Blind M-FSK Modulation in the ISI Channel

by

Mohammad Karami

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Channel estimation has received considerable attention over the years for its contribution to more reliable signal decoding. General wireless communication environment would cause multi-path fading for signals that propagate through them. Multi-path fading has two major effects on the system; causing inter symbol interference (ISI) and reshaping signal constellation. Estimating the channel would enable us to combat these two effects. Channel estimation can be done either blindly or with the help of training sequences.

In this thesis, we propose a new blind channel estimation technique for M-FSK modulation systems. Our method can decrease the effect of signal reshaping and thus decreasing the probability of error. It also has the ability to track the channel variations in a time-variant environment.

In our method, an initial estimation is assumed as the channel impulse response. Utilizing this channel, received signals are demodulated and decoded. Based on output of the demodulator, a new estimation is generated for the channel. Consequently, a new output can be produced by exploiting the new channel estimate. This process can be done iteratively n times to reach the minimum possible probability of error.

Acknowledgements

I would like to express my gratitude to my supervisor professor Amir K. Khandani whom without his guidance, this thesis would have not been possible. Furthermore, I would like to thank Mr. Milad Johnny for his useful comments, remarks and engagement through this thesis.

Dedication

This thesis is dedicated to my parents for their everlasting love and support.

Table of Contents

List of Figures

Chapter 1

Introduction

In this chapter, we initially express the necessity of channel estimation and introduce two main channel estimation techniques which are trained or blind. Then, we compare these two techniques and find out why blind estimation is a better option. After that, we review some of the earlier works done on blind channel estimation. And at the of this chapter, we are going to outline a summary of contributions of this thesis.

1.1 Communication Model

The channel model shown in figure 1.1 is considered as our channel model, where $s(t)$ is the transmitted signal, $n(t)$ is AWGN [1] noise, and $y(t)$ is the received signal.

According to figure 1.1, $y(t)$ can be represented as follows

$$y(t) = s(t) * h(t) + n(t) \tag{1.1}$$

or equivalently in the frequency domain as:

$$Y(f) = S(f)\, H(f) + N(f)\,. \tag{1.2}$$

Thus, both the amplitude and phase of $s(t)$ are distorted. The goal of the receiver is to estimate the transmitted signal $\hat{s}(t)$ from the received signal $y(t)$. This can be done by applying the received signal to a new filter with a transfer function equal to the inverse of

[1] additive white Gaussian noise

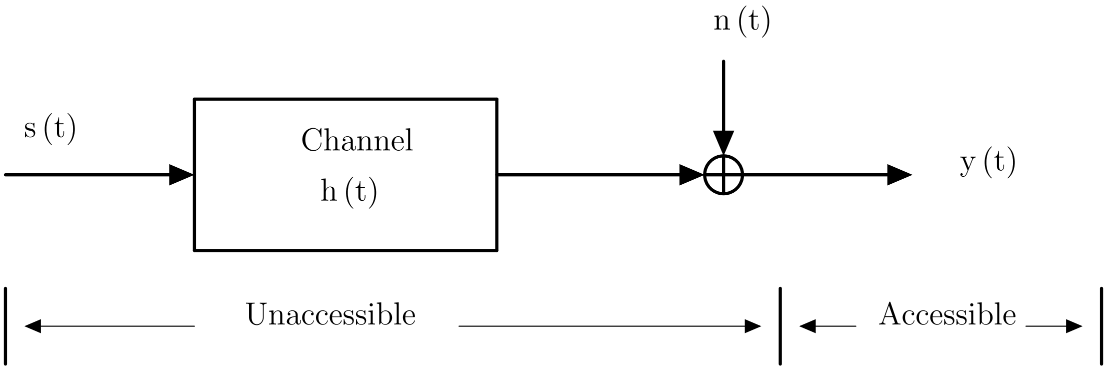

Figure 1.1: System Model

$H(f)$, which would be $H^{-1}(f)$. This procedure is called, channel equalization. Finding the exact value of $H^{-1}(f)$ is not usually feasible because of the practical realization of filter, and the fact that channel is noisy. Therefore, an estimation of channel is found using a linear minimum mean square error (LMMSE) estimator. This estimator minimizes the mean square error between $\hat{s}(t)$ and $s(t)$:

$$\hat{s}(t) = \underset{\hat{s}(t)}{\operatorname{argmin}} \, \mathrm{E}\left[(s(t) - \hat{s}(t))^2\right]. \tag{1.3}$$

Channel identification and equalization techniques can be divided into two major categories, the supervised or trained estimation, and the unsupervised or blind estimation. In trained estimation, there is access to either transmitted signal, or some property or distribution of transmitted signal. Whereas, in blind estimation, the only available information is received signal as is shown in figure 1.1, and estimation is done solely based on that received signal.

Trained estimation usually involves sending training or pilot sequences. In this method, a training sequence is attached to the beginning of transmitted signal. This training sequence is used to estimate the impulse response of channel for the receiver, while allowing the receiver to have a simple structure. In time-varying channels, the channel estimate must be updated fast enough to keep track of changes in channel impulse response. Thus, this method will have problem in fast time-varying channels. Furthermore, sending training sequences would result in reduction of throughput and efficiency.

Because of the problems involved in aforementioned technique, the adaptive channel equalization techniques have been proposed to blindly estimate and equalize the channel.

Blind estimation and equalization of the channel have received considerable attention in the past few decades because of its application in real time communication. This technique has applications in some cases where sending training sequence in not possible, or when there are advantages in reducing data overhead, like wireless transmission over time-varying channels. Some of applications of blind channel estimation are listed below [2]:

- In data communication, an unknown channel frequency response with finite bandwidth would usually cause inter symbol interference (or ISI). In order to eliminate ISI when there is no information about the transmitted signal, channel is needed to be blindly estimated. The existence of time-varying multi-path fading channel that is common in a mobile communication network can cause severe ISI. The varying channel needs to be estimated and if trained estimate is to be used, a major fraction of channel capacity should be dedicated to sending training sequence. Thus, using blind channel estimation which rely only on the received signal would save channel capacity.

- In speech recognition, the received signal at recognizer is the convolution of original speech, impulse response of the channel, and impulse response of the surrounding environment. If training is used to inform the recognizer of transducer, the recognizer might not work in a different transducer or in a different environment. Therefore, it is desirable to build a recognizer that can estimate the channel itself and recover the original speech.

- In image restoration applications like astronomy, remote sensing, or medical imaging, in which the system shows blurring effects caused by camera motion, or inaccurately focused lenses, channel estimation is needed to restore the image. Unfortunately, the original image in these practical cases are not available, and thus blind channel estimation is the only option.

1.2 Blind Channel Estimation Techniques

In this section, we mention some works that are previously done on blind channel estimation, along with their contributions, problems, and in some cases, the environment that they can work optimally.

In earlier works on the blind channel estimation, the knowledge about higher order statistics of the received signal was the main attribute in finding channel coefficients [4], [6], and [25]. These approaches, although reliable, require a large amount of received symbols and a lot of computation to make an accurate estimation. Because of these requirements, these methods have limited application in fast time-varying channels.

Tong, Xu, and Kailath [23] used second-order statistics of the channel output to do blind channel estimation which was made possible by exploiting the cyclostational properties of an oversampled communication signal. Although their method is more computationally efficient than other approaches that rely on high-order statistics, still requires a lot of received signals before it can estimate the channel properly.

In [14], a method is proposed to divide the observation space into two orthogonal subspaces, signal subspace and noise subspace, by applying eigenvalue decomposition on correlation matrix of received blocks. This method is more computationally efficient than [23], and yields reasonable estimates even for short data frames, and thus is used when fast convergence is required.

In [1], Abed-Meraim, Moulines, and Loubaton propose a method based on second order statistics that has restrictions on zero locations of FIR subchannels from oversampling or multiple antennas. However, Ding in [7] showed that [23] and [1] methods cannot be used to blindly estimate all channels and there is a practical limitation on their ability to estimate channels. These methods fail for some channels that have common zeros among sub-channels which are obtained by oversampling.

It should not be forgotten that second-order statistics (SOS) based techniques require a system model that has more than one output signal. However, there are some applications in which there are only one output signal available, and thus SOS is insufficient to reveal all of the information of system unless some properties of the input are known e.g. stationary single output system model. Thus, high-order statistic (HOS) techniques are often necessary for single output systems.

In order to overcome the problem regarding zero positions, Tsatsanis and Giannakis [24] proposed to induce cyclostationarity by repeating input blocks. In other words, instead of introducing cyclostationarity at receiver, they introduce it at transmitter. This cyclostationarity allows the channel to be estimated from second-order statistics only. In their work, they show that coding information can be used to facilitate the receiver's equalization, even though the objective of channel coding is error correction without any concern about channel dispersion. Moreover, channel equalization methods usually assume inputs to be i.i.d without any concern about channel coding. Despite all benefits of their method, repetition of input blocks, reduces information rate.

To minimize this rate reduction, Giannakis himself used a different technique in [10] to introduce cyclostationarity. He used a precoding filterbank at transmitter and a decoding filterbank at receiver, to keep benefits of [24], and also have a small rate reduction. This filterbank precoder at transmitter create input diversity, similar to the diversity obtained by using training sequences.

Cyclostationary-based methods find application in many systems. Heath, and Giannakis exploit its applications in cyclic prefix orthogonal-frequency-division multiplexing (CP-OFDM) systems [11], while Doukopoulos and Moustakides exploit its applications in zero-padding orthogonal-frequency-division multiplexing (ZP-OFDM) systems [8]. OFDM is widely used in modern transmission because of its ease of channel equalization, high spectral efficiency, and flexible data rate. Due to the existence of either cyclic prefix or zero-padding, OFDM systems have cyclostationarity as an inherent feature and can be considered a special case of precoding structure in [10]. By inserting a cyclic prefix (or zero-padded sequence) longer than the channel order, OFDM changes a frequency-selective channel into flat-fading channel. Therefore, only a one-tap equalization is required, but this comes at expense of a loss of 10-25 % in efficiency. Moreover, if the length of the channel impulse response is longer than cyclic prefix (zero-padded sequence), ISI occurs and this simple equalization is not possible anymore.

In some of OFDM systems, some of subcarriers, which are called virtual carriers (VC), are not used for data transmission for various reasons. The method in [13] provided a channel estimation method for OFDM systems by exploiting virtual carriers over time-dispersive channels. This method works for both OFDM with cyclic prefix and without it as long as a sufficient channel identifiability condition is met. As said before, using cyclic prefix longer than the length of channel multi-path spread, would result in a significant loss in channel utilization. Moreover, the channel in some applications is time-varying and would need a periodic training sequence to be transmitted which further reduces throughput. Therefore, OFDM systems with short CP or no CP at all are more desired. Reduction of cyclic prefix allows OFDM to use a higher portion of channel capacity in comparison to previous estimators that were cyclic prefix based.

In [19], the application of cyclostationary-based estimation in code-division multiple access (CDMA) is discussed. It has applications in estimating the down-link channel of TDMA and CDMA systems. Simulations in this paper show that performance of the cyclostationary-based channel estimation is superior to that of RAKE receiver, which assumes perfect knowledge of the channel impulse response, at high SINR when multiuser interference is the dominating disturbance.

Using mathematical models developed in [2], papers [17], [19], and [21] proposed blind channel identification techniques using trailing zeros and they are called subspace-based techniques. In [19], a deterministic method was proposed to estimate the channel. The deterministic nature of the solution, contrary to previous statistical methods, is appealing for transmission over slow-varying channels. In this method, if the length of channel is not greater than the length of trailing zero sequence (guard intervals), channel matrix can be modeled as a Toeplitz matrix and its kernel would be channel coefficients. This method needs to receive at least M blocks of data before it can find channel coefficients. This prevents the method from being effective when channel is changing fast, especially when M is large.

Pham and Manton propose a method that only depend on the presence of gurad intervals to prevent ISI [17]. They derive a practical algorithm to identify and estimate the channel from just two received blocks by using guard intervals. In this method, given two received blocks $Y_1(z) = S_1(z) H(z)$ and $Y_2(z) = S_2(z) H(z)$, the channel can be estimated by computing the greatest common divisor (GCD) of $Y_1(z)$ and $Y_2(z)$, with the condition that $S_1(z)$ and $S_2(z)$ are co-prime. This method can be compared to previous works that has been done on Single Input Multiple Output (SIMO) systems [23], and [14]. In SIMO system, the output of mth channel is given by $Y_m(z) = S(z) H_m(z)$. Therefore, blind channel identification algorithms for SIMO system are basically algorithms for finding the GCD of outputs $\{Y_1(z), Y_2(z), ..., Y_N(z)\}$. In a Single Input Single Output (SISO) system with guard intervals, GCD is $H(z)$, whereas in SIMO systems, GCD is $S(z)$. This means that both of these techniques are essentially trying to do the same thing. Although these algorithms reduced the number of received blocks needed to estimate the channel, they have more computational complexity especially when M is large.

In [21], Su proposes an algorithm that [17],and [19] are special cases of. In it, Su and Vaidyanathan, provide a generalized algorithm that made a trade-off between the number of repetitions and received blocks. These two parameters can be chosen freely as long as they satisfy a certain constraint. They show that when the system parameter or as they call it repetition index (Q) is chosen properly, their generalized algorithm outperforms previous methods in [17] and [19].

In [5], Choi and Lim proposed a method using Cholesky factorization of covariance matrix to estimate the channel impulse response. Cholesky factorization would result in an upper triangular covariance matrix that contains the full information of channel impulse response. A parameterization technique is used to estimate the channel using this information. It should be noted that the computational complexity of the proposed method depends on the length of channel impulse response, whereas the computational complexity of methods in [17], [19], and [21] depend on the length of packet. Therefore, when the length of packet is long, these techniques are much more efficient. Through simulation, this method showed that it continues to be effective even if the number of data blocks is small. This feature is important in time-varying channels.

In [26], Zeng, Li and Cheng proposed another blind and semi-blind channel estimation technique that has relatively lower complexity than [17], [19], and [21], and also requires significantly less number of received blocks compared to them, which is attractive when channel is varying fast. In their paper, they used blind channel estimation for zero-padding single-carrier block transmission (ZP-SCBT) systems using SOS. SCBT has nearly all benefits of OFDM but has significant performance improvement over OFDM in specific environments. SCBT has smaller peak-to-average power ratio (PAPR) than OFDM. Moreover, in SCBT, carrier frequency offsets cause less degradation in SNR than OFDM. Unlike subspace-based techniques, this approach tries to identify the inverse of channel impulse response instead of the channel itself by using the redundancy of zero padding.

Conventional subspace-based algorithms for blind channel estimation in zero-padded systems were frequently used because they can accurately estimate the channel using only a few blocks. However, they don't work when they are applied to ZP-OFDM with VC. Pan and Phoong proposed an improved version of subspace-based algorithm for blind channel estimation using only a few received blocks [15]. In their paper, they proposed an improved algorithm for SISO zero padded systems. Their algorithm has two main advantages compared to previous subspace-based blind channel estimation methods for ZP-OFDM.

First of all, they showed that the deterministic channel estimation method in [21], due to its repetitive structure, would not result in a noise autocorrelation matrix equal to a scaled version of identity matrix, even if the channel noise is independent and identically distributed (i.i.d.). This problem would degrade the performance of estimation. They instead proposed a simple diagonal noise weighting matrix, which unlike the weighted least square methods, depends only on the repetition index (Q) [21] and not on channel variance or SNR.

Furthermore, they showed that methods in [17], [19], and [21] would not work for ZP-OFDM with VC. Instead, they proposed a generalization of these blind estimation methods for ZP-OFDM with VC, which uses both repetition index Q and information that VC has to offer (there is no constraint on the location of the VC). Their simulation results show that their method not only works in presence of VC, it also outperforms previous methods in [17], [19] and [21].

1.3 Contributions of the thesis

In this thesis, M-FSK is considered as the modulation scheme. The FSK modulation symbols are considered as basis vectors for M-dimensional space which are orthogonal to each other. Because of this orthogonality, all of these symbols have the same distance in space with respect to each other. However, as these symbols go through the channel, they do not remain orthogonal to each other and thus some of them will get closer while some of them will become farther than before. Those that got closer are more probable to be mistaken with each other. Thus, making the decoding more susceptible to error. The decoder does not know about reshaped signals, and make its decision based on original bases which causes further degradation in performance. Therefore, by estimating the channel, reshaped signals could be found and error probability will decrease.

In our model, the channel is not flat fading and has more than one tap. This would cause multi-path fading in the received signal. One of the effects of multi-path fading is inter symbol interference (ISI), in which each symbol leaks some of its energy to adjacent symbols. Fortunately, we have low transmission rate which will allow us to work in low SNR where the energy of each symbol is low compared to noise. Therefore, the leaked energy, which is only a portion of each symbol's energy, is not noticeable compared to noise and ISI effect is minimized.

In this thesis, a blind channel estimation method is proposed. Our method uses an iterative algorithm in the receiver to decode the received signal. In each iteration of the algorithm, a new decoded output frame is generated which is an estimate of transmitted frame data. An estimation of channel is then generated from output frame. This channel estimation is then used to derive reshaped bases. At the end of each iteration, a new output is generated from the received signal using new reshaped bases.

Chapter 2

System Design

This chapter is dedicated to representing our system model. In the first section, we give a brief description of our signaling method. In the second section, we define our channel model, multi-path fading, and its effect on our system. Then, we derive signal to noise and interference ratio (SINR) of the system. In the last two sections, product coding and the corresponding decoding method are introduced.

2.1 System Model

We are using multiple frequency shift keying (M-FSK) as our modulation scheme. In this modulation, M signals with different center frequencies are chosen to transmit data. We choose the following set of signals as transmission signals

$$\begin{aligned}
&S_1 = A\sin\left(2\pi f_0 t\right) && 0 \le t \le T_s \\
&S_2 = A\sin\left(2\pi\left(f_0 + \Delta f\right)t\right) && 0 \le t \le T_s \\
&S_3 = A\sin\left(2\pi\left(f_0 + 2\Delta f\right)t\right) && 0 \le t \le T_s \\
&\qquad\vdots \\
&S_M = A\sin\left(2\pi\left(f_0 + \left(M-1\right)\Delta f\right)t\right) && 0 \le t \le T_s\ ,
\end{aligned} \tag{2.1}$$

in which f_0 is an arbitrary modulation frequency. Parameter Δf can be found using the condition that these set of signals should be orthogonal in $[0, T_s]$ interval. Therefore, applying the orthogonality condition would yield

$$\begin{aligned}\langle s_{k_1}, s_{k_2} \rangle &= \int_0^{T_s} A^2 \sin\left(2\pi\left(f_0 + (k_1 - 1)\Delta f\right)t\right) \sin\left(2\pi\left(f_0 + (k_2 - 1)\Delta f\right)t\right) \\ &= \int_0^{T_s} \frac{A^2}{2}\left[\cos\left(2\pi\left(2f_0 + (k_1 + k_2 - 2)\Delta f\right)t\right) - \cos\left(2\pi\left(k_1 - k_2\right)\Delta f t\right)\right] \\ &= \frac{A^2}{2}\left[\frac{\sin\left(2\pi\left(2f_0 + (k_1 + k_2 - 2)\Delta f\right)t\right)}{2\pi\left(2f_0 + (k_1 + k_2 - 2)\Delta f\right)} - \frac{\sin\left(2\pi\left(k_1 - k_2\right)\Delta f t\right)}{2\pi\left(k_1 - k_2\right)\Delta f}\right]_0^{T_s} = 0 \quad (2.2)\end{aligned}$$

for every $k_1, k_2 \in \{1, 2, ..., M\}$ and $k_1 \neq k_2$. Due to the fact that $f_0 \gg \sin(\alpha)$, the first term in (2.2) would be negligible. Therefore, we get

$$\frac{1}{4\pi\left(k_1 - k_2\right)\Delta f} \sin\left(2\pi\left(k_1 - k_2\right)\Delta f t\right)\Big|_0^{T_s} = 0.$$

The above equation can be simplified to

$$\sin\left(2\pi\left(k_1 - k_2\right)\Delta f T_s\right) - \sin 0 = 0. \quad (2.3)$$

The argument of sine should be a multiplier of π in 2.3. Therefore,

$$2\pi\left(k_1 - k_2\right)\Delta f T_s = k\pi.$$

The minimum value for Δf is desirable so k will be set as $k = 1$ and

$$\Delta f = \frac{1}{2\left(k_1 - k_2\right)T_s}.$$

Because adjacent signals should also be orthogonal, $k_1 - k_2$ should be equal to one and Δf would be

$$\Delta f = \frac{1}{2T_s} \quad (2.4)$$

Parameter A has to be determined based on the fact that the energy of each basis signal should be $\mathcal{E}$ for every i

$$\begin{aligned}\langle s_i, s_i \rangle &= \int_0^{T_s} A^2 \sin^2\left(2\pi\left(f_0 + (i - 1)\Delta f\right)t\right)dt \\ &= \int_0^{T_s} A^2 \left(\frac{1 - \cos\left(4\pi\left(f_0 + (i - 1)\Delta f\right)t\right)}{2}\right) dt = \mathcal{E}.\end{aligned}$$

Therefore, we have

$$\frac{A^2}{2}\left[t-\frac{\sin\left(4\pi\left(f_0+\left(i-1\right)\Delta f\right)t\right)}{4\pi\left(f_0+\left(i-1\right)\Delta f\right)}\right]_0^{T_s}=\mathcal{E}. \tag{2.5}$$

In equation (2.5), the second term in brackets contains f_0 in denominator. This means that the value of denominator is big compared to the sine term in numerator and can be ignored. This will result in

$$\frac{A^2}{2}T_s=\mathcal{E}, \tag{2.6}$$

or equivalently

$$A=\sqrt{\frac{2\mathcal{E}}{T_s}}. \tag{2.7}$$

Referring to equation 2.1, these set of signals are M *sinc* functions at M center frequencies in the frequency domain. These basis signals can be interpreted in a different manner e.g. a space with M basis vectors. This interpretation can be seen in figure 2.1

The set of signals in 2.1 are used to transmit data. Each time that we want to send a symbol, we transmit one of these signals instead. The chosen basis signal experience multi-path fading and is added to noise before arriving at the receiver. At the receiver, the received signal is then projected into basis vectors of M-dimensional space and these projections form the received vector.

The main criteria in deciding which basis was sent is to find the minimum distance between the received vector and basis signals. Therefore, we introduce a matrix that measures the distance between each two of these bases in 2.1, called distance matrix. Distance matrix is a symmetric matrix with zero diagonal elements. The distance matrix is represented as follows

$$D=\left[\begin{array}{ccccc} d_{1,1} & d_{1,2} & d_{1,3} & .... & d_{1,M} \\ d_{2,1} & d_{2,2} & d_{2,3} & .... & d_{2,M} \\ d_{3,1} & d_{3,2} & d_{3,3} & .... & d_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & d_{M,3} & .... & d_{M,M} \end{array}\right] \tag{2.8}$$

The error probability of the system can be represented as follows

$$P_e=\sum_i\sum_j P\left(s=i\right)P\left(\hat{s}=j|s=i\right) \qquad i\neq j\ ,$$

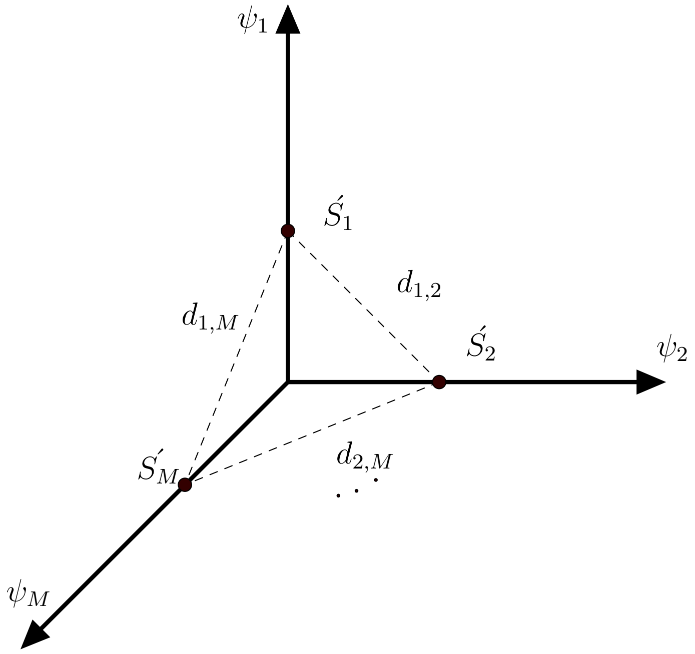

Figure 2.1: Basis vectors in M-dimension space

where $P\left(s=i\right)$ is the probability that symbol i has been sent and $P\left(\hat{s}=j|s=i\right)$ is the probability that ith symbol has been sent but it is demodulated as jth symbol.

In the presented interpretation, all bases are equally spaced with respect to each other, which means that all entries except $d_{i,i}$ are equal to $2\sqrt{\frac{\mathcal{E}}{T_s}}$. Therefore, the probability that the ith symbol is mistaken with jth symbol is the same for all i and j as long as $j \neq i$. Therefore, the error probability will become

$$P_e = (M-1)\sum_i P\left(s=i\right)P\left(\hat{s}=j_0|s=i\right) \ ,$$

in which $j_0 \in \{1,2,...,M\} - \{i\}$. $P\left(\hat{s}=j_0|s=i\right)$ is the same for all values of i because all bases have the same distance, so

$$P_e = (M-1)\sum_i P\left(s=i\right)P\left(\hat{s}=j_0|s=i_0\right) = (M-1)\,P\left(\hat{s}=j_0|s=i_0\right). \qquad (2.9)$$

As it is evident from (2.9), error probability is independent of transmitted and received symbols.

2.2 Channel Model

In our system, the transmitted signal goes through a channel that has an specific impulse response and also adds noise to the transmitted signal. Thus, the received signal can be represented as follows

$$y\left(t\right) = x\left(t\right) * h\left(t\right) + n\left(t\right) \tag{2.10}$$

in which the channel $h\left(t\right)$ can be modeled with L different taps in time as the following equation

$$h\left(t\right) = \sum_{i=0}^{L-1} a_i \delta\left(t - i\tau\right), \tag{2.11}$$

where

$$\sum_{i=0}^{L-1} a_i^2 = 1 \tag{2.12}$$

in which a_i represents the ith tap of channel.

Substituting channel model of 2.11 in 2.10 would give us the received signal $y\left(t\right)$. This channel model causes some problems for us. First of all, if the channel has more than one tap, then it causes multi-path fading for transmitted signals. One of the effects of multi-path fading is the leakage of some portion of the energy of each symbol into next symbols. This leakage would cause ISI. Fortunately, our designed system has low transmission rate and therefore operates in low SINR. When SINR is low, the energy leakage of one symbol is not noticeable in comparison to the energy of noise.

Another problem generated by channel model of 2.11 is signal reshaping. After $S_i\left(t\right)$ $i \in \{1, 2, ..., M\}$ went through the channel, they are not in the previous position in M-dimensional space. For instance, $S_i\left(t\right)$ can be represented by $\sum \alpha_i \psi_i\left(t\right)$. This signal reshaping causes some of vectors to become closer to each other and thus more susceptible to error. Therefore, P_e depends more on these close vectors. As a result, equation 2.9 is not valid anymore and P_e is bigger than the value calculated by 2.9.

Referring to figure 2.2, the error probability can be decreased if the distance is computed between the received signal and $S_i^{'}$ instead of being computed between the received signal and S_i. This decrease in error depends on either knowing the channel, or somehow being able to estimate it. Therefore, if the receiver has access to reshaped bases $\{S_1^{'}, S_2^{'}, ..., S_M^{'}\}$, the error probability can be reduced drastically.

In this work, we have two different types of information. The first one is the information which is related to the transmitted data and the second one is the information about

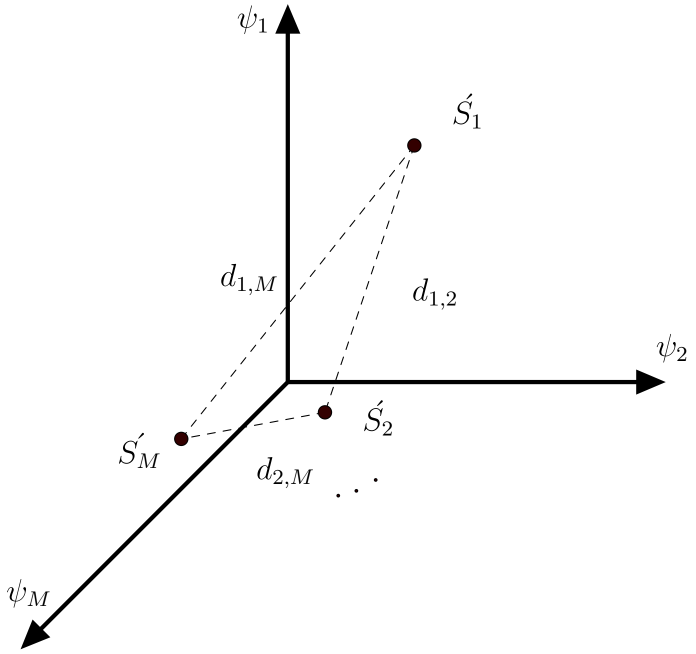

Figure 2.2: Basis vectors convolved with channel

channel state. The second type of information has a key role in decodability of received signals. However, the channel has limited capacity which generates some constraints on transmission rate and the amount of channel distortion. Lower channel distortion at the receiver would result in a more precise estimation of the location of distorted symbols in signal constellation, but occupies higher portion of channel capacity to itself. Therefore, the remaining channel capacity which can be dedicated to data transmission reduces. Thus, there is a trade-off between high precision and channel capacity available for data transmission.

Since it would take infinite amount of bits to send a random real number, each tap of $h(t)$ in its continuous format needs infinite bits per transmission to be sent. On the other hand, a finite set of discrete numbers can be expressed by limited amount of bits. Thus, channel information can be sent to the receiver if we accept distortion in channel tap values.

The channel distortion for each tap is λ. Therefore, the estimated channel can be represented by

$$\hat{h}\left(t\right)=\sum_{k=0}^{L-1}\hat{a}_k\delta\left(t-k\tau\right) \tag{2.13}$$

with $\hat{a}_k$ being

$$\hat{a}_k=\lfloor\frac{a_k}{\lambda}\rfloor\lambda. \tag{2.14}$$

In this estimation, the channel estimation error will be

$$\Delta\hat{h}\left(t\right)=\sum_{k=0}^{L-1}\Delta\hat{a}_k\delta\left(t-k\tau\right) \tag{2.15}$$

where

$$\Delta\hat{a}_k=a_k-\hat{a}_k \tag{2.16}$$

and $|\Delta\hat{a}_k|\leq\lambda$. Therefore, the channel model of 2.10 can be extended as follows

$$\begin{aligned}y\left(t\right)&=x\left(t\right)*h\left(t\right)+n\left(t\right)\\&=x\left(t\right)*\left(\hat{h}\left(t\right)+\Delta\hat{h}\left(t\right)\right)+n\left(t\right)\\&=x\left(t\right)*\hat{h}\left(t\right)+x\left(t\right)*\Delta\hat{h}\left(t\right)+n\left(t\right)\end{aligned} \tag{2.17}$$

where $x\left(t\right)*\hat{h}\left(t\right)$ is the desired term. $x\left(t\right)*\Delta\hat{h}\left(t\right)$ along with $n\left(t\right)$ are unknown parameters that represent distorted transmitted signal and noise respectively. Based on above equations, signal to interference and noise ratio will be

$$\text{SINR}=\frac{\text{Power of }\{x\left(t\right)*\hat{h}\left(t\right)\}}{\text{Power of }\{x\left(t\right)*\Delta\hat{h}\left(t\right)+n\left(t\right)\}}\ , \tag{2.18}$$

or equivalently in the frequency domain we have

$$\text{SINR}\left(f\right)=\frac{\left|X\left(f\right)\right|^2\left|\hat{H}\left(f\right)\right|^2}{\left|X\left(f\right)\right|^2\left|\Delta\hat{H}\left(f\right)\right|^2+\sigma_n^2}, \tag{2.19}$$

where the interference comes from the signal itself. Thus, one of the methods to compute SINR is to compute the power spectral density (or PSD) of $x\left(t\right)$, $\hat{h}\left(t\right)$, and $\Delta\hat{h}\left(t\right)$.

First of all, the PSD of transmitted signal is going to be calculated. The transmitted signal is not diterministic, thus PSD can only be calculated through autocorrelation function. As it is shown in [9], the autocorrelation function for this phase-incoherent FSK signalling can be found by expressing the transmitted signal as the following form:

$$x\left(t\right)=\sum_{n=0}^{\infty}a_n\left(t\right)p\left(t-nT_s\right) \tag{2.20}$$

where

$$a_n\left(t\right)=\cos\left(2\pi\left(f_0+f\left(n\right)\right)t+\theta_n\right), \tag{2.21}$$

and

$$f\left(n\right)=\sum_{m=1}^{M}\delta_m\left(n\right)f_m. \tag{2.22}$$

In the above equations, M is alphabet size, $p\left(t\right)$ is pulse shaping signal, f_m is the frequency of the mth basis, and

$$\delta\left(n\right)=\left[\delta_1\left(n\right),\delta_2\left(n\right),...,\delta_M\left(n\right)\right] \tag{2.23}$$

is an indicator vector (for each value of n) with one element equal to unity and the rest equal to zero and $\delta_m\left(n\right)$ is the mth element of that vector.

If θ_n is an independent sequence, has uniform distribution on the interval $[-\pi,\pi]$, and is statistically independent of $\{f\left(n\right)\}$, then it can be shown that autocorrelation for this phase-incoherent FSK signal is given by:

$$R_x\left(\tau\right)=\frac{1}{2T_s}r_p\left(\tau\right)Z_y\left(\tau\right) \tag{2.24}$$

where

$$r_p\left(\tau\right)=\int_{-\infty}^{\infty}p\left(t-\tau/2\right)p\left(t+\tau/2\right)dt \tag{2.25}$$

and

$$Z_y\left(\tau\right)=\lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N}y_n\left(\tau\right) \tag{2.26}$$

for which

$$y_n\left(\tau\right)=\cos\left(2\pi\left[f_0+f\left(n\right)\right]\tau\right). \tag{2.27}$$

Thus, the power spectral density is given by the following convolutional form:

$$S_x\left(f\right)=\frac{1}{2T_s}\left[P\left(f\right)P^*\left(f\right)\right]\otimes\int_{-\infty}^{\infty}Z_y\left(\tau\right)e^{-i2\pi f\tau}d\tau\ , \tag{2.28}$$

where $P(f)$ is the Fourier transform of $p(t)$. In this case, $p(t)$ is assumed to be a full-duty-cycle rectangle pulse

$$p(t) = \begin{cases} 1, & |t| \leq T_s/2 \\ 0, & |t| > T_s/2 \end{cases} \tag{2.29}$$

which in the frequency domain can be translated to

$$P(f) = \frac{\sin \pi f T_s}{\pi f}. \tag{2.30}$$

If $\{f(n)\}$ is stationary and basis signals have discrete distribution of $\{Pr_m\}_1^M$, then (2.26) can be simplified to

$$Z_y(\tau) = \sum_{m=1}^{M} Pr_m \cos\left(2\pi\left(f_0 + f_m\right)\tau\right)\ , \tag{2.31}$$

and the PSD of $x(t)$ will be

$$S_x(f) = \frac{1}{4T_s}\sum_{m=1}^{M} Pr_m \left[Q\left(f + f_0 + f_m\right)Q^*\left(f + f_0 + f_m\right) + Q\left(f - f_0 - f_m\right)Q^*\left(f - f_0 - f_m\right)\right]. \tag{2.32}$$

After the calculation of PSD for $x(t)$, the PSD for channel estimate $(\hat{h}(t))$ and channel error $(\Delta\hat{h}(t))$ will be calculated. Therefore, the Fourier transform of the estimated channel in (2.15) is taken

$$\hat{H}(f) = \sum_{k=0}^{L-1} \hat{a}_k e^{-i2\pi k f \tau}, \tag{2.33}$$

and PSD of this estimation will be

$$\begin{aligned} S_{\hat{h}}(f) &= \hat{H}(f)\,\hat{H}^*(f) \\ &= \left(\sum_{k=0}^{L-1} \hat{a}_k e^{-i2\pi k f\tau}\right)\left(\sum_{u=0}^{L-1} \hat{a}_u^* e^{i2\pi u f\tau}\right) \\ &= \sum_{u=0}^{L-1}\sum_{k=0}^{L-1} \hat{a}_k \hat{a}_u^* e^{-i2\pi(k-u)f\tau} \end{aligned} \tag{2.34}$$

Because estimation and error have the same form, Δh has the same PSD form $\hat{h}$

$$S_{\Delta\hat{h}}(f) = \sum_{u=0}^{L-1}\sum_{k=0}^{L-1} \Delta\hat{a}_k \Delta\hat{a}_u^* e^{-i2\pi(k-u)f\tau}. \tag{2.35}$$

$\mathrm{SINR}\left(f\right)$ can be computed based on $\left|X\left(f\right)\right|^2$, $\left|\hat{H}\left(f\right)\right|^2$, and $\left|\Delta\hat{H}\left(f\right)\right|^2$. The capacity of a channel with AWGN noise can calculated by the following relation

$$C = \frac{\mathrm{BW}}{2}\log_2\left(1+\mathrm{SINR}\right). \tag{2.36}$$

where BW is the signaling bandwidth. Then the total capacity will be

$$C = \frac{1}{2}\int_{\mathrm{BW}}\log_2\left[1+\frac{\left|X\left(f\right)\right|^2\left|\hat{H}\left(f\right)\right|^2}{\left|X\left(f\right)\right|^2\left|\Delta\hat{H}\left(f\right)\right|^2+\sigma_n^2}\right]df. \tag{2.37}$$

As it was discussed before, this capacity can upper bound both the transmission rate R and the channel coefficients rate $R_{\hat{h}}$ or in other words we have

$$R+R_{\hat{h}}\leq C. \tag{2.38}$$

One of the constraints on channel is defined in equation (2.12). Based on this equation, the following should hold for every i

$$\left|h_i\right|\leq 1. \tag{2.39}$$

In order to have a channel estimation with distortion less than $L\lambda$, each tap of channel can be identified by a discrete random variable from $\{-1,-1+\lambda,...,\lfloor\frac{1}{\lambda}\rfloor\lambda\}$ with equal probability. Therefore, identifying the channel with distortion of $L\lambda$ occupies the following portion of channel capacity

$$R_{\hat{h}} = \sum_{a_0}...\sum_{a_{L-1}}\Pr\left(\hat{a}_0,...,\hat{a}_{L-1}\right)\log_2\left(\frac{1}{\Pr\left(\hat{a}_0,...,\hat{a}_{L-1}\right)}\right) \tag{2.40}$$

which can be upper bounded as follows

$$\begin{aligned} R_{\hat{h}} \leq &\sum_{a_0}\Pr\left(\hat{a}_0=x_i\right)\log_2\left(\frac{1}{\Pr\left(\hat{a}_0=x_i\right)}\right)\\ &+...\\ &+\sum_{a_{L-1}}\Pr\left(\hat{a}_{L-1}=x_i\right)\log_2\left(\frac{1}{\Pr\left(\hat{a}_{L-1}=x_i\right)}\right). \end{aligned} \tag{2.41}$$

However, each tap has the same distribution. Thus, the above inequality can be simplified as follows

$$\begin{aligned}R_{\hat{h}} &\leq LH\left(\hat{a_k}\right) = L\sum_{x_i}\Pr\left(X = x_i\right)\log_2\left(\frac{1}{\Pr\left(X = x_i\right)}\right)\\ &= L\sum_{x_i}\frac{1}{\lfloor\frac{2}{\lambda}\rfloor}\log_2\left(\lfloor\frac{2}{\lambda}\rfloor\right)\\ &= L\log_2\left(\lfloor\frac{2}{\lambda}\rfloor\right).\end{aligned} \tag{2.42}$$

The rate computed in 2.42 is for channels that are time-invariant. If a channel changes every T_c seconds and each symbol time is T_s, then the channel entropy with acceptable $L\lambda$ distortion would be

$$R_{\hat{h}} = H\left(\hat{h}\right) \leq \frac{T_c}{T_s}L\log_2\left(\lfloor\frac{2}{\lambda}\rfloor\right) \tag{2.43}$$

Based on 2.38 and 2.43, the achievable data rate can be upper bounded by the following inequality:

$$R + \frac{T_c}{T_s}L\log_2\left(\lfloor\frac{2}{\lambda}\rfloor\right) \leq \int_{\mathrm{BW}}\log_2\left[1 + \frac{\left|X\left(f\right)\right|^2\left|\hat{H}\left(f\right)\right|^2}{\sigma_n^2 + \left|X\left(f\right)\right|^2\left|\Delta\hat{H}\left(f\right)\right|^2}\right] \tag{2.44}$$

2.3 Product Code

In this section, a brief introduction of product code is provided based on the works of Pellikaan, Wu, Bulygin, and Jurrius in [16].

If C_1 is a $[n_1, k_1, d_1]$ code, C_2 is a $[n_2, k_2, d_2]$ code, ..., and C_N is a $[n_N, k_N, d_N]$ code, where n_i is the length of the ith code's codeword, k_i is the length of the ith code's message, and d_i is the minimum distance of ith code, then product code denoted by $C_1 \otimes C_2 \otimes ... \otimes C_N$ is defined by

$$C_1 \otimes C_2 \otimes ... \otimes C_N = \left\{ \left(c_{ij...u}\right)_{1\leq i\leq n_1, 1\leq j\leq n_2, ..., 1\leq u\leq n_N} \;\middle|\; \begin{array}{ll} \left(c_{ij...u}\right)_{1\leq i\leq n_1} \in C_1 & \text{for all } i \\ \left(c_{ij...u}\right)_{1\leq j\leq n_1} \in C_2 & \text{for all } j \\ \vdots & \\ \left(c_{ij...u}\right)_{1\leq u\leq n_N} \in C_N & \text{for all } u \end{array} \right\}. \tag{2.45}$$

In $C_1 \otimes C_2$, $\otimes$ indicate the *Kronecker* product of C_1 and C_2.

The entries of this code are $k_1 \times k_2 \times ... \times k_N$ arrays with elements from a common GF(q) (a Galois field of order q) and the codewords are $n_1 \times n_2 \times ... \times n_N$ matrices with elements from the same GF(q).

Based on this definition, $C_1 \otimes C_2 \otimes ... \otimes C_N$ is the set of all $n_1 \times n_2 \times ... \times n_N$ arrays whose columns belong to C_1, rows to C_2, ..., and the Nth dimension vectors to C_N. This product code will be a $[n_1 n_2 ... n_N, k_1 k_2 ... k_N, d_1 d_2 ... d_N]$ code and this new code has information rate of $R = R_1 R_2 ... R_N = \frac{k_1 k_2 ... k_N}{n_1 n_2 ... n_N}$. It is worth noting that $C_1 \otimes C_2 \otimes ... \otimes C_N$ is linear if all of C_1, C_2, ...,C_N are linear.

As it was shown, combining two or more codes will result in a product code. Product code is also called direct product, *Kronecker* product, or tensor product code.

Here, for simplicity, C_1, C_2, ...,C_N are all considered as parity check codes. This means that in the ith code, $n_i = k_i + 1$ and the parity check symbol can be found solving the following equation

$$\left(\sum_{l=1}^{k_i} x_l^i \right) + x_n \equiv 0 \qquad (mod\ q) \tag{2.46}$$

in which x_l^i, $l \in \{1, ..., k_i\}$ are known and x_n is the unknown parity.

As N increases, each symbol is in a higher number of parity equations and the probability that this symbol is mistaken with another one is reduced. Since this symbol should check in N different equations, there is a small chance that an incorrectly decoded symbol would satisfy all of them. Thus, if there are constant number of symbols, arranging them in a higher dimension cube would result in a lower probability of error. However, it should be noted that with changing the values of n, k, and N, the spectral efficiency also changes.

As an example, a $C_1 \otimes C_2$ product code is shown in figure 2.3. As it can be seen, message bits are put in the upper left hand corner, and parity bits are calculated in each row and column. Moreover, symbols in the down right hand corner are parity checks on parity checks, which means that even parity bits are held in their own parity equation.

As it is proved in [3], one of the facts that should be considered for product code is that, utilizing it will reduce each code's ability to correct errors that have length up to half the code's length. Consider the following for the above example

$$\begin{aligned} \mu_1 &= \frac{d_1}{n_1} \\ \mu_2 &= \frac{d_2}{n_2}, \end{aligned} \tag{2.47}$$

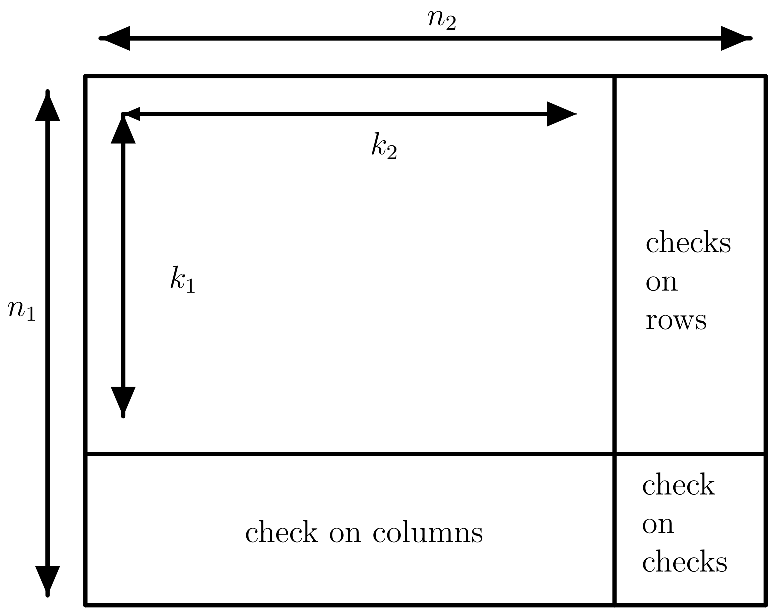

Figure 2.3: product code with N=2

where μ_1 and μ_1 show the ability of each code to correct errors up to half their lengths, and for product code this value is

$$\mu = \mu_1\mu_2 = \frac{d_1 d_2}{n_1 n_2} \leq \mu_1 \text{ and } \mu_2 \tag{2.48}$$

which is less than the correction ability of both of codes.

Although having less minimum distance is not a good quality for product code, it should also be noted that product code has interleaving as an inherent feature in itself, thus making this code robust to burst errors. Furthermore, the simple structure of product code along with the use of low complexity decoding algorithm , has made product code appealing.

2.4 Decoding of Product Code

For decoding product code, a soft iterative symbol decoding algorithm that operates on a code trellis is used. This algorithm is called *BCJR* after its founders Bahl, Cocke, Jelinek and Raviv. This algorithm is also called Maximum a Posteriori (MAP)or forward-backward algorithm. There are some simplified versions of MAP algorithm, namely log-MAP and max-log-MAP algorithms, and in this thesis, the log-MAP is used to decode.

As it is stated in [12] and [18], a MAP symbol decoder maximizes the a posteriori symbol probabilities. This algorithm initially computes so called a posteriori probability (or APP) values for each symbol

$$\mathrm{APP}^{(k)}\left(x_j\right) = \ln\left(\Pr\left(x_j = k \mid r\right)\right), \tag{2.49}$$

where x_j is the jth transmitted symbol, r is the received signal, and $k \in \{1, 2, ..., M\}$ is the number of each basis. After computing these APPs, the decoder make use of the fact that each symbol is in N parity equations and thus uses the information of other symbols to decode that symbol. These APPs handover among different symbols for a number of iterations that can be set. In each iteration, these APPs are computed again using the information of other symbols.

Finally, the demodulator uses hard decision to find the estimate of x_j

$$\hat{x}_j = \underset{k}{max}\{\mathrm{APP}^{(k)}\left(x_j\right)\}. \tag{2.50}$$

In order to find these APPs, BCJR algorithm computes and uses the following metrics in trellis

$$\begin{aligned}
&\alpha_j\left(s_k\right), && \text{forward metric} && (2.51)\\
&\beta_j\left(s_k\right), && \text{backward metric} && (2.52)\\
&\gamma_j\left(s_k, s_u^{\prime}\right), && \text{branch metric} && (2.53)
\end{aligned}$$

The forward metric is the probability of being in state s_k at time j given the received sequence up to time j in forward direction, whereas the backward metric is the probability of being in state s_k at time j given the received sequence up to time j in backward direction. For a given received vector at time j, $\gamma_j\left(s_k, s_u^{\prime}\right)$ is the probability of going from state s_k to $s_u^{\prime}$.

At first, all metrics in the trellis are calculated, and then the APP for each transmitted symbol and each basis will be computed by the following equation

$$\mathrm{APP}^{(k)}\left(x_j\right) = \underset{\left(s_k, s_u^{\prime}\right)}{m\hat{a}x}\{\hat{\alpha}_j\left(s_k\right), \hat{\gamma}_j\left(s_k, s_u^{\prime}\right), \hat{\beta}_{j+1}\left(s_u^{\prime}\right)\}, \tag{2.54}$$

where s_k is state at time j, $s_u^{\prime}$ is state at time j+1, and $\left(s_k, s_u^{\prime}\right)$ is the set of all state pairs corresponding to input being $x_j = k$ at time j. Metrics with hat are logarithmic versions

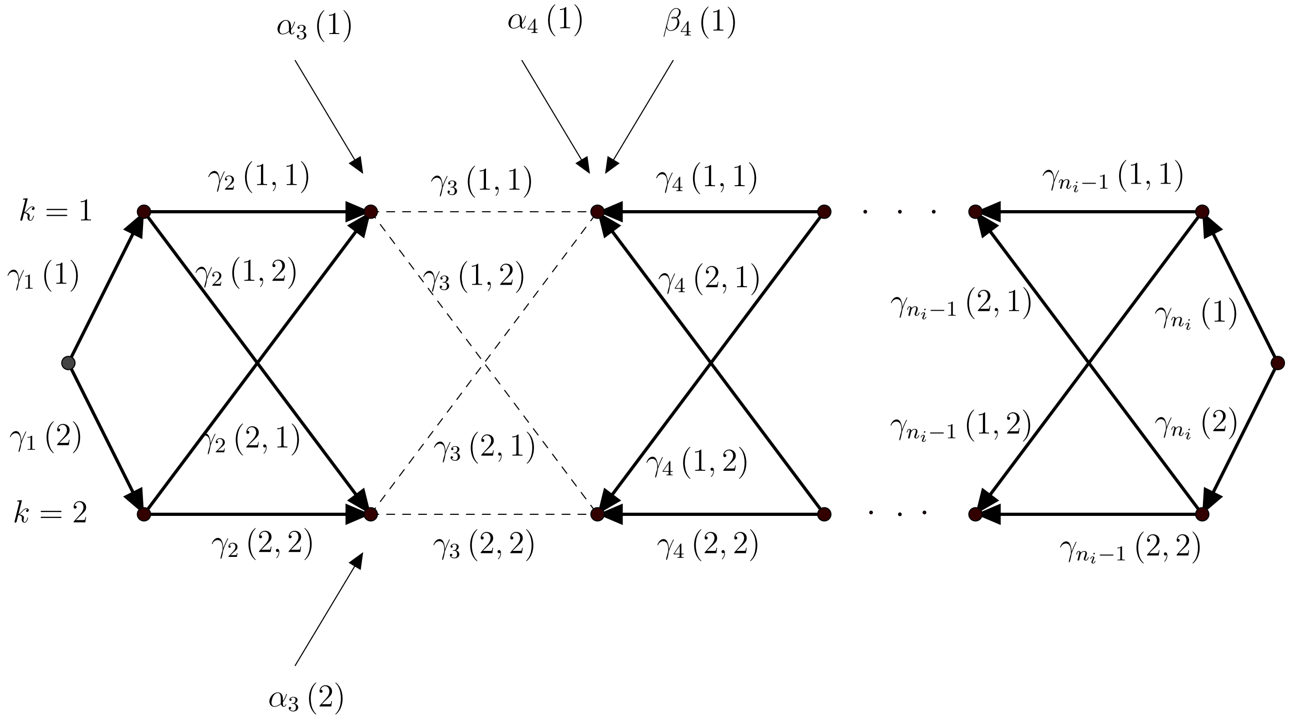

Figure 2.4: Three metrics on trellis for M=2

of original metrics, meaning that:

$$\hat{\alpha}_j(s_k) = \ln(\alpha_j(s_k)) \tag{2.55}$$

$$\hat{\beta}_j(s_k) = \ln(\beta_j(s_k)) \tag{2.56}$$

$$\hat{\gamma}_j(s_k, \acute{s_u}) = \ln(\gamma_j(s_k, \acute{s_u})) \tag{2.57}$$

and m$\hat{\text{a}}$x operation is defined as follows

$$\underset{1\le n\le N}{\hat{\max}}\{x_n\} \triangleq \ln\left(\sum_{n=1}^{N} e^{x_n}\right) \tag{2.58}$$

.

At the end of last iteration, remaining APPs are then considered as outputs of decoder and these outputs are fed into demodulator to determine which of these APPs in each x_j is bigger than others.

As an example, the trellis for M=2 is shown in figure 2.4 along with all three metrics. As it can be seen, for computing $\mathrm{APP}^{(k)}\left(x_3\right)$, $\alpha_3\left(s_k\right)$s and $\beta_4\left(s_k\right)$s are considered along with γ_3s, and they are used in equation (2.54) to find APP.

Chapter 3

Analysis of Error Probability

This chapter is dedicated to channel estimation and the error probability associated with it. At first, we find the error probability of our system model. After that, we define the procedure in which we estimate the channel with. At the end of the chapter, we show the relation between our estimation method and the error probability.

3.1 Error Probability Computation

The set of sinusoidal signals 2.1 that were used to transmit data can be shown in the M-dimension space as

$$\begin{aligned} S_1 &= \left(\sqrt{\frac{2\mathcal{E}}{T_s}}, 0, ..., 0\right) \\ S_2 &= \left(0, \sqrt{\frac{2\mathcal{E}}{T_s}}, ..., 0\right) \\ &\vdots \\ S_M &= \left(0, 0, ..., \sqrt{\frac{2\mathcal{E}}{T_s}}\right). \end{aligned} \tag{3.1}$$

After going through the channel, basis signals are reshaped by the channel impulse response and noise. Based on the channel model that was introduced in chapter 2, each reshaped basis vector will be computed by

$$S_i^r(t) = S_i(t) * h(t) + n(t) = S_i^{'}(t) + n(t), \tag{3.2}$$

where $i \in \{1, 2, ..., M\}$. In the above equation, $S_i^{'}(t)$ is the reshaped basis signal corresponding to channel $h(t)$. Utilizing the following equation

$$h(t) = \sum_{k=0}^{L-1} a_k \delta(t - k\tau)$$

$S_i^{'}(t)$ would become

$$S_i^{'}(t) = S_i(t) * \sum_{k=0}^{L-1} a_k \delta(t - k\tau), \tag{3.3}$$

or equivalently

$$S_i^{'}(t) = \sum_{k=0}^{L-1} a_k S_i(t - k\tau). \tag{3.4}$$

By replacing $S_i(t)$ with its sinusoidal signal form in 2.1, $S_i^{'}(t)$ will become

$$\begin{aligned} S_i^{'}(t) &= \sum_{k=0}^{L-1} a_k \sqrt{\frac{2\mathcal{E}}{T_s}} \sin(2\pi(f_0 + (i-1)\Delta f)(t - k\tau)) \\ &= \sum_{k=0}^{L-1} a_k \sqrt{\frac{2\mathcal{E}}{T_s}} \left[\sin(2\pi(f_0 + (i-1)\Delta f)t - 2\pi(f_0 + (i-1)\Delta f)k\tau)\right]. \end{aligned} \tag{3.5}$$

By using the formula $\sin(\alpha - \beta) = \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta)$, $S_i^{'}(t)$ would become

$$\begin{aligned} S_i^{'}(t) = &\sum_{k=0}^{L-1} a_k \sqrt{\frac{2\mathcal{E}}{T_s}} \sin(2\pi(f_0 + (i-1)\Delta f)t)\cos(2\pi(f_0 + (i-1)\Delta f)k\tau) \\ &- \sum_{k=0}^{L-1} a_k \sqrt{\frac{2\mathcal{E}}{T_s}} \cos(2\pi(f_0 + (i-1)\Delta f)t)\sin(2\pi(f_0 + (i-1)\Delta f)k\tau) \end{aligned} \tag{3.6}$$

or

$$S_i^{'}(t) = S_{i_1}^{'}(t) - S_{i_2}^{'}(t). \tag{3.7}$$

In M-dimensional space, this reshaped signal can be expressed by projecting it onto each one of the space basis vectors with unit energy

$$S_i^{'} = (C_{i,1}, ..., C_{i,i}, ..., C_{i,M}). \tag{3.8}$$

$\{\psi_1, \psi_2, ..., \psi_M\}$ are defined as orthonormal bases of M-dimensional space. Therefore, each one of them is defined by the following equation

$$\psi_i = \frac{S_i}{\sqrt{\int_0^{T_s} S_i^2 dt}} = \frac{S_i}{\sqrt{\mathcal{E}}} = \sqrt{\frac{2}{T_s}} \sin\left(2\pi\left(f_0 + (i-1)\Delta f\right)t\right) \qquad 0 \le t \le T_s. \tag{3.9}$$

For $j \neq i$, these projections are computed by

$$\begin{aligned} C_{i,j} &= \int_0^{T_s} S_i'(t)\, \psi_j(t)\, dt = \int_0^{T_s} \left(S_{i_1}'(t) - S_{i_2}'(t)\right) \psi_j(t)\, dt \\ &= \int_0^{T_s} S_{i_1}'(t)\, \psi_j(t)\, dt - \int_0^{T_s} S_{i_2}'(t)\, \psi_j(t)\, dt = C_{i,j}^{(1)} - C_{i,j}^{(2)}. \end{aligned} \tag{3.10}$$

Each term in $C_{i,j}$ will be calculated separately, starting with $C_{i,j}^{(1)}$. We have

$$\begin{aligned} C_{i,j}^{(1)} = \int_0^{T_s} &\left[\sum_{k=0}^{L-1} a_k \sqrt{\frac{2\mathcal{E}}{T_s}} \sin\left(2\pi\left(f_0 + (i-1)\Delta f\right)t\right) \cos\left(2\pi\left(f_0 + (i-1)\Delta f\right)k\tau\right)\right] \times \\ &\sqrt{\frac{2}{T_s}} \sin\left(2\pi\left(f_0 + (j-1)\Delta f\right)t\right) dt \end{aligned}, \tag{3.11}$$

or equivalently

$$C_{i,j}^{(1)} = \sum_{k=0}^{L-1} a_k \sqrt{\frac{1}{\mathcal{E}}} \cos\left(2\pi\left(f_0 + (i-1)\Delta f\right)k\tau\right) \langle S_i, S_j \rangle. \tag{3.12}$$

Due to the fact that $j \neq i$ and S_i signals are an orthogonal set, $C_{i,j}^{(1)} = 0$. In the next step we are going to compute $C_{i,j}^{(2)}$ as follows

$$\begin{aligned} C_{i,j}^{(2)} = \int_0^{T_s} &\left[\sum_{k=0}^{L-1} a_k \sqrt{\frac{2\mathcal{E}}{T_s}} cos(2\pi\left(f_0 + (i-1)\Delta f\right)t) \sin\left(2\pi\left(f_0 + (i-1)\Delta f\right)k\tau\right)\right] \times \\ &\sqrt{\frac{2}{T_s}} \sin\left(2\pi\left(f_0 + (j-1)\Delta f\right)t\right) dt. \end{aligned} \tag{3.13}$$

Since $\sin(\alpha)\cos(\beta) = \frac{1}{2}(\sin(\alpha+\beta) + \sin(\alpha-\beta))$, then the above equation can be simplified as follows:

$$C_{i,j}^{(2)} = \sum_{k=0}^{L-1} a_k \frac{2}{T_s}\sqrt{\mathcal{E}} \sin(2\pi(f_0 + (i-1)\Delta f) k\tau) \times \\ \int_0^{T_s} \left[\frac{\sin(2\pi(2f_0 + (i+j-2)\Delta f)t) + \sin(2\pi(j-i)\Delta f t)}{2}\right] dt. \tag{3.14}$$

Let us indicate the integral term of the above equation by W. Therefore, W can be simplified and calculated as follows

$$W = \left[\frac{-\cos(2\pi(2f_0 + (i+j-2)\Delta f)t)}{4\pi(2f_0 + (i+j-2)\Delta f)} + \frac{-\cos(2\pi(j-i)\Delta f t)}{4\pi(j-i)\Delta f}\right]_0^{T_s}. \tag{3.15}$$

The first term in (3.15) contains f_0 in denominator, which is a big number compared to the cosine function in nominator. Thus, it will be negligible and we have

$$W \cong \frac{1 - \cos(2\pi(j-i)\Delta f T_s)}{4\pi(j-i)\Delta f}. \tag{3.16}$$

As it was stated in (2.4), $\Delta f = \frac{1}{2T_s}$. Therefore, W would become

$$W = \frac{(1 - \cos(\pi(j-i)))T_s}{2\pi(j-i)} = \frac{\left(1 + (-1)^{(j-i+1)}\right)T_s}{2\pi(j-i)}. \tag{3.17}$$

Finally, $C_{i,j}$ for $j \neq i$ would be calculated by

$$C_{i,j} = -C_{i,j}^{(2)} = -\left(\frac{1 + (-1)^{(j-i+1)}}{\pi(j-i)}\right) \sum_{k=0}^{L-1} a_k \sqrt{\mathcal{E}} \sin(2\pi(f_0 + (i-1)\Delta f) k\tau). \tag{3.18}$$

The next step is to compute the projection of S_i' on its own (ith) dimension. Based on equation (3.10) we have:

$$C_{i,i} = \int_0^{T_s} S_{i_1}'(t)\psi_i(t)\,dt - \int_0^{T_s} S_{i_2}'(t)\psi_i(t)\,dt = C_{i,i}^{(1)} - C_{i,i}^{(2)}, \tag{3.19}$$

where $C_{i,i}^{(1)}$ has the same form as (3.11). If j is replaced with i, we get the following

$$C_{i,i}^{(1)} = \sum_{k=0}^{L-1} a_k \sqrt{\frac{1}{\mathcal{E}}} \cos(2\pi(f_0 + (i-1)\Delta f) k\tau) \langle S_i, S_i\rangle \\ = \sum_{k=0}^{L-1} a_k \sqrt{\mathcal{E}} \cos(2\pi(f_0 + (i-1)\Delta f) k\tau). \tag{3.20}$$

$C_{i,i}^{(2)}$ also has the same form as equation (3.13). Therefore, if j is replaced with i in (3.13) and we use the formula $\sin(\alpha)\cos(\alpha) = \frac{1}{2}\sin(2\alpha)$, $C_{i,i}^{(2)}$ can be calculated as

$$\begin{aligned} C_{i,i}^{(2)} &= \sum_{k=0}^{L-1} a_k \frac{2}{T_s}\sqrt{\mathcal{E}} \sin\left(2\pi\left(f_0 + (i-1)\Delta f\right)k\tau\right) \int_0^{T_s} \frac{1}{2}\sin\left(4\pi\left(f_0 + (i-1)\Delta f\right)t\right)dt \\ &= \sum_{k=0}^{L-1} a_k \frac{1}{T_s}\sqrt{\mathcal{E}} \sin\left(2\pi\left(f_0 + (i-1)\Delta f\right)k\tau\right) \left[\frac{-\cos\left(4\pi\left(f_0 + (i-1)\Delta f\right)t\right)}{4\pi\left(f_0 + (i-1)\Delta f\right)}\right]_0^{T_s}. \end{aligned}$$

$C_{i,i}^{(2)}$ has f_0 in its denominator and can be ignored. Therefore, for each value of i we have

$$C_{i,i} = C_{i,i}^{(1)} - C_{i,i}^{(2)} = C_{i,i}^{(1)}. \tag{3.21}$$

The set of reshaped basis vectors will be

$$S^{'} = \begin{bmatrix} S_1^{'} \\ S_2^{'} \\ S_3^{'} \\ \vdots \\ S_M^{'} \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & .... & C_{1,M} \\ C_{2,1} & C_{2,2} & C_{2,3} & .... & C_{2,M} \\ C_{3,1} & C_{3,2} & C_{3,3} & .... & C_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M,1} & C_{M,2} & C_{M,3} & .... & C_{M,M} \end{bmatrix}. \tag{3.22}$$

After projecting the reshaped basis signal on each dimension using the above formulas, noise has to be projected on each dimension as well. In other words, we have

$$n = \left(n_1, n_2, ..., n_M\right). \tag{3.23}$$

where n_i is the projection of noise signal on ith basis.

The decoding is based on APPs, which in turn depend on the distance of the vector representation of the received signal from reshaped bases in 3.22. The demodulator chooses the closest basis vector to the received signal. Therefore, the probability of correctly demodulating the received signal is

$$P_c = \Pr\left(|S_i^r - S_i^{'}|^2 < |S_i^r - S_1^{'}|^2, ..., |S_i^r - S_i^{'}|^2 < |S_i^r - S_M^{'}|^2 \mid S_i\left(t\right) \text{was sent}\right), \tag{3.24}$$

or equivalently

$$P_c = \prod_{\substack{k=1 \\ k\neq i}}^{M} \Pr\left(|S_i^r - S_i^{'}|^2 < |S_i^r - S_k^{'}|^2 \mid S_i\left(t\right) \text{was sent}\right) = \prod_{\substack{k=1 \\ k\neq i}}^{M} P_c^k. \tag{3.25}$$

Let us compute P_c^k

$$\begin{aligned} P_c^k &= \Pr\left(|S_i^r - S_i^{'}|^2 < |S_i^r - S_k^{'}|^2 \mid S_i\left(t\right) \text{was sent}\right) \\ &= \Pr\left(|S_i^{'} + n - S_i^{'}|^2 < |S_i^{'} + n - S_k^{'}|^2\right). \end{aligned}$$

P_c^k expands to

$$\begin{aligned} P_c^k &= \Pr\left(|\left(n_1, ..., n_M\right)|^2 < |\left(C_{i,1} - C_{k,1} + n_1, ..., C_{i,M} - C_{k,M} + n_M\right)|^2\right) \\ &= \Pr\left(\sum_{j=1}^{M} n_j^2 < \sum_{j=1}^{M} \left(C_{i,j} - C_{k,j} + n_j\right)^2\right). \end{aligned} \tag{3.26}$$

or

$$\begin{aligned} P_c^k &= \Pr\left(\sum_{j=1}^{M} n_j^2 < \sum_{j=1}^{M} \left(C_{i,j} - C_{k,j}\right)^2 + 2n_j\left(C_{i,j} - C_{k,j}\right) + n_j^2\right) \\ &= \Pr\left(-\sum_{j=1}^{M} \left(C_{i,j} - C_{k,j}\right)^2 < \sum_{j=1}^{M} 2n_j\left(C_{i,j} - C_{k,j}\right)\right). \end{aligned} \tag{3.27}$$

The right-hand side of the inequality in 3.27 is a summation of M independent, Gaussian random variables with zero mean and variance of N_0. Thus, the whole summation is equal to a Gaussian random variable with the following distribution

$$\sum_{j=1}^{M} 2n_j\left(C_{i,j} - C_{k,j}\right) \sim N\left(0, 4N_0 \sum_{j=1}^{M} \left(C_{i,j} - C_{k,j}\right)^2\right). \tag{3.28}$$

Therefore, P_c^k can be expressed using Q-function, which is defined as follows

$$Q\left(x\right) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du. \tag{3.29}$$

Q-function is defined for Gaussian random variables with zero mean and unit variance, therefore its modified version must be used to find P_c^k

$$\begin{aligned} P_c^k &= Q\left(\frac{-\sum_{j=1}^{M}\left(C_{i,j} - C_{k,j}\right)^2}{\sqrt{4N_0 \sum_{j=1}^{M}\left(C_{i,j} - C_{k,j}\right)^2}}\right) \\ &= Q\left(-\frac{1}{2}\sqrt{\frac{\sum_{j=1}^{M}\left(C_{i,j} - C_{k,j}\right)^2}{N_0}}\right). \end{aligned} \tag{3.30}$$

Due to symmetric property of Q-function that states $Q\left(-x\right)=1-Q\left(x\right)$, we can say that

$$P_c^k=1-Q\left(\frac{1}{2}\sqrt{\frac{\sum_{j=1}^{M}\left(C_{i,j}-C_{k,j}\right)^2}{N_0}}\right). \tag{3.31}$$

Finally, the probability of error will become

$$P_e=1-P_c=1-\prod_{\substack{k=1\\k\neq i}}^{M}\left[1-Q\left(\frac{1}{2}\sqrt{\frac{\sum_{j=1}^{M}\left(C_{i,j}-C_{k,j}\right)^2}{N_0}}\right)\right]. \tag{3.32}$$

In what was discussed so far, the assumption is that the channel coefficients are known to the decoder. However, only an estimate of these coefficients can be extracted from the received signal. Thus, 3.2 would be rewritten as

$$\begin{aligned}S_i^r\left(t\right)&=S_i\left(t\right)*\left(\hat{h}\left(t\right)+\Delta h\left(t\right)\right)+n\left(t\right)\\&=S_i\left(t\right)*\hat{h}\left(t\right)+S_i\left(t\right)*\Delta h\left(t\right)+n\left(t\right)\\&=\hat{S}_i\left(t\right)+\Delta S_i\left(t\right)+n\left(t\right),\end{aligned}$$

where $\hat{S}_i\left(t\right)$ and $\Delta S_i\left(t\right)$ have the same form as (3.4) except that a_k is replaced with $\hat{a}_k$ and Δa_k, respectively. Therefore,

$$\hat{S}_i\left(t\right)=\sum_{k=0}^{L-1}\hat{a}_kS_i\left(t-k\tau\right) \tag{3.33}$$

$$\Delta S_i\left(t\right)=\sum_{k=0}^{L-1}\Delta a_kS_i\left(t-k\tau\right). \tag{3.34}$$

The projection of these two signals onto $\{\psi_1,...,\psi_M\}$ would result in a formula like (3.18) for $j\neq i$ and in a formula like (3.20) for $i=j$. Therefore, we have the following equations

$$\begin{aligned}\hat{C}_{i,j}&=-\left(\frac{1+\left(-1\right)^{\left(j-i+1\right)}}{\pi\left(j-i\right)}\right)\sum_{k=0}^{L-1}\hat{a}_k\sqrt{\mathcal{E}}\sin\left(2\pi\left(f_0+\left(i-1\right)\Delta f\right)k\tau\right)\\\Delta C_{i,j}&=-\left(\frac{1+\left(-1\right)^{\left(j-i+1\right)}}{\pi\left(j-i\right)}\right)\sum_{k=0}^{L-1}\Delta a_k\sqrt{\mathcal{E}}\sin\left(2\pi\left(f_0+\left(i-1\right)\Delta f\right)k\tau\right)\end{aligned}$$

for $j \neq i$ and we have

$$\hat{C}_{i,i} = \sum_{k=0}^{L-1} \hat{a}_k \sqrt{\mathcal{E}} \cos\left(2\pi \left(f_0 + (i-1)\Delta f\right) k\tau\right)$$
$$\Delta C_{i,i} = \sum_{k=0}^{L-1} \Delta a_k \sqrt{\mathcal{E}} \cos\left(2\pi \left(f_0 + (i-1)\Delta f\right) k\tau\right)$$

for $i = j$. The estimated reshaped basis vectors would become

$$\hat{S} = \begin{bmatrix} \hat{S}_1 \\ \hat{S}_2 \\ \hat{S}_3 \\ \vdots \\ \hat{S}_M \end{bmatrix} = \begin{bmatrix} \hat{C}_{1,1} & \hat{C}_{1,2} & \hat{C}_{1,3} & .... & \hat{C}_{1,M} \\ \hat{C}_{2,1} & \hat{C}_{2,2} & \hat{C}_{2,3} & .... & \hat{C}_{2,M} \\ \hat{C}_{3,1} & \hat{C}_{3,2} & \hat{C}_{3,3} & .... & \hat{C}_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{M,1} & \hat{C}_{M,2} & \hat{C}_{M,3} & .... & \hat{C}_{M,M} \end{bmatrix}. \tag{3.35}$$

The projection of noise onto $\{\psi_1, ..., \psi_M\}$ would result in M independent Gaussian noise elements like (3.23).

Assume that $\hat{S}_i$, $i \in \{1, 2, ..., M\}$ are known to the decoder and APPs are measured with respect to them. Demodulator chooses the closest basis vector to the received signal. Thus,

$$\begin{aligned} P_c &= \prod_{\substack{k=1 \\ k\neq i}}^{M} \Pr\left(|S_i^r - \hat{S}_i|^2 < |S_i^r - \hat{S}_k|^2 \mid S_i(t) \text{ was sent}\right) \\ &= \prod_{\substack{k=1 \\ k\neq i}}^{M} P_c^k. \end{aligned}$$

We begin by computing P_c^k

$$\begin{aligned} P_c^k &= \Pr\left(|\Delta S_i + n|^2 < |\hat{S}_i - \hat{S}_k + \Delta S_i + n|^2\right) \\ &= \Pr\left(\sum_{j=1}^{M} (n_j + \Delta C_{i,j})^2 < \sum_{j=1}^{M} \left(\hat{C}_{i,j} - \hat{C}_{k,j} + n_j + \Delta C_{i,j}\right)^2\right) \\ &= \Pr\left(\sum_{j=1}^{M} (n_j + \Delta C_{i,j})^2 < \sum_{j=1}^{M} \left(\hat{C}_{i,j} - \hat{C}_{k,j}\right)^2 + (n_j + \Delta C_{i,j})^2 + 2\left(n_j + \Delta C_{i,j}\right)\left(\hat{C}_{i,j} - \hat{C}_{k,j}\right)\right) \end{aligned} \tag{3.36}$$

or equivalently

$$P_c^k = \Pr\left(-\sum_{j=1}^{M}\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2 < \sum_{j=1}^{M} 2\left(n_j+\Delta C_{i,j}\right)\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)\right). \tag{3.37}$$

The distribution of $\Delta C_{i,j}$s are not known. In the worst case scenario, they have independent, Gaussian distributions with zero mean and variance $\nu_{i,j}$. In that case we have

$$\sum_{j=1}^{M} 2\left(n_j+\Delta C_{i,j}\right)\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right) \sim N\left(0, 4\sum_{j=1}^{M}\left(N_0+\nu_{i,j}\right)\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2\right). \tag{3.38}$$

As a result, P_c^k can be expressed by Q-function as follows

$$\begin{aligned} P_c^k &\geq Q\left(\frac{-\sum_{j=1}^{M}\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2}{2\sqrt{\sum_{j=1}^{M}\left(N_0+\nu_{i,j}\right)\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2}}\right) \\ &= 1-Q\left(\frac{\sum_{j=1}^{M}\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2}{2\sqrt{\sum_{j=1}^{M}\left(N_0+\nu_{i,j}\right)\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2}}\right), \end{aligned} \tag{3.39}$$

and consequently P_e would be

$$P_e = 1-P_c \leq 1-\prod_{\substack{k=1\\k\neq i}}^{M}\left[1-Q\left(\frac{\sum_{j=1}^{M}\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2}{2\sqrt{\sum_{j=1}^{M}\left(N_0+\nu_{i,j}\right)\left(\hat{C}_{i,j}-\hat{C}_{k,j}\right)^2}}\right)\right]. \tag{3.40}$$

3.2 Channel Estimation

The previous section provided the error probability for our M-FSK modulated system when the channel is known and when we estimate the channel from received signals. In this section, the main objective is to find a method to estimate the channel from received signals. This estimation would enable us to find reshaped basis signals and therefore, decrease the error probability.

In the first iteration of decoding, we do not have any knowledge about the channel or reshaped bases. Therefore, we have to use original bases of 3.1 to demodulate and decode received signals. Next, we define two $M \times M$ matrices R^* and R.

Matrix R^* is our summation matrix. Vector representation of received signals that are demodulated as S_i are added together and put in the ith row of R^*. This procedure is done for all of received vectors. We fill R^* with these summations. Let the number of received signals demodulated as S_i be α_i where $\alpha_i \in \{0, 1, ..., T\}$. T is the number of symbols in each frame of product code. We keep these numbers in a vector $\{\alpha_1, ..., \alpha_M\}$. Matrix R^* is represented as follows

$$R^* = \begin{bmatrix} R_1^* \\ R_2^* \\ R_3^* \\ \vdots \\ R_M^* \end{bmatrix} = \begin{bmatrix} R_{1,1}^* & R_{1,2}^* & R_{1,3}^* & .... & R_{1,M}^* \\ R_{2,1}^* & R_{2,2}^* & R_{2,3}^* & .... & R_{2,M}^* \\ R_{3,1}^* & R_{3,2}^* & R_{3,3}^* & .... & R_{3,M}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{M,1}^* & R_{M,2}^* & R_{M,3}^* & .... & R_{M,M}^* \end{bmatrix}. \tag{3.41}$$

Matrix R is our mean matrix. The ith row of R is the result of dividing the ith row of R^* by α_i. Matrix R can be represented as follows

$$R = \begin{bmatrix} R_1^{(1)} \\ R_2^{(1)} \\ R_3^{(1)} \\ \vdots \\ R_M^{(1)} \end{bmatrix} = \begin{bmatrix} R_{1,1}^{(1)} & R_{1,2}^{(1)} & R_{1,3}^{(1)} & .... & R_{1,M}^{(1)} \\ R_{2,1}^{(1)} & R_{2,2}^{(1)} & R_{2,3}^{(1)} & .... & R_{2,M}^{(1)} \\ R_{3,1}^{(1)} & R_{3,2}^{(1)} & R_{3,3}^{(1)} & .... & R_{3,M}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{M,1}^{(1)} & R_{M,2}^{(1)} & R_{M,3}^{(1)} & .... & R_{M,M}^{(1)} \end{bmatrix}. \tag{3.42}$$

Matrix R is now an estimation of matrix S' in 3.22. Consequently, each element of R is an estimate of its corresponding element in S'. Therefore, we try to find the relation between $R_{i,j}^{(1)}$ and $C_{i,j}$. $R_{i,j}^{(1)}$ is the summation of $C_{i,j}$ and a noise $n_{i,j}$ which can be represented as follows

$$R_{i,j}^{(1)} = C_{i,j} + n_{i,j}, \tag{3.43}$$

where $n_{i,j}$ is the mean of the projection of α_i noises onto ψ_j, or equivalently

$$n_{i,j} = \frac{\sum_{d=1}^{\alpha_i} n_j^d}{\alpha_i}.$$

Furthermore, $C_{i,j}$ can be written as a linear combination of a_k, $k \in \{0, ..., L-1\}$ like what follows

$$C_{i,j} = \sum_{k=0}^{L-1} a_k \zeta_{i,j,k},$$

where $\zeta_{i,j,k}$ is

$$\zeta_{i,j,k} = \begin{cases} -\left(\frac{1+(-1)^{(j-i+1)}}{\pi(j-i)}\right)\sqrt{\mathcal{E}}\sin\left(2\pi\left(f_0+(i-1)\,\Delta f\right)k\tau\right) & \text{when } j \neq i \\ \sqrt{\mathcal{E}}\cos\left(2\pi\left(f_0+(i-1)\,\Delta f\right)k\tau\right) & \text{when } j \neq i, \end{cases} \tag{3.44}$$

Therefore, $R^{(1)}_{i,j}$ will be represented as

$$R^{(1)}_{i,j} = \sum_{k=0}^{L-1} a_k \zeta_{i,j,k} + n_{i,j}. \tag{3.45}$$

The goal is to find a_k where $k \in \{0, 1, ..., L-1\}$, thus the estimation of channel must has at least L taps to correctly portray the channel. The number of taps are assumed to be N where $N \geq L$, in which the first L terms are estimate of channel taps and the rest have very small values.

From (3.18) and (3.20), we know that only some of $R^{(1)}_{i,j}$s are non-zero. These non-zero elements are put in an $\eta \times 1$ matix

$$R^{(1)} = \begin{bmatrix} R^{(1)}_{1,1} \\ R^{(1)}_{1,2} \\ R^{(1)}_{1,4} \\ \vdots \\ R^{(1)}_{M,M} \end{bmatrix}_{\eta\times 1}. \tag{3.46}$$

where η is the number of non-zero elements calculated by finding the number of (i, j) tuples that satisfy $\zeta_{i,j,k} = 0$. Calculated values for η will be

$$\eta = \begin{cases} \frac{M(M+1)}{2} & \text{when M is odd} \\ \frac{M(M+2)}{2} & \text{when M is even.} \end{cases} \tag{3.47}$$

$C_{i,j}$ can be shown as the multiplication of two matrices. One of these matrices is an $N \times 1$ matrix which contains a_k or channel taps

$$A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_N \end{bmatrix}_{N\times 1}. \tag{3.48}$$

The other one is an $\eta \times N$ matrix which contains $\zeta_{i,j,k}$

$$\zeta = \begin{bmatrix} \zeta_{1,1,1} & \zeta_{1,1,2} & \zeta_{1,1,3} & \cdots & \zeta_{1,1,N} \\ \zeta_{1,2,1} & \zeta_{1,2,2} & \zeta_{1,2,3} & \cdots & \zeta_{1,2,N} \\ \zeta_{1,4,1} & \zeta_{1,4,2} & \zeta_{1,4,3} & \cdots & \zeta_{1,4,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \zeta_{M,M,1} & \zeta_{M,M,2} & \zeta_{M,M,3} & \cdots & \zeta_{M,M,N} \end{bmatrix}_{\eta \times N} . \tag{3.49}$$

Moreover, the noise of each term of $R_{i,j}$ is also put in an $\eta \times 1$ matrix

$$n = \begin{bmatrix} n_{1,1} \\ n_{1,2} \\ n_{1,4} \\ \vdots \\ n_{M,M} \end{bmatrix}_{\eta \times 1} . \tag{3.50}$$

Finally, channel taps can be found by solving the following matrix equation

$$R^{(1)}_{\eta \times 1} = \zeta_{\eta \times N} A_{N \times 1} + n_{\eta \times 1}. \tag{3.51}$$

It is not possible to simply multiply both sides by ζ^{-1}, because ζ is not a square matrix and thus not invertible. Referring to [22], the LMMSE estimation of the channel would be computed as:

$$\zeta^H R^{(1)} = \zeta^H \zeta A + \zeta^H n \tag{3.52}$$

where ζ^H is the Hermition transpose of H. Thus, A would be

$$A_{N \times 1} = \left(\zeta^H \zeta\right)^{-1} \zeta^H R^{(1)} - \left(\zeta^H \zeta\right)^{-1} \zeta^H n, \tag{3.53}$$

where $\left(\zeta^H \zeta\right)^{-1} \zeta^H n$ shows the estimator error which was caused by noise. Therefore, $\hat{A}^{(1)}$ which is a matrix with elements that represent estimated channel taps is computed by

$$\hat{A}^{(1)} = \left(\zeta^H \zeta\right)^{-1} \zeta^H R^{(1)}. \tag{3.54}$$

Referring to equations (3.18) and (3.20), our initial estimation of reshaped bases $\{\hat{S}^{(1)}_1, \hat{S}^{(1)}_2, ..., \hat{S}^{(1)}_M\}$ can be formed.

In the second iteration, the decoder uses $\{\hat{S}^{(1)}_1, \hat{S}^{(1)}_2, ..., \hat{S}^{(1)}_M\}$ to demodulate and decode received signals. Like the first iteration, matrices R^* and R can be formed. The new R contains $R^{(2)}_{i,j}$ as its element. $R^{(2)}_{i,j}$ has the following relation with $C_{i,j}$

$$R^{(2)}_{i,j} = C_{i,j} + n^{'}_{i,j} = \sum_{k=0}^{L-1} a_k \zeta_{i,j,k} + n^{'}_{i,j} \tag{3.55}$$

Then, matrix $R^{(2)}$ is formed by replacing each $R^{(1)}_{i,j}$ with $R^{(2)}_{i,j}$ in 3.46

$$R^{(2)} = \begin{bmatrix} R^{(2)}_{1,1} \\ R^{(2)}_{1,2} \\ R^{(2)}_{1,4} \\ \vdots \\ R^{(2)}_{M,M} \end{bmatrix}_{\eta \times 1} . \tag{3.56}$$

Finally, $\hat{A}^{(2)}$ would be calculated like (3.54) in the following manner

$$\hat{A}^{(2)} = \left(\zeta^H \zeta\right)^{-1} \zeta^H R^{(2)}. \tag{3.57}$$

$\hat{A}^{(2)}$ is the second estimation of the channel. This procedure can go on for a sufficiently large iteration $(n = 2, 3)$.

3.3 Relation Between Channel Estimation Error and SER

In the previous section, we propose our method to estimate the channel. In our calculations we assumed that all of received signals are demodulated correctly. However, we know that this is not the case and our initial demodulation and decoding has error probability of $P_e^{(1)}$. If $P_e^{(1)}$ is too big, then our first channel estimation will be misleading. This causes our method to diverge from the real channel and the error probability will converge to one. Therefore, we have to consider this $P_e^{(1)}$ in our calculations.

Referring to 3.41, the summation matrix R^* can be divided into two parts. One part is the summation of the vector representation of received signals that are correctly decoded (R^*_C). Another part is the summation of the vector representation of received signals that are incorrectly decoded (R^*_I). Therefore, R^* can be represented as

$$R^* = R^*_C + R^*_I. \tag{3.58}$$

The matrix R_C^* can be written as

$$R_C^* = \begin{bmatrix} \alpha_1 S_1^{'} \\ \alpha_2 S_2^{'} \\ \alpha_3 S_3^{'} \\ \vdots \\ \alpha_M S_M^{'} \end{bmatrix} + \begin{bmatrix} \alpha_1 n_1 \\ \alpha_2 n_2 \\ \alpha_3 n_3 \\ \vdots \\ \alpha_M n_M \end{bmatrix} = \begin{bmatrix} \alpha_1 C_{1,1} & \alpha_1 C_{1,2} & \alpha_1 C_{1,3} & .... & \alpha_1 C_{1,M} \\ \alpha_2 C_{2,1} & \alpha_2 C_{2,2} & \alpha_2 C_{2,3} & .... & \alpha_2 C_{2,M} \\ \alpha_3 C_{3,1} & \alpha_3 C_{3,2} & \alpha_3 C_{3,3} & .... & \alpha_3 C_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_M C_{M,1} & \alpha_M C_{M,2} & \alpha_M C_{M,3} & .... & \alpha_M C_{M,M} \end{bmatrix}$$
$$+ \begin{bmatrix} \alpha_1 n_{1,1} & \alpha_1 n_{1,2} & \alpha_1 n_{1,3} & .... & \alpha_1 n_{1,M} \\ \alpha_2 n_{2,1} & \alpha_2 n_{2,2} & \alpha_2 n_{2,3} & .... & \alpha_2 n_{2,M} \\ \alpha_3 n_{3,1} & \alpha_3 n_{3,2} & \alpha_3 n_{3,3} & .... & \alpha_3 n_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_M n_{M,1} & \alpha_M n_{M,2} & \alpha_M n_{M,3} & .... & \alpha_M n_{M,M} \end{bmatrix}, \tag{3.59}$$

where α_i is the number of correctly decoded S_i symbols. Moreover, $\alpha_i n_{i,j}$ is the summation of the projection of α_i noises onto ψ_j.

The matrix R_I^* can be written as

$$R_I^* = \begin{bmatrix} \sum_{u=2}^{M} \beta_1^u S_u^{'} \\ \\ \sum_{\substack{u=1 \\ u\neq 2}}^{M} \beta_2^u S_u^{'} \\ \\ \sum_{\substack{u=1 \\ u\neq 3}}^{M} \beta_3^u S_u^{'} \\ \\ \vdots \\ \\ \sum_{u=1}^{M-1} \beta_M^u S_u^{'} \end{bmatrix} + \begin{bmatrix} \beta_1 n_1^{'} \\ \beta_2 n_2^{'} \\ \beta_3 n_3^{'} \\ \vdots \\ \beta_M n_M^{'} \end{bmatrix}, \tag{3.60}$$

or equivalently

$$R_I^* = \begin{bmatrix} \beta_1 n_{1,1} + \sum_{u=2}^{M} \beta_1^u C_{u,1} & \beta_1 n_{1,2} + \sum_{u=2}^{M} \beta_1^u C_{u,2} & .... & \beta_1 n_{1,M} + \sum_{u=2}^{M} \beta_1^u C_{u,M} \\ \beta_2 n_{2,1} + \sum_{\substack{u=1\\u\neq 2}}^{M} \beta_2 C_{u,1} & \beta_2 n_{2,2} + \sum_{\substack{u=1\\u\neq 2}}^{M} \beta_2 C_{u,2} & .... & \beta_2 n_{2,M} + \sum_{\substack{u=1\\u\neq 2}}^{M} \beta_2 C_{u,M} \\ \beta_3 n_{3,1} + \sum_{\substack{u=1\\u\neq 3}}^{M} \beta_3 C_{u,1} & \beta_3 n_{3,2} + \sum_{\substack{u=1\\u\neq 3}}^{M} \beta_3 C_{u,2} & .... & \beta_3 n_{3,M} + \sum_{\substack{u=1\\u\neq 3}}^{M} \beta_3 C_{u,M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_M n_{M,1} + \sum_{u=1}^{M-1} \beta_M C_{u,1} & \beta_M n_{M,2} + \sum_{u=1}^{M-1} \beta_M C_{u,2} & .... & \beta_M n_{M,M} + \sum_{u=1}^{M-1} \beta_M C_{u,M} \end{bmatrix}.$$

β_i^u is the number of times that symbol S_u was sent but mistakenly demodulated as S_i, and β_i is

$$\beta_i = \sum_{\substack{u=1\\u\neq i}}^{M} \beta_i^u. \tag{3.61}$$

$\beta_i n_{i,j}$ is the summation of the projection of β_i noises onto ψ_j.

Referring to 3.42, the mean matrix R can be calculated. Each row of R is the mean of the corresponding row in R^*

$$R = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_M \end{bmatrix}_{M\times M} = \begin{bmatrix} \frac{R_1^*}{\alpha_1+\beta_1} \\ \frac{R_2^*}{\alpha_2+\beta_2} \\ \frac{R_3^*}{\alpha_3+\beta_3} \\ \vdots \\ \frac{R_M^*}{\alpha_M+\beta_M} \end{bmatrix} = \begin{bmatrix} \frac{R_{C_1}^*+R_{I_1}^*}{\alpha_1+\beta_1} \\ \frac{R_{C_2}^*+R_{I_2}^*}{\alpha_2+\beta_2} \\ \frac{R_{C_3}^*+R_{I_3}^*}{\alpha_3+\beta_3} \\ \vdots \\ \frac{R_{C_M}^*+R_{I_M}^*}{\alpha_M+\beta_M} \end{bmatrix} = R_C + R_I, \tag{3.62}$$

or equivalently

$$R = R_C + R_I = \begin{bmatrix} \frac{\alpha_1}{\alpha_1+\beta_1} S_1^{'} \\ \frac{\alpha_2}{\alpha_2+\beta_2} S_2^{'} \\ \frac{\alpha_3}{\alpha_3+\beta_3} S_3^{'} \\ \vdots \\ \frac{\alpha_M}{\alpha_M+\beta_M} S_M^{'} \end{bmatrix} + \begin{bmatrix} \frac{1}{\alpha_1+\beta_1} \sum_{u=2}^{M} \beta_1^u S_u^{'} \\ \frac{1}{\alpha_2+\beta_2} \sum_{\substack{u=1 \\ u\neq 2}}^{M} \beta_2^u S_u^{'} \\ \frac{1}{\alpha_3+\beta_3} \sum_{\substack{u=1 \\ u\neq 3}}^{M} \beta_3^u S_u^{'} \\ \vdots \\ \frac{1}{\alpha_M+\beta_M} \sum_{u=1}^{M-1} \beta_M^u S_u^{'} \end{bmatrix} + \begin{bmatrix} \frac{\alpha_1 n_1+\beta_1 n_1^{'}}{\alpha_1+\beta_1} \\ \frac{\alpha_2 n_2+\beta_2 n_2^{'}}{\alpha_2+\beta_2} \\ \frac{\alpha_3 n_3+\beta_3 n_3^{'}}{\alpha_3+\beta_3} \\ \vdots \\ \frac{\alpha_M n_M+\beta_M n_M^{'}}{\alpha_M+\beta_M} \end{bmatrix}. \tag{3.63}$$

Each $R_{i,j}^{(1)}$ can be demonstrated as the summation of $C_{i,j}$ and noise like what follows

$$R_{i,j}^{(1)} = R_{C_{i,j}}^{(1)} + R_{I_{i,j}}^{(1)} = C_{i,j} + n_{i,j}. \tag{3.64}$$

The non-zero elements of R are chosen and put in a new $\eta \times 1$ matrix

$$R^{(1)} = R_C^{(1)} + R_I^{(1)} = \begin{bmatrix} R_{1,1}^{(1)} \\ R_{1,2}^{(1)} \\ R_{1,4}^{(1)} \\ \vdots \\ R_{M,M}^{(1)} \end{bmatrix}_{\eta\times 1} = \begin{bmatrix} R_{C_{1,1}}^{(1)} + R_{I_{1,1}}^{(1)} \\ R_{C_{1,2}}^{(1)} + R_{I_{1,2}}^{(1)} \\ R_{C_{1,4}}^{(1)} + R_{I_{1,4}}^{(1)} \\ \vdots \\ R_{C_{1,M}}^{(1)} + R_{I_{1,M}}^{(1)} \end{bmatrix}. \tag{3.65}$$

$R^{(1)}$ can be put in a matrix equation like 3.51. Therefore, we have

$$R^{(1)} = R_C^{(1)} + R_I^{(1)} = \zeta A + n \tag{3.66}$$

where $\zeta = [\zeta_{i,j,k}]_{\eta\times N}$. $\zeta_{i,j,k}$ can be computed based on 3.44. Referring to [22], the LMMSE estimation of channel would be computed as:

$$A = \left(\zeta^H \zeta\right)^{-1} \zeta^H R_C^{(1)} + \left(\zeta^H \zeta\right)^{-1} \zeta^H R_I^{(1)} - \left(\zeta^H \zeta\right)^{-1} \zeta^H n, \tag{3.67}$$

and our channel estimation would be

$$\hat{A} = \left(\zeta^H \zeta\right)^{-1} \zeta^H R_C^{(1)} + \left(\zeta^H \zeta\right)^{-1} \zeta^H R_I^{(1)} = \left(\zeta^H \zeta\right)^{-1} \zeta^H R^{(1)}. \tag{3.68}$$

Therefore, the channel estimation error will be

$$E_{ce} = \left|A - \hat{A}\right| = \begin{bmatrix} |a_1 - \hat{a}_1| \\ |a_2 - \hat{a}_2| \\ |a_3 - \hat{a}_3| \\ \vdots \\ |a_N - \hat{a}_N| \end{bmatrix} = \left| \left(\zeta^H \zeta\right)^{-1} \zeta^H R_I^{(1)} + \left(\zeta^H \zeta\right)^{-1} \zeta^H n \right|, \tag{3.69}$$

which is caused by two different terms, $\left(\zeta^H \zeta\right)^{-1} \zeta^H R_I^{(i)}$ and $\left(\zeta^H \zeta\right)^{-1} \zeta^H n$. $\left(\zeta^H \zeta\right)^{-1} \zeta^H R_I^{(i)}$ can be reduced as the iterations goes on, but $\left(\zeta^H \zeta\right)^{-1} \zeta^H n$ will remain constant during the iterations.

The probability of error for this case is going to be computed based on equation 3.40. However, $\nu_{i,j}$ terms in 3.40, which are variances of errors in channel estimation, need to be calculated first.

Consider the case where the initial error probability is $P_e^{(1)}$. If the number of all the transmitted data are T_r, then the number of symbols that are mistakenly demodulated are $P_e^{(1)} T_r$.

For sake of simplicity, let us assume that each symbol has the same probability to be mistaken with other symbols. Thus, each one of $P_e^{(1)} T_r$ errors can be uniformly chosen from the following set of received matrices based on 3.41:

$$G = \{g_1^2, g_1^3, ..., g_1^M, g_2^1, ..., g_M^{M-1}\}, \tag{3.70}$$

where

$$g_i^j = R^* \left(S_j\left(t\right) \text{is demodulated} \mid S_i\left(t\right) \text{was sent}\right). \tag{3.71}$$

g_i^j can be shown in matrix form as

$$g_i^j = \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{i,1}^{'} & S_{i,2}^{'} & S_{i,3}^{'} & ... & S_{i,M}^{'} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & ... & 0 \end{bmatrix}. \tag{3.72}$$

where only the jth row is nonzero.

The probability of error (X) for each tuple (i, j) can be defined as follows

$$\Pr\left(X = g_i^j\right) = \frac{1}{M\left(M-1\right)} \qquad \text{for all i,j where } i \neq j. \tag{3.73}$$

The expected value of this random variable would be

$$E\left(X\right) = \sum_{i,j} XP\left(X\right) = \frac{1}{M\left(M-1\right)} \sum_{i,j} X, \tag{3.74}$$

or as defined in 3.72 as

$$E\left(X\right) = \frac{1}{M\left(M-1\right)} \begin{bmatrix} \sum_{u=2}^{M} S_u^{'} \\ \sum_{\substack{u=1 \\ u\neq 2}}^{M} S_u^{'} \\ \sum_{\substack{u=1 \\ u\neq 3}}^{M} S_u^{'} \\ \vdots \\ \sum_{u=1}^{M-1} S_u^{'} \end{bmatrix}. \tag{3.75}$$

In the same manner, $E\left(X^2\right)$ can be found to be

$$E\left(X^2\right) = \sum_{i,j} X^2 P\left(X\right) = \frac{1}{M\left(M-1\right)} \begin{bmatrix} \sum_{u=2}^{M} \left(S_u^{'}\right)^2 \\ \sum_{\substack{u=1 \\ u\neq 2}}^{M} \left(S_u^{'}\right)^2 \\ \sum_{\substack{u=1 \\ u\neq 3}}^{M} \left(S_u^{'}\right)^2 \\ \vdots \\ \sum_{u=1}^{M-1} \left(S_u^{'}\right)^2 \end{bmatrix}. \tag{3.76}$$

Therefore, variance of these errors will be

$$Var\left(X\right) = E\left(X^2\right) - E\left(X\right)^2 \tag{3.77}$$

$$Var\left(X\right)=\frac{1}{M\left(M-1\right)}\begin{bmatrix}\sum_{u=2}^{M}\left(S_u^{'}\right)^2\\ \sum_{\substack{u=1\\u\neq 2}}^{M}\left(S_u^{'}\right)^2\\ \sum_{\substack{u=1\\u\neq 3}}^{M}\left(S_u^{'}\right)^2\\ \vdots\\ \sum_{u=1}^{M-1}\left(S_u^{'}\right)^2\end{bmatrix}-\frac{1}{M^2\left(M-1\right)^2}\begin{bmatrix}\left(\sum_{u=2}^{M}S_u^{'}\right)^2\\ \left(\sum_{\substack{u=1\\u\neq 2}}^{M}S_u^{'}\right)^2\\ \left(\sum_{\substack{u=1\\u\neq 3}}^{M}S_u^{'}\right)^2\\ \vdots\\ \left(\sum_{u=1}^{M-1}S_u^{'}\right)^2\end{bmatrix} \quad (3.78)$$

This is the variance matrix for one error in estimating the channel, thus the variance matrix for all of these errors will be

$$\nu^{(1)}=P_e^{(1)}T_rVar\left(X\right), \quad (3.79)$$

because all errors are independent from each other. Matrix $\nu^{(1)}$ contains $\nu_{i,j},\ i,j\in\{1,2,...,M\}$ that are needed in 3.40. Utilizing this matrix, a new P_e can be computed. $P_e^{(2)}$ is the result of the following equation

$$P_e^{(2)}=f\left(P_e\right), \quad (3.80)$$

where $f\left(.\right)$ is the function that shows the effect of coding. With $P_e^{(2)}$ and a new estimate of reshaped basis $\hat{S}$, a new matrix for $\nu^{(2)}$ can be found. Based on the iterative method explained in the previous section, each time a lower error probability will be achieved. This procedure can be done n times to achieve the minimum possible error probability.

Chapter 4

Simulation and Observations

This chapter is dedicated to simulating blind channel estimation for M-FSK modulation systems with product code and its results. The simulation is done in MATLAB. At first, different parts of our simulation is explained. Then, the results for each scenario is presented as figures to give a better understanding of the simulation process.

4.1 Outline of the Simulation Process

The simulation process for our communication system is divided into three major parts. The first part is the simulation of transmitted signal blocks, which is shown as a block diagram in figure 4.1. This part consists of following steps

- *Step 1*: At first, the transmitted data needs to be generated. Therefore, based on k_i which is the number of symbols in each dimension and N which is the total number of dimensions, a frame of $k = k_1 \times k_2 \times ... \times k_N$ random number from $\{0, 1, ..., M-1\}$ is chosen. This frame is shown as K in the block diagram of figure 4.1. Each element of this frame represents one of the symbols in 2.1. For example, if the value of an element is 3, this means that this element represents S_4.
- *Step 2*: After generating transmission symbols, we have to code them. Product code is the channel coding scheme being used. In order to implement the coding process, we add a parity check symbol in each dimension. For instance, in the first dimension, elements in each row are added together. Then, the additive inverse of this summation in GF(M) is concatenated at the end of that row. The product coder block in 4.1 does

this procedure for each dimension to produce a $n = (k_1 + 1) \times (k_2 + 1) \times ... \times (k_N + 1)$ frame which is represented as N.

- *Step 3*: Referring to 2.1, each transmitted symbol that represents S_i can be modeled as a vector with M discrete values V_i e.g. $V_i = [0, ..., 1, ..., 0]$, where the ith element of V_i is one and the rest are zero. Therefore, to send a frame through channel, N vectors of length M are needed. The resulting block of vectors is represented by U in 4.1.

- *Step 4*: In step 3, each vector V_i represents symbol S_i in frequency domain. In order to get the corresponding time domain vector v_i, we get M-point $IFFT$ of V_i. We do this process for all of the vectors of a frame. The resulting transmitted signal block of time samples is represented by x in 4.1.

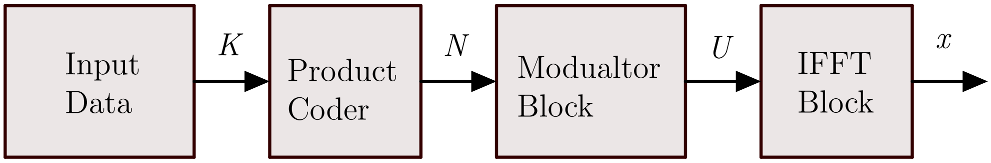

Figure 4.1: Block diagram of the transmitter

The second part is the simulation of the transmitted signal block passing through the channel. Block diagram of figure 4.2 shows this process. This process consist of following steps

- *Step 5*: As a transmitted signal block x goes through the channel, the time samples are convolved with a L tap channel impulse response represented by h. The resulting block is demonstrated as $x^{'}$ in figure 4.2.

- *Step 6*: Based on equation 2.10, an AWGN vector is added to $x^{'}$. This addition produce the received signal block y in 4.2.

The last part of the simulation is dedicated to decoding the received signal block y. Block diagram of this procedure is provided in figure 4.3. Following steps would explain the last part of simulation:

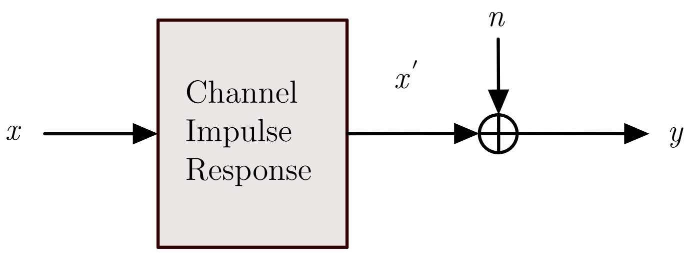

Figure 4.2: Block diagram of the channel

- *Step 7*: The received signal y is first applied to FFT block. This block's role is to first divide the received signal into vectors of length M. Then, it takes M-point FFT of each of these vectors to get a block of vectors in frequency domain. This frequency block is represented by $\hat{U}$ in figure 4.3.

- *Step 8*: BCJR decoder, which is also called MAP decoder, is a soft decoder that maximizes the a posteriori symbol probabilities. APPs need to be calculated based on formula 2.49. These APPs are calculated for each M sample vector and with respect to all M symbols possible. Resulting APPs are demonstrated by P in 4.3.

- *Step 9*: APPs are then fed into BCJR algorithm. This algorithm uses equation 2.54 in an iterative manner to produce decoded APPs which are represented by $\hat{P}$ in 4.3.

- *Step 10*: Based on equation 2.50, APPs are demodulated to one of the symbols $\{S_1, S_2, ..., S_M\}$. The resulting demodulated symbols can be put in a $n = (k_1 + 1) \times (k_2 + 1) \times ... \times (k_N + 1)$ output frame called Z^1.

- *Step 11*: Utilizing output frame Z^1, along with the formula 3.57, an estimate of the channel can be produced. This estimate $\hat{h}$ can has more taps than L while representing the original h closely.

- *Step 12*: Based on equations 3.18 and 3.20, reshaped basis vectors can be obtained by using the estimated channel. These reshaped bases are then put in a matrix like $\hat{S}$ in figure 4.3.

- *Step 13*: The APP calculator block of step 8 once again calculates APPs based on $\hat{U}$, $\hat{S}$, and 2.49.

- *Step 14*: Based on these new APPs, steps 9 till 13 are repeated again and again until the difference between two consecutive output frames is less than an specified ϵ.

$$Z^i - Z^{(i-1)} \leq \epsilon \tag{4.1}$$

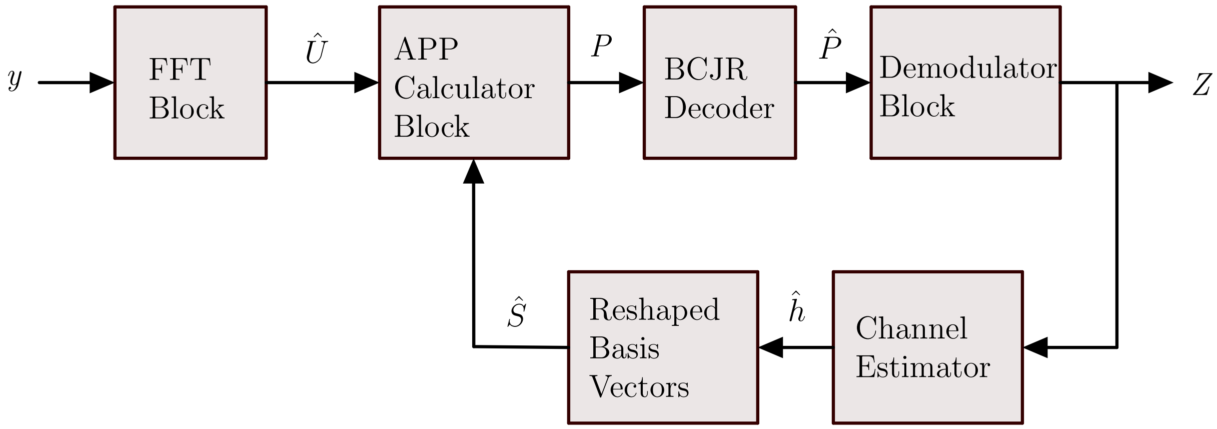

Figure 4.3: Block diagram of the receiver

4.2 Simulation Results

In results that follows, the number of the basis vectors M is set to be 64. Each frame of data contains 64 symbols in which they are distributed in a three-dimensional matrix of size $[4, 4, 4]$. Consequently, each frame of transmitted signals contain 125 symbols in which they are distributed in a $[5, 5, 5]$ matrix. Furthermore, from one transmission frame to the next, channel impulse response changes. Each tap of the channel impulse response changes based on the following equation

$$a_k^{t+1} = a_k^t + \rho \qquad \text{for every } 0 \leq k \leq L-1, \tag{4.2}$$

where a_k^t, $k \in \{0, 1, ..., L-1\}$ are channel coefficients at time t, a_k^{t+1}, $k \in \{0, 1, ..., L-1\}$ are channel coefficients at time $t + 1$, and ρ is a Gaussian random number that has the following distribution:

$$\rho \sim N(0, \delta) \tag{4.3}$$

with δ as its variance. However, the energy constraint of 2.12 is still in place, which means that

$$\sum_{k=0}^{L-1} \left(a_k^{t+1}\right)^2 = \sum_{k=0}^{L-1} \left(a_k^t\right)^2 = 1. \tag{4.4}$$

In one of the simulations, we change parameter δ to find the value of δ that limits our ability to estimate the channel, because as δ grows, each channel differs more from the previous one and the performance of estimation method degrades.

Another parameter is also considered in the simulations which shows how fast our algorithm is going to converge to its final value. How much each output frame Z^i is going to effect the previous channel estimation depends on parameter ϵ. The relationship among consecutive channel estimations, mean of the received vector, and epsilon is as follows

$$\hat{h}^{t+1} = \hat{h}^{t} - \epsilon \left(\hat{h}^{t} - \left(\zeta^H \zeta\right)^{-1} \zeta^H R^{t+1} \right), \tag{4.5}$$

where $\hat{h}^t$ is the estimated channel at time t, $\hat{h}^{t+1}$ is the estimated channel at time $t+1$, and $\left(\zeta^H \zeta\right)^{-1} \zeta^H R^{t+1}$ is the mean of received signal vector calculated based on matrix 3.42.

Different scenarios can be discussed by using different values for L (channel length), ϵ, and δ. At first, let us take a look at the effect of product code. In figure 4.4, two curves are presented. Both of these curves are assuming that channel is a delta function ($L = 1$, $\epsilon = 0$, $\delta = 0$):

$$h[n] = \delta[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases} \tag{4.6}$$

As can be seen, product code vastly improves the symbol error rate (or SER).

Figure 4.5 depicts the effect of channel impulse response on system's performance. For both of the curves in the figure 4.5, $\delta = 0$, which means that channel is not varying with time. For one of the curves it is assumed that the channel is a delta function. However, for the other curve, it is assumed that there is a channel impulse response with $L = 10$ that meets the condition of 2.12 and the decoder knows this channel impulse response. Therefore, there is no need for estimating the channel impulse response.

Based on 4.5, the effect of channel impulse response can be modeled as a reduction in SINR. In this case, this reduction in SINR is around 0.5 dB.

Next, we are going to explore the effect of the length of channel. In figures 4.6 - 4.9, it is assumed that channel is not varying with time or $\delta = 0$. In each one of these figures, three curves are presented. The blue solid line curve is the result of simulation when channel is unknown and there is no attempt to estimate it ($\epsilon = 0$). The red dashed line curve belongs to the case when channel is unknown but we try to estimate it and our estimation gets better as more data arrives ($\epsilon = 1$). And, the green dotted curve belongs to the case where we know the channel and thus there is no need for estimation ($\epsilon = 0$).

As the length of channel increases, a larger portion of the energy of each symbol leaks into next symbols. This increase in energy leakage would cause more ISI. Increase in ISI

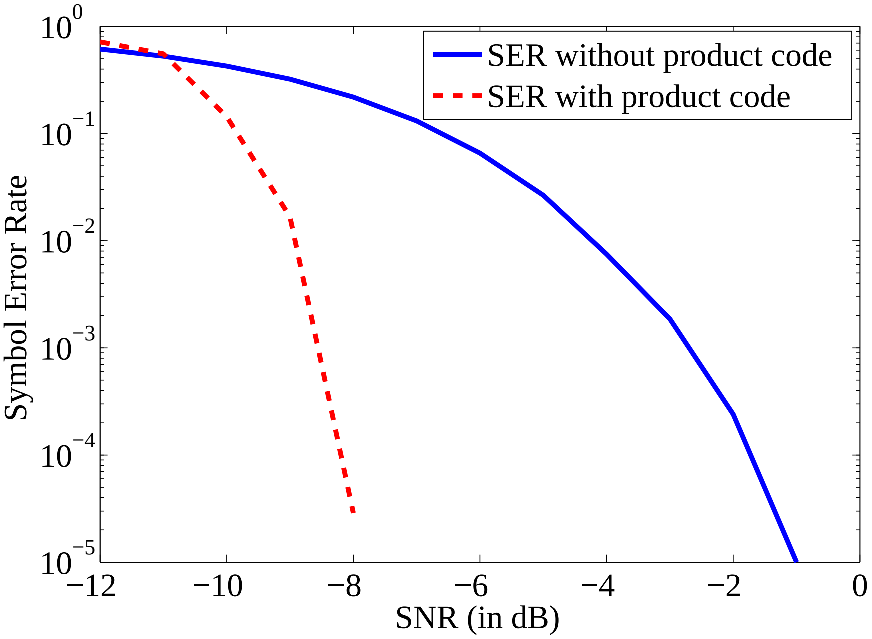

Figure 4.4: Effect of product code

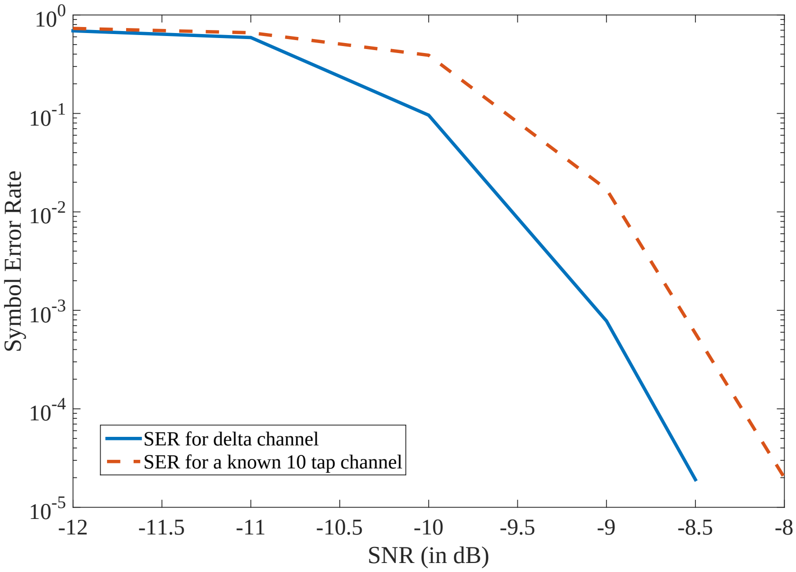

Figure 4.5: Effect of channel impulse response

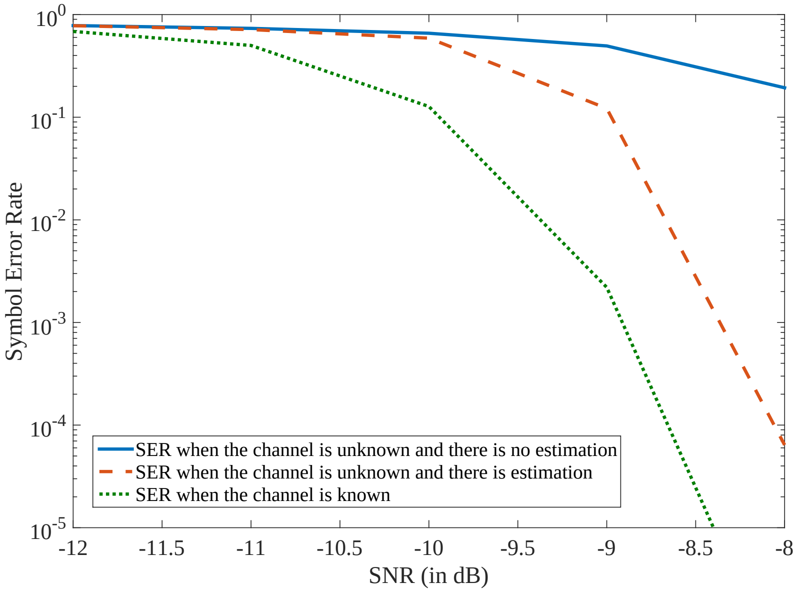

Figure 4.6: Effect of the channel length when L=10

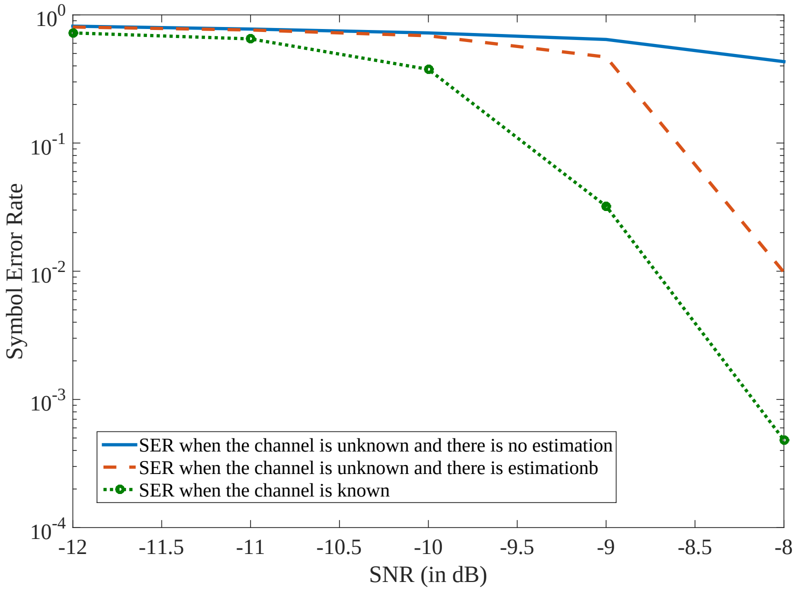

Figure 4.7: Effect of the channel length when L=40

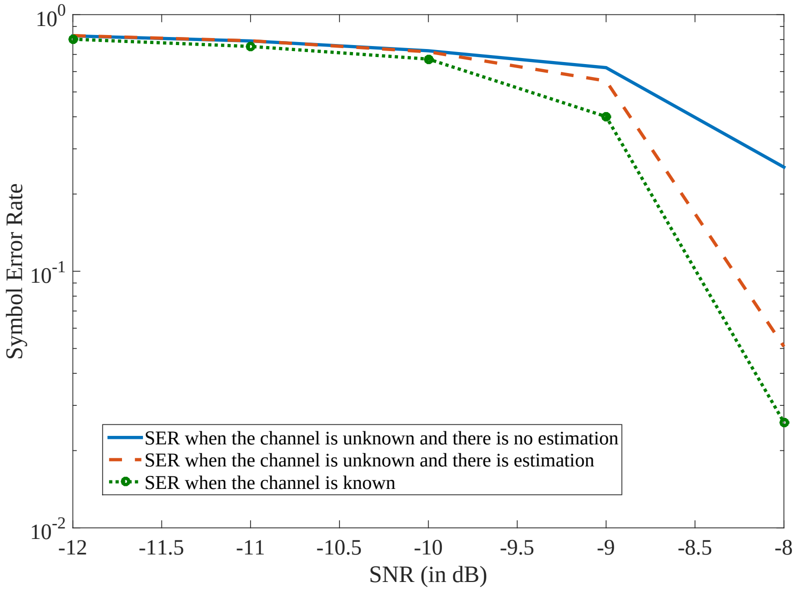

Figure 4.8: Effect of the channel length when L=160

would result in an increase in error probability. Figures 4.6 - 4.9 would support this claim. It can be seen in these figures that as the length of channel grows, the error probability of all three curves increases.

From figures 4.6 - 4.9, it is implied that channel estimation decreases error probability compared to the case where there is no estimation. However, it should be noted that this reduction of error probability can continue until the error probability reaches the green dotted curve.

Figure 4.10 depicts how δ can affect the performance of the system. For this figure, a channel with length of 10 ($L = 10$) is considered. Moreover, in all of these curves, ϵ is set to be equal to one.

Based on figure 4.10, as the value of δ increases, the channel varies faster and it becomes more difficult for the estimator to track the channel. The SER for the curve that represents $\delta = 0.1$ shows that our estimator does not work properly anymore. Thus, there is a limit to the performance of our estimator.

The last parameter to be discussed is ϵ. It indicates the speed of convergence of our algorithm. As the value of ϵ changes from zero to one, the estimation error converges faster to its final value. Estimation error is the energy difference between the estimated channel

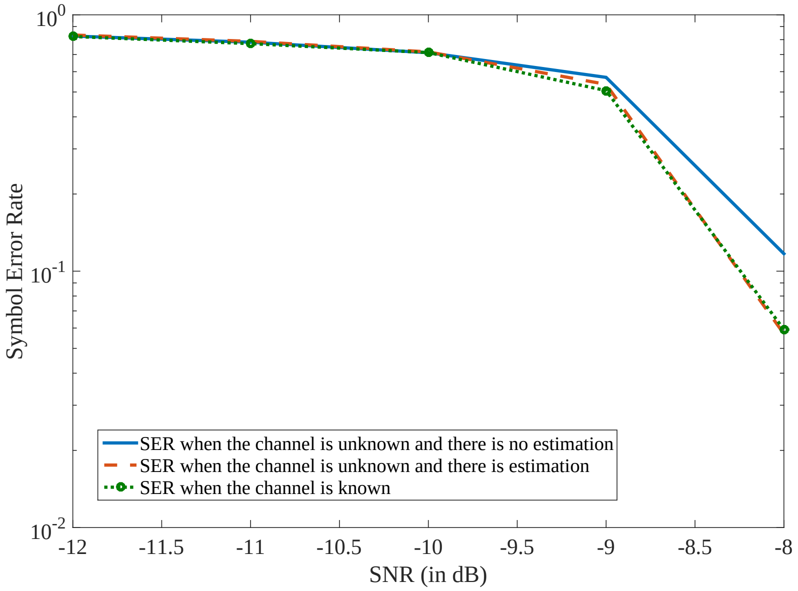

Figure 4.9: Effect of the channel length when L=640

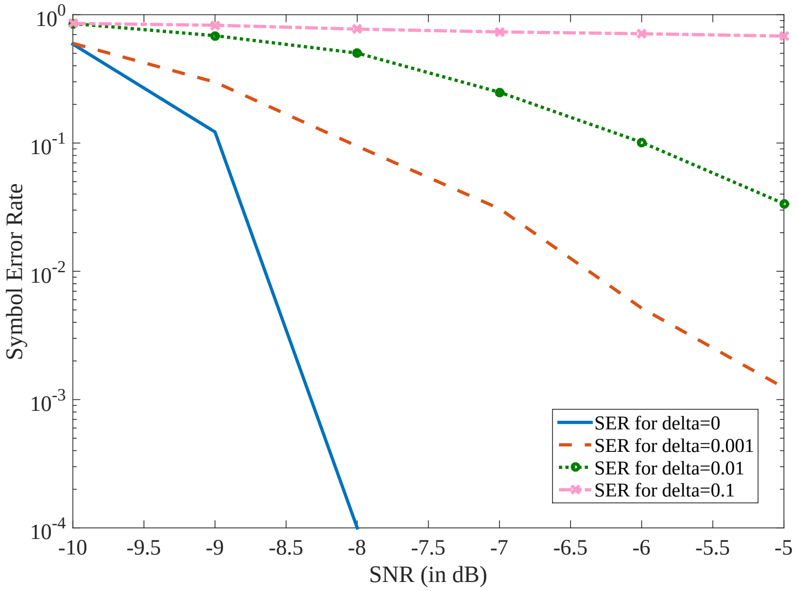

Figure 4.10: Effect of δ

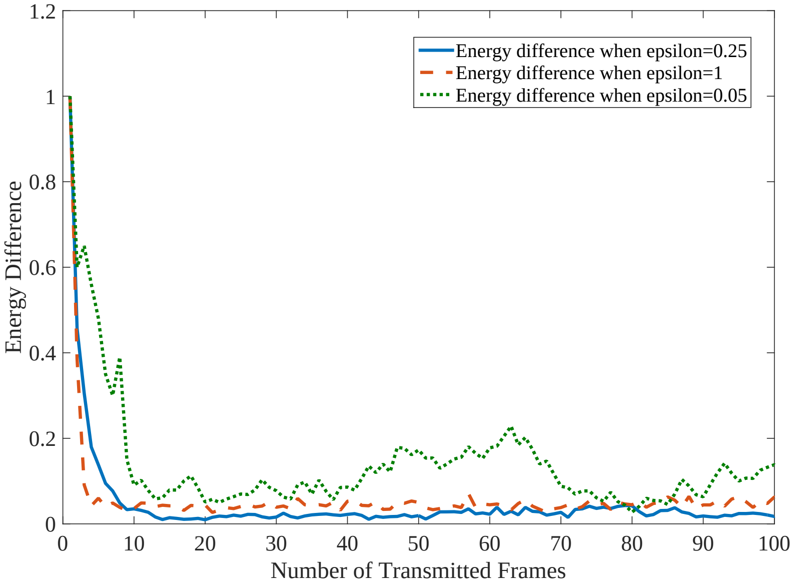

Figure 4.11: Effect of ϵ

and the real value of the channel

$$E_e = \sum_{k=1}^{64} \left| A_k - \hat{A}_k \right|^2 \tag{4.7}$$

where A_k, $k \in \{1, 2, ..., 64\}$ are 64 coefficients obtained by taking 64-point FFT of channel impulse response, and $\hat{A}_k$, $k \in \{1, 2, ..., 64\}$ are 64 FFT coefficients of the estimated channel. Figure 4.11 depicts the result of convergence when the length of channel is $L = \mathit{10}$, and $\delta = 0.001$.

As it is evident from figure 4.11, channel converges faster when ϵ has a higher value. However, if the value of ϵ is very small, then the estimator is too slow and cannot track channel variations.

Chapter 5

Conclusion

In this thesis, we proposed a blind channel estimation method for M-FSK modulation systems that employs product code. This algorithm has low transmission rate, and thus perform in low SINR. Performing in low SINR would decrease the effect of ISI on the performance of our system.

The proposed product code, which is used in our system, has a simple coding procedure, while the decoder uses a soft decoding scheme called BCJR method that reduces the probability of error. Moreover, utilizing product code would also improve the robustness of the system against burst noise because of its inherent interleaving feature.

Our proposed method uses an iterative algorithm in the receiver to decode and demodulate received frames of data. In the first iteration of this method, an output frame is generated using the initial estimation of channel. A new estimation of the channel is then computed based on this output frame. Finally, this estimated channel is used to find reshaped basis signals. Using these reshaped signals, a new output frame is generated. The goal of this algorithm is to improve error probability and reduce the difference between the transmitted frame and the estimated one in each iteration. This procedure can be done iteratively n times.

In order to simulate our estimation method, two parameters were introduced. A parameter to show the gradual change of channel in each frame time (δ) and another parameter to show the speed of convergence of the algorithm (ϵ). Using these two parameters along with channel length (L), different scenarios were considered.

At first, we investigate the effect of channel length L. As L increases, a larger portion of the energy of each symbol leaks into adjacent symbols. Therefore, as L increases, the effect of ISI as well as error probability increases.

Next, we showed that as δ increases, the difference between two consecutive channels increases as well, which would restrict our method's ability to keep track of the channel. Therefore, the performance of the system degrades and the error probability increases.

Finally, we investigate the effect of ϵ. The parameter ϵ shows how much each output frame can affect the channel estimation and represents convergence speed of the algorithm. If the value of ϵ is low, the decoder slowly tracks the channel. However, if the value of the ϵ is near one, then the decoder tracks channel variations faster but it is more probable to diverge.

References

[1] K. Abed-Meraim, E. Moulines, and P. Loubaton, “Prediction error method for second-order blind identification,” *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 694–705, Mar 1997.

[2] K. Abed-Meraim, W. Qiu, and Y. Hua, “Blind system identification,” *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1310–1322, Aug 1997.

[3] O. Al-Askary, “Iterative decoding of product codes,” 2003.

[4] A. Benveniste, M. Goursat, and G. Ruget, “Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communications,” *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 385–399, Jun 1980.

[5] J. Choi and C. C. Lim, “A cholesky factorization based approach for blind fir channel identification,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1730–1735, April 2008.

[6] Z. Ding, R. A. Kennedy, B. D. O. Anderson, and C. R. Johnson, “Ill-convergence of godard blind equalizers in data communication systems,” *IEEE Transactions on Communications*, vol. 39, no. 9, pp. 1313–1327, Sep 1991.

[7] Z. Ding, “Characteristics of band-limited channels unidentifiable from second-order cyclostationary statistics,” *IEEE Signal Processing Letters*, vol. 3, no. 5, pp. 150–152, May 1996.

[8] X. G. Doukopoulos and G. V. Moustakides, “Blind adaptive channel estimation in ofdm systems,” *IEEE Transactions on Wireless Communications*, vol. 5, no. 7, pp. 1716–1725, July 2006.

[9] W. Gardner, W. Brown, and C.-K. Chen, “Spectral correlation of modulated signals: Part ii - digital modulation,” *IEEE Transactions on Communications*, vol. 35, no. 6, pp. 595–601, June 1987.

[10] G. B. Giannakis, “Filterbanks for blind channel identification and equalization,” *IEEE Signal Processing Letters*, vol. 4, no. 6, pp. 184–187, June 1997.

[11] R. W. Heath and G. B. Giannakis, ““exploiting input cyclostationarity for blind channel identification in ofdm systems”,” *IEEE Transactions on Signal Processing*, vol. 47, no. 3, pp. 848–856, Mar 1999.

[12] J. Larsson and F. Wesslén, “On the m*-bcjr and liss algorithms for soft-output decoding of convolutional codes,” 2006, student Paper.

[13] C. Li and S. Roy, “Subspace-based blind channel estimation for ofdm by exploiting virtual carriers,” *IEEE Transactions on Wireless Communications*, vol. 2, no. 1, pp. 141–150, Jan 2003.

[14] E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrargue, “Subspace methods for the blind identification of multichannel fir filters,” *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 516–525, Feb 1995.

[15] Y. C. Pan and S. M. Phoong, “An improved subspace-based algorithm for blind channel identification using few received blocks,” *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3710–3720, September 2013.

[16] X.-W. W. R. Pellikaan and S. Bulygin, “Error-correcting codes and cryptology,” .http://www.win.tue.nl/~ruudp/courses2WC11/2WC11-book.pdf, 2012, website accessed 2013-02-28.

[17] D. H. Pham and J. H. Manton, “A subspace algorithm for guard interval based channel identification and source recovery requiring just two received blocks,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03). 2003 IEEE International Conference on*, vol. 4, April 2003, pp. IV–317–20 vol.4.

[18] R. M. Pyndiah, “Near-optimum decoding of product codes: block turbo codes,” *IEEE Transactions on Communications*, vol. 46, no. 8, pp. 1003–1010, Aug 1998.

[19] A. Scaglione, G. B. Giannakis, and S. Barbarossa, “Redundant filterbank precoders and equalizers. ii. blind channel estimation, synchronization, and direct equalization,” *IEEE Transactions on Signal Processing*, vol. 47, no. 7, pp. 2007–2022, Jul 1999.

[20] B. Su and P. P. Vaidyanathan, “Subspace-based blind channel identification for cyclic prefix systems using few received blocks,” *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 4979–4993, Oct 2007.

[21] ——, “A generalized algorithm for blind channel identification with linear redundant precoders,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 1–13, 2006. [Online]. Available: http://dx.doi.org/10.1155/2007/25672

[22] G. Taricco, “Optimum receiver design and performance analysis of arbitrarily correlated rician fading mimo channels with imperfect channel state information,” *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1114–1134, March 2010.

[23] L. Tong, G. Xu, and T. Kailath, “Blind identification and equalization based on second-order statistics: a time domain approach,” *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 340–349, Mar 1994.

[24] M. K. Tsatsanis and G. B. Giannakis, “Cyclostationarity in partial response signaling: a novel framework for blind equalization,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 5, Apr 1997, pp. 3597–3600 vol.5.

[25] J. Tugnait, “Identification of linear stochastic systems via second- and fourth-order cumulant matching,” *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 393–407, May 1987.

[26] W. J. Zeng, X. Li, and E. Cheng, “Blind and semiblind channel estimation for single-carrier block transmission systems using few received blocks,” in *2011 IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–5.