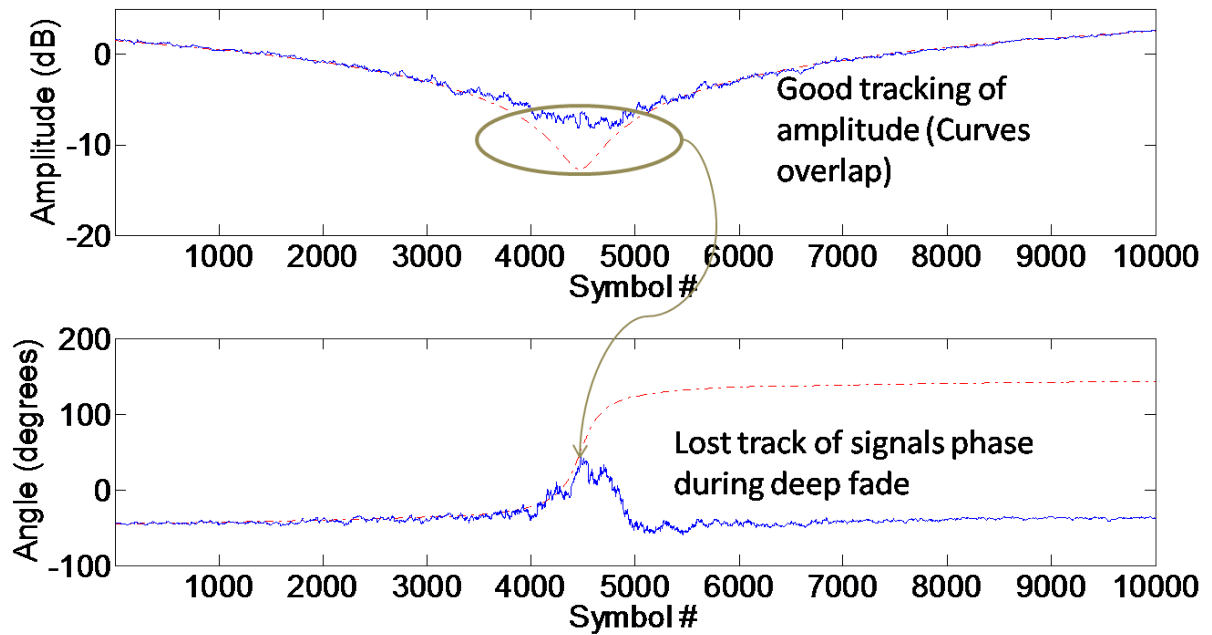


Fig. 21.: Amplitude and phase tracking

shown in Figure 22.

From the Figure, we can make the following observations:

- An error floor is observed using this algorithm without any differential encoding. This error floor is related to the probability of a deep fade causing a loss of tracking before the end of a packet.
- Ideal tracking with differential encoding suffers some loss as compared to ideal tracking without differential encoding. This is due to the fact that if we make one error in decoding a bit in differential encoding, we essentially make two errors in the original data bits. However, the use of differential encoding is quite efficient in removing the error floor caused by loss of lock on the fading process.

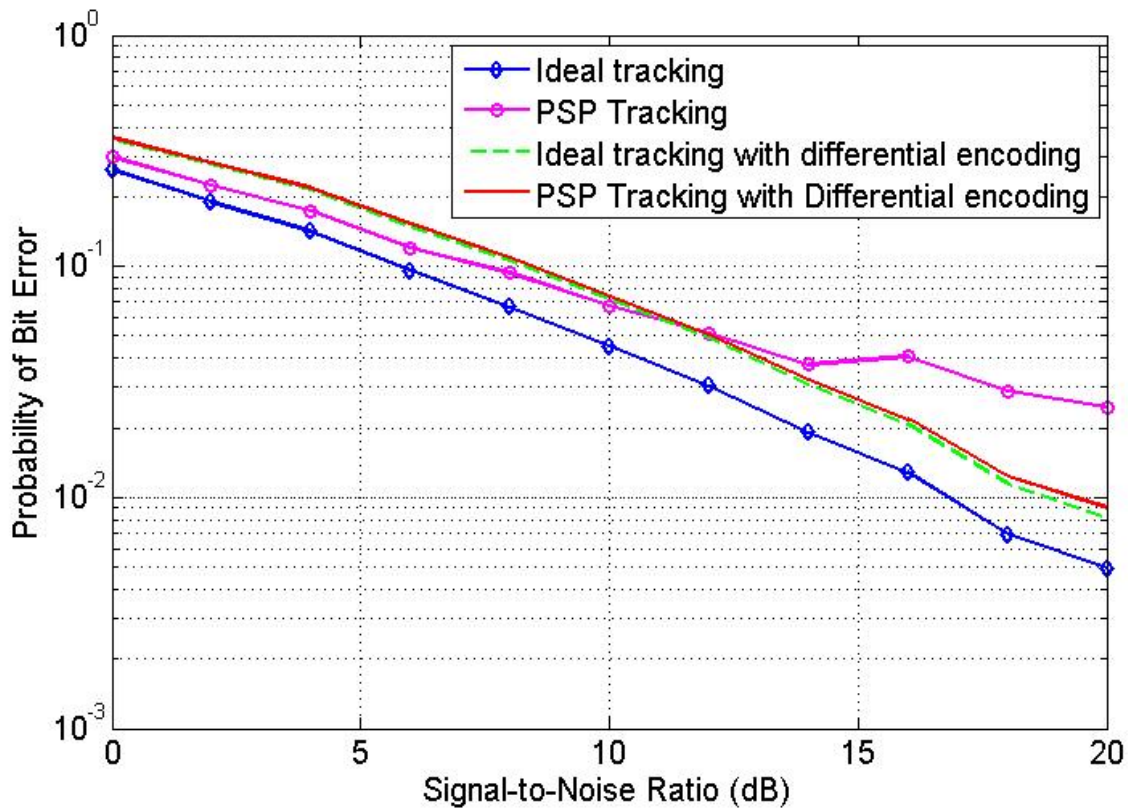


Fig. 22.: Performance of tracking algorithm

E. Receiver Diversity

In wireless communications, antenna diversity is very useful in terms of partially recovering the losses due to fading. The algorithm and the framework we proposed allows us to easily handle the case where antenna diversity is used at the relay. If the relay has two antennas, then the framework can accommodate the receiver diversity without adding a lot of complexity to our MLSE decoder. Since, the two noise samples at different receiver branches are uncorrelated, the metric can be modified as

$$\gamma = \int (|y_1(t) - s_{A1}(t) - s_{B1}(t - \tau)|^2 + |y_2(t) - s_{A2}(t) - s_{B2}(t - \tau)|^2) dt$$

$$\begin{aligned}
&= \int (|y_1(t)|^2 + |s_{A1}(t)|^2 + |s_{B1}(t - \tau)|^2 + |y_2(t)|^2 + |s_{A2}(t)|^2 + |s_{B2}(t - \tau)|^2) dt + \\
&\quad \sum_k \operatorname{Re} \left\{ -a_k^* h_{A1}^* r_{p1}(kT_s) - h_{B1}^* b_k^* r_{p1}(kT_s + \tau) + h_{A1}^* h_{B1} a_k^* (b_k(T - \tau) + b_{k-1}\tau) \right. \\
&\quad \left. - a_k^* h_{A2}^* r_{p2}(kT_s) - h_{B2}^* b_k^* r_{p2}(kT_s + \tau) + h_{A2}^* h_{B2} a_k^* (b_k(T - \tau) + b_{k-1}\tau) \right\}.
\end{aligned}$$

After removing constant terms as done for the non-diversity case, the metric reduces to:

$$\begin{aligned}
\gamma = &\sum_k \operatorname{Re} \left\{ -a_k^* h_{A1}^* r_{p1}(kT_s) - h_{B1}^* b_k^* r_{p1}(kT_s + \tau) + h_{A1}^* h_{B1} a_k^* (b_k(T - \tau) + b_{k-1}\tau) \right\} \\
&+ \sum_k \operatorname{Re} \left\{ -a_k^* h_{A2}^* r_{p2}(kT_s) - h_{B2}^* b_k^* r_{p2}(kT_s + \tau) + h_{A2}^* h_{B2} a_k^* (b_k(T - \tau) + b_{k-1}\tau) \right\}.
\end{aligned}$$

where the subscripts 1 and 2 indicate the two different receiver branches. Note that we assume here that the timing offsets are the same for both branches. This is a reasonable assumption for antennas which may be physically separated from each other by the distances on the order of a few feet. However, it is very straightforward to extend the problem to the case of two different timing offsets at the receiver (by changing τ to τ_2 in metric corresponding to second user. In that case, the above metric can be written as:

$$\begin{aligned}
\gamma = &\sum_k \operatorname{Re} \left\{ -a_k^* h_{A1}^* r_{p1}(kT_s) - h_{B1}^* b_k^* r_{p1}(kT_s + \tau_1) + h_{A1}^* h_{B1} a_k^* (b_k(T - \tau_1) + b_{k-1}\tau_1) \right\} \\
&+ \sum_k \operatorname{Re} \left\{ -a_k^* h_{A2}^* r_{p2}(kT_s) - h_{B2}^* b_k^* r_{p2}(kT_s + \tau_2) + h_{A2}^* h_{B2} a_k^* (b_k(T - \tau_2) + b_{k-1}\tau_2) \right\}.
\end{aligned}$$

However, this involves sampling output of the matched filter for the diversity receiver at time instants different from that of the primary antenna. It should be noted that the the channel parameters for the diversity antenna i.e. amplitude, phase and timing offsets, must be estimated and tracked, along with the primary antenna parameters. It is again straightforward to extend the PSP tracking algorithm explained

in the previous subsection to track the fading gains for the second antenna as well.

Figure 23 shows the improvement in performance encountered through the use of receiver diversity.

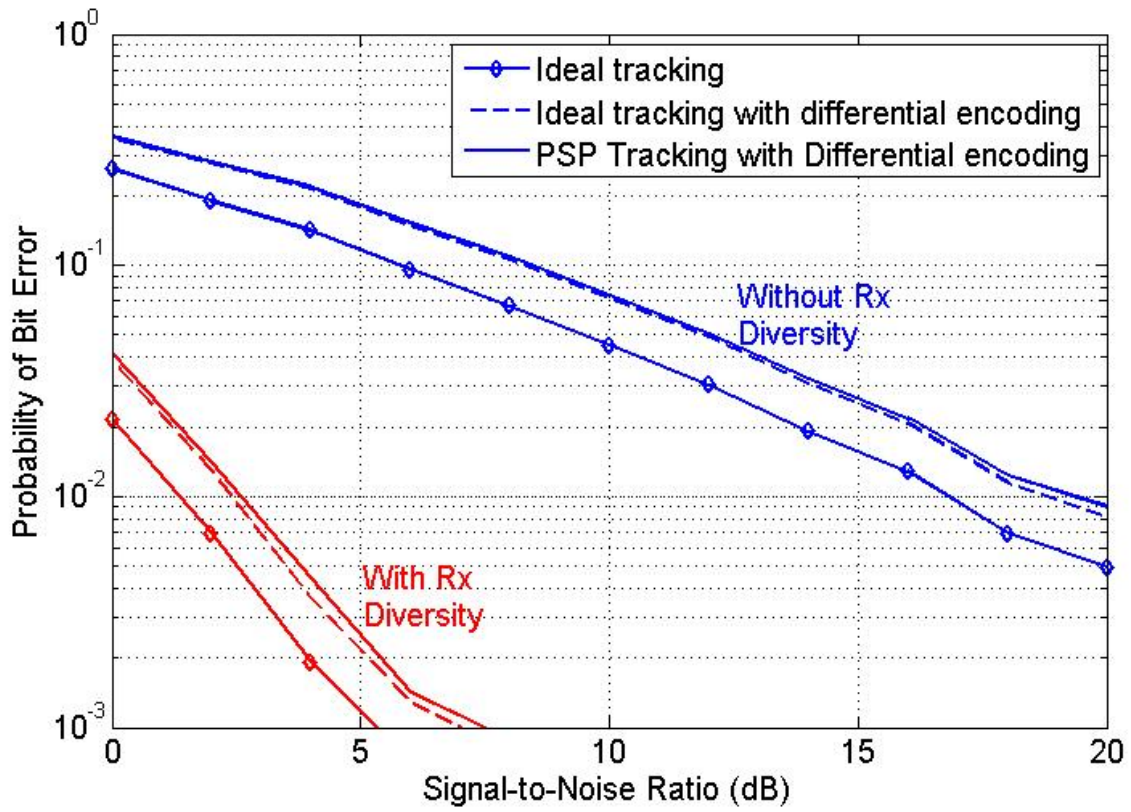


Fig. 23.: Performance of receiver with diversity

CHAPTER IV

CONCLUSION AND FUTURE WORK

In this thesis, a receiver architecture is presented which can be used for detection at the relay node in a physical layer network coding system. The receiver is practical in the sense that it is of manageable complexity and operates under reasonable assumptions about the channel. In particular, there are no assumptions made about prior information regarding various channel parameters, nor do we assume that the two signals incident on the relay node are perfectly synchronized. Rather, we present algorithms for estimating the timing of each users signals and for tracking the fading processes encountered by each signal. We present performance results for both the case of single antenna reception and for the case when the receiver employs antenna diversity. Results are quite encouraging, and they demonstrate that the performance of receiver is fairly close to the case when channel parameters are perfectly known.

There are several areas in which this thesis work can be extended. Some of them are :

- Extend the framework to allow for more practical modulation formats like QAM and observe their performance. It is important to note here that these constellations are more sensitive to the phase and timing offsets and hence, need more precise phase information.
- Extend the framework to allow other pulse shapes so that the spectrum is used efficiently.

REFERENCES

- [1] Y. Wu, P. Chou, and S. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proc. of the Conference on Information Sciences and Systems*, March 2005, pp. 157-168.
- [2] W. Pu, C. Luo, S. Li, and C. W. Chen, "Continuous network coding in wireless relay networks," *IEEE INFOCOM 2008*, in *Proc. of the 27th Conference on Computer Communications*, April 2008, pp. 1526-1534.
- [3] M. Riemensberger, Y. E. Sagduyu, M. L. Honig, W. Utschick, "Comparison of Analog and Digital relay methods with network coding for wireless multicast," in *Proc. of the IEEE International Conference on Communications*, June 2009, pp. 1-5.
- [4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, June 2008, vol. 16, no. 3, pp. 497-510.
- [5] S. Katti, S. Gollakota, and D. Katabi, "Embracing Wireless Interference: Analog Network Coding," in *Proc. of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 07)*, Aug. 2007, pp. 397-408.
- [6] D. Maduiké, H. Pfister, and A. Sprintson, "Design and Implementation of Physical-Layer Network-Coding Protocols," in *Proc. of the 43rd Asilomar Conference on Signals, Systems and Computers*, Oct. 2009, pp. 781 - 787.
- [7] Raymond H. Y. and Yonghui Li, "Practical physical layer network coding for two-way relay channels: Performance analysis and comparison," *IEEE Transactions*

- on *Wireless Communications*, Feb. 2010, vol. 9, no. 2, pp. 764-777.
- [8] T. Rappaport, *Wireless Communications: Principles & Practice*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, Jan. 2002.
- [9] J. Proakis, *Digital Communications*, 5th ed. New York, NY: Mc-Graw Hill, Nov. 2007.
- [10] M. C. Valenti, D. Torrieri, and T. Ferrett, "Non Coherent Physical-Layer Network Coding Using Binary CPFSK Modulation," in *Proc. of the IEEE Military Communication Conference*, Oct. 2009, pp. 413-428.
- [11] T. Koike-Akino, P. Popovski, and V. Tarokh, "Denoising maps and constellations for wireless network coding in two-way relaying systems," *IEEE Global Telecommunications*, Dec. 2008, pp. 1-5.
- [12] T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," *IEEE Journal on selected Areas in Communications*, vol. 27, np. 5, June 2009, pp. 134-152.
- [13] D. Wang, S. Fu, and K. Lu, "Channel coding design to support asynchronous physical layer network coding," *IEEE Globecom*, Dec. 2009, pp. 3267-3272.
- [14] R. Raheli, A. Polydoros, and C. Tzou, "The principle of per-survivor processing: A general approach to approximate and adaptive MLSE," *IEEE Global Telecommunications*, Dec. 1991, pp. 1170-1175.

VITA

Manish Jain received his B. Tech degree in Electronics and Communication engineering from Indian Institute of Technology, Roorkee in 2009. During his undergraduate studies, he did an internship in Honda Research Institute, Japan where he designed a system to be embedded in robot, which would enable it to track the sound source location and direction. He started research work under the guidance of Dr. Scott L. Miller and Dr. Alex Sprintson in January 2010. His research interests include wireless networks, digital communication and signal processing. While pursuing his Masters degree, he was the Teaching Assistant for courses ECEN-444 (Digital Signal Processing), ECEN-455 (Digital Communications) and ECEN-248 (Introduction to Digital Systems Design). During his Masters degree, he went for internship in Qualcomm, Santa Clara in Summer 2010, where he worked as a systems intern in UMTS Modem design group. His summer project involved the study of impact of intermittent scheduling on HSPA+ downlink performance. The company and his manager, impressed by his work, offered him an internship extension for the Fall 2010 semester. During the Fall 2010 semester, he implemented a Genie Equalizer which would establish the upper bound on the performance of HSPA+ Modem. He returned from this six months long internship in Spring 2011, and defended his thesis successfully on 4th March 2011. He will be joining the same group in Qualcomm, as a full time employee, after his graduation in May 2011. Manish Jain can be reached at Office # 241E C/O Dr. Scott Miller, Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX- 77843. He can be reached electronically at the e-mail address *manishjain.iitr@gmail.com*.

The typist for this thesis was Manish Jain.