12-12-2018

Automatic Trellis Generation for Demodulation of Faster Than Nyquist Signals

Deepan Govindaraj
dxg9266@rit.edu

RIT

Automatic Trellis Generation For Demodulation of Faster Than Nyquist Signals

by

Deepan Govindaraj

Advisor: Prof. Miguel Bazdresch

Thesis Committee

Prof. William P Johnson, Professor, Graduate Program Director

Prof. Mark Indelicato, Professor

This Thesis is submitted for the partial fulfillment of the requirements for the Degree of Master in Telecommunications Engineering Technology

Electrical, Computer and Telecommunications Engineering Technology

College of Applied Science and Technology

Rochester Institute of Technology

Rochester, NY

December 12, 2018

Abstract

Mobile communication has become one of the most important and fast developing technology in the past couple of decades. Future of telecommunication raises a high demand for higher data rate and system capacity. There are plenty of researches taking place across the world to provide a better service. One such research is Faster than Nyquist signaling and it has grabbed the attention of many researchers in the recent past. In digital communication implemented using Nyquist pulses, the pulse rate is upper-bounded by twice the channel bandwidth. Signaling above this rate results in the loss of pulse orthogonality and introduces ISI. However, under certain conditions, it is possible to lose orthogonality and still maintain the same error probability, as Nyquist signaling. This allows time-compression of the transmitted symbols, resulting in a larger data rate than predicted by classic information theory results. The ISI caused by FTN signaling has a trellis structure and the transmitted symbols can be decoded using the Viterbi or BCJR algorithms.

In this thesis, we introduce an algorithm that can automatically generate the trellis for any pulse shape, constellation and time-compression factor. we have simulated the FTN system, processed and decoded by the Viterbi decoder using the trellis generated by the proposed algorithm for BPSK and PAM 4 constellations with raised cosine pulses. The simulation results are promising and encourage more research in this direction. We have discussed possible directions this research can be pursued in future work. Overall, the results would indicate that the FTN technology has a significant potential for the next generation wireless communication.

Acknowledgements

First and foremost, I would like to thank my advisor Prof. Miguel Bazdersch, for his excellent supervision and guidance throughout the course of my Masters degree and my thesis. This thesis signifies innovative research conducted for over two years at Department of Telecommunication Engineering, RIT. The department has always been supportive with my courses, thesis and my research assistance-ship. I still remembered the day when I approached him for research assistant-ship and he passionately introduced me to the concept of FTN signaling although the terminology was unaware to me at that time. His belief on me has been a great motivation to get to this point in my career. He has always been there for me, academically and emotionally and helped me get through some tough times.

I would like to thank all the faculty members of my department for supporting me during my time at RIT. I am grateful to the university for providing me with necessary resources for my work. My special thanks to my thesis defense committee Prof.William Johnson and Prof. Mark Indelicato, who contributed their valuable time to read, understand and provide helpful suggestions to my work. I great-fully thank my friend Mageswaren Ravichandran who helped me in many occasions in solving some complex logic.

Last but not least, I thank my family and friend for their unconditional love and strength, especially my grand father who has been an inspiration all my life and my cousin who encouraged and helped me in many occasions. My entire family has supported me in all the decisions I made down the road in my life. I humbly dedicate this thesis to them.

Contents

List of Figures

List of Acronyms used

AWGN	Additive White Gaussian Noise
BCJR	Bahl-Cocke-Jelinek-Raviv
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
FDMA	Frequency Division Multiple Access
FPGA	Field Programmable Gate Array
FTN	Faster Than Nyquist
i.i.d	independent and identically distributed
ISI	inter symbol interference
MATLAB	Matrix laboratory
MIMO	Multiple-input-multiple-output
MF	Matched Filter
OFDM	Orthogonal frequency-division multiplexing
PAM	Pulse amplitude modulation
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RC	Raised Cosine
RRC	Root Raised Cosine
SNR	Signal to Noise Ratio

List of Nomenclature used

E_b	bit energy
B	Bandwidth
a_n	Information Sequence
a_k	input sequence estimate
b_k	Output sequence estimate
N_o	noise power density
$n(t)$	Noise
β	Roll-off Factor
τ	Time Acceleration Factor
E_s	Symbol energy
f_s	Sampling frequency
R_p	Pulse Rate
T_p	Pulse Duration
$s(t)$	Transmitted signal

Chapter 1

Introduction

An enormous amount of progress has been made in the communication technologies in the past couple of decades. On the other hand, computer networking has laid a foundation for the most significant global medium for information exchange called the internet. The improvement in the existing technology is a never ending urge for humanity, researchers have continuously exploited the available resources to enhance the technologies, adapting to the demand of the consumers.

This thesis work is based on research that has attracted many researchers in recent years called FTN signaling. Harry Nyquist along with Shannon has made some important contribution to the communication theory. While Nyquist was working at Bell Labs, he published a paper on transmission theory in which he discusses the Nyquist criterion. The Nyquist criterion has served as the base of communication systems design for a long time[1]. In this thesis, I will explore a contradicting idea to that of the Nyquist criterion in order to improve the data rate of the communication system.

1.1 Background

In digital communication, the source information is converted into digital bits, then transform them into a sequence of real information carrying symbols $\{a_k\}$, $k \in \mathbb{Z}$, using any modulation techniques. These symbols are further converted into continuous signals just before transmitting through a noisy wireless channel. These random symbols are assumed to belong to a finite set (or *constellation*) $\mathcal{A}$, and to be uncorrelated and uniformly distributed.

Most of the wireless communication technologies use raised cosine pulses for transmission as they are practical to implement compared to the sinc pulse or any other pulse shape for that matter, and they have a configurable excess bandwidth. It also relatively has fast decaying tails based on its roll off factor. According to Nyquist, if the pulses are transmitted at a pulse rate that is upper bound by twice the channel bandwidth, they will be orthogonal to each other[1], which means the pulses will not interfere with the adjacent pulses. This pulse rate was defined as Nyquist rate, and the pulses can be recovered using a matched filter. Signaling above this rate results in the loss of pulse orthogonality and introduces Inter Symbol Interference (ISI). Most of the communication systems today are based on the Nyquist criterion and trying to approach Shannon's capacity.

The idea of increasing the spectral efficiency by ignoring the Nyquist criterion and transmitting pulses faster than the Nyquist rate is called Faster than Nyquist signaling. The history of Faster than Nyquist signaling started way back in 1975 when James Mazo [2] published his findings in the possibility of transmitting sinc pulses Faster than Nyquist and still able to achieve

the Euclidean distance between symbol sequence as same as in Nyquist signaling, where Euclidean distance in the distance between the pulses which determines the probability of symbol error.

The set of modulation techniques that compress symbols in time and frequency or both are known as faster than Nyquist signaling. Mazo did not continue the research in FTN in 1975 as there were significant challenges in implementing FTN systems like increased complexity in the receiver to handle the ISI, the scarcity of resources back then and the demand for higher data rate was less. As the demand for higher data rate is too high nowadays, the researchers have started to show a keen interest in designing FTN system for the next generation mobile communication.

1.2 Focus of the thesis

This master's thesis mainly focuses in exploiting the possibilities of unfolding the FTN systems and to recover the transmitted FTN symbols that are influenced by ISI with better SER. The main goals of the thesis are as follows:

- Derive at a discrete time FTN model.
- Devise an algorithm to generate a trellis for demodulation of FTN signals automatically.
- Study the results of symbol error rate for various modulation techniques.

1.3 Contribution

The pulses that are influenced by ISI has a trellis structure, and the transmitted symbols can be decoded using techniques such as Viterbi or BCJR algorithms. In this thesis, we present an algorithm, that can automatically generate the trellis of an FTN signaling scheme given the pulse shape, the signaling rate and the constellation. The calculated trellis is stored in a matrix with four columns: the current state, the input, the next state, and the branch metric. This matrix along with the sampled symbols received from the AWGN channel can serve as input to any trellis decoder. The main idea behind the algorithm is to calculate all possible ISI values by considering every amplitude of the interfering pulses. More background on the Viterbi decoder and the algorithm is provided in chapter 5 and 6.

Chapter 2

Literature Review

In this chapter, we will be discussing the survey on the literature of FTN signaling, starting from the early developments to the current state-of-the-art in chronological order. The Faster than Nyquist signaling makes its first appearance in the literature in the mid 1960s. Even though it is more than 50 years in the past, there seems to be only a handful of papers regarding FTN signaling up until very recently. The literature review is classified into three chronological periods. Each section will focus on the highlights in the research in the period and the problems faced by the researchers.

2.1 Early Days of FTN (the 1960s - 1985)

The early 1960s was the period when telecommunication started to emerge, and the availability of resources was very minimal. It was Bell Labs and its researchers who contributed to the advancement of communication for the most part. The possibility of transmitting faster than Nyquist rate on a

band limited communication channel, for better spectral efficiency, grabbed the thoughts of those researchers. Several researchers published their findings, but none of them had promising results until Mazo's work in 1975. In those days it was considered that usage of FTN signaling posed more complexity and sufficient technology was not available to handle the Inter-symbol Interference. It was rendered not suitable for practical use, and the research was suppressed within Bell labs.

To the best of our Knowledge, the first published reference on FTN dates back to 1965, when Tufts, of Harvard University, showed that "it is possible to transmit a finite sequence of real numbers at an arbitrarily high rate through any linear, time invariant, noiseless transmission medium" [3]. Tufts also derived an FTN scheme with an analytical framework for designing minimum mean square equalization that was limited to a short burst of pulses. In 1967, Landau [4] from Bell Labs undermined Tufts work by saying "data cannot be transmitted at a rate higher than Nyquist's Sampling rate" by defining a concept called stable sampling. Soon after that Tufts challenged Landau's claim in 1968 by arguing that "it is possible to transmit any finite number of data elements at rates Faster than the Nyquist rate" [5].

Also in 1967, Andrew.J.Viterbi [6] published an article in which he proposed an algorithm for decoding convolution codes that was asymptotically. During the same time, Saltzberg attempted to reduce the channel bandwidth below the Nyquist bandwidth to simulate effective FTN transmission and observed that "the system bandwidth can be reduced slightly below the Nyquist band without catastrophic results" [7]. In 1970, Lucky from Bell labs argued in his paper [8] that FTN transmission causes ISI that cannot be removed

by a decision feedback equalization, despite the advancement in equalization which renders FTN unsuited for practical use.

Despite all these misconceptions, Mazo from Bell Labs published his paper in 1975 that showed promising results that changed the viewpoint on FTN transmission for all the researchers. He proposed that the pulses can be transmitted Faster than Nyquist by 25% and still preserve the minimum Euclidean distance which is considered to be one of the most important metric to measure the performance of the communication system. He used sinc modulated pulses in most of his work and proposed a transceiver architecture based on it. He also focused on channel capacity and error control coding techniques in his paper. Nevertheless, Mazo led a path for the researchers to build on for the future. Around this time, Forney [9],[10] was pursuing Viterbi's work on the optimum viterbi decoder. He planted the idea of using the Viterbi algorithm for maximum likelihood ISI receivers.

Around 10 years later, Foschini in 1984 analyzed the feasibility of FTN Signaling with Quadrature Amplitude Modulation (QAM) [11]. He compared the results of FTN transmission between the binary symbols and QAM signals and concluded that the binary symbols offer only a minor gain over QAM due to high spectral side-lobes and implementation complexity of the ISI. Foschini said in his paper that "one cannot dismiss (FTN signaling using multi-level symbols)" and "(such) systems may have some value (over QAM)." Unfortunately, Foschini's paper in 1984 was the last one on FTN signaling from Bell labs. On the other hand, Mazo's work was pursued by other researchers like Hajela and Mazo's himself published a joint paper with Landau in 1988 [12], which was his final work on FTN Signaling.

2.2 Mid Days of FTN (mid 1980s - 2000s

After the mid-1980's, the Bell labs researchers seemed to have lost interest in the FTN signaling, while other researchers picked up where they left and continued on in various paths. Most of the work during this period was an extension to Mazo's work on calculating the minimum Euclidean distance. There were very few papers on FTN published during this period involving the concepts of using different modulated pulses, channel coding techniques and ways to suppress the ISI caused by the FTN pulses. Most of the publishers did not pursue their course of research in FTN signaling after their initial publications.

During the years 1987 to 1992, Hajela [13],[14] published a series of papers, mostly extending the work of Mazo, out of which his most important contribution was to mathematically formulate the problem of finding minimum Euclidean distance and prove that 25% increase in data rate was the best possible result for an FTN system. Most of his work also employed sinc pulse for simulations. It was in the year 2003, Liveris and Georghiades [15] moved from sinc pulses to raised cosine pulses which were more practical to implement. They mathematically showed that the 25% increase was possible with rRc pulses along with the minimum distance calculations. They also designed constrained coding to keep the minimum Euclidean distance constant for higher signaling rates. They claim that the ISI caused by the FTN system can be removed practically using advanced coding techniques like iterative Turbo equalization.

Since the approach towards raised cosine pulses was introduced, there were many publications on different practical pulses and coding techniques.

After Foschini's work with QAM signals in 1984, there were no attempts of multi-level FTN systems until Wang and Lee's [16] publication in 1995. Their work was crucial with some fascinating ideas multi-level FTN, modification of FTN transmit filter response and using a whitened matched filter at the receiver. They also proposed the idea of using an iterative decoder to handle ISI. Meanwhile, in the 1990s, few independent developments were made in the field of equalization of coded and uncoded systems[17], and Berrou discovered turbo codes and turbo equalization [18],[19].

2.3 Recent days of FTN (mid 2000s - present)

Most of the early work in FTN was hindered due to the unavailability of processing resources and no need for higher data rates. However, due to the recent escalation in the cost of channel bandwidth and reduced cost of memory and processing resources, FTN has grasped the attention of many researchers. FTN is considered as a method of trading processing complexity for improved spectral efficiency. With that being said and all the other advancements in the communication technologies, FTN has currently become the most important ongoing research and a potential contender in the 5th generating wireless communication.

Rusek and Anderson from Lund University in Sweden, have become the most important contributors for the research in FTN signaling. It was initially started as Rusek's Ph.D. thesis work. They continued to pursue this course of research made various approaches to study FTN signaling in great detail. They were the first to explore the possibility of combined time and

frequency FTN and expanded their research in channel capacity and finally implementation of FPGA hardware. They also analyzed the capacity of FTN signals when the modulation symbols are constrained to be independent and identically distributed is higher than the traditional orthogonal signaling. They also extended Mazo's work on the in-variance of minimum Euclidean distance for FTN signaling to Frequency-Time FTN using root-raised cosine pulses. Their simulation results were better than the single dimensional FTN system and also achieved same error performance as the conventional OFDM system.

Over the years, Rusek and Anderson conducted various research on FTN systems like the design of practical coding systems, equalization techniques like turbo equalization and decoders such as BCJR decoders and Viterbi decoders[20],[21]. In most of their simulations, they used an additive white Gaussian noise (AWGN) channel. Moreover, they were the first to bring FTN into practice by implementing their FTN transceiver in a complementary metal oxide semiconductor(CMOS) in a 65nm architecture and field-programmable gate array(FPGA)[22],[23]. FTN was not only considered for wireless communication, but also fiber optic communication. Many kinds of research were performed on considering FTN in long-haul fiber optic communication links to increase spectral efficiency.They also attempted to review the possibilities of applying FTN signalling to higher level modulation techniques [24],[25].

In 2010, Yoo and Cho [26] proved that the FTN signaling using binary modulation symbols could achieve the capacity od i.i.d Gaussian FTN signaling as signaling tends to infinity. During the same year, Mcguire and

Sima reformulated the FTN system to propose a design for a low complexity FTN receiver. They simulated an FTN receiver that can achieve ISI free performance at high SNR. In 2017, Ji Zhou along with hi colleagues proposed a capacity limit for the faster than Nyquist in which the performed the simulation on a frequency division multiplex signaling[27].

Lately, FTN research has taken various directions like hardware implementation of frequency-Time FTN multi carrier system and study of FTN in multiple input multiple output(MIMO) channels, multiple access, and broadcast systems[28]. In case of MIMO, not only frequency and time diversity is considered, even spatial diversity is currently under investigation for massive MIMO FTN systems. All these recent vibrant research developments indicate that there is now a growing interest in the topic of FTN signaling and its potentials are beginning to be recognized in the research community.

Chapter 3

FTN Theory

Even though radio communication emerged way back in the 18th century, the technologies were commercialized only in the early 19th century. The contribution of renowned scientists like Marconi, Bose, Tesla has paved the way for the future in wireless communication. It was in the late 1980s, when the first generation of the mobile telephone was introduced. These used analog frequency modulation and they were meant only for voice communication. Later generation after generation the researchers continued to make it better and faster. One of the ground-breaking discovery in digital communication was Shannon's Information theory based on the work of Harry Nyquist[1]. In this chapter, we will be discussing the traditional Nyquist signaling and idea of faster than Nyquist signaling along with its advantages and shortcomings.

3.1 Nyquist Signaling

In digital communication, every information undergoes a series of stages before transmitted like encoder which converts the information into bits and then these bits are modulated into a sequence of real symbols $\{a_k\}$, $k \in \mathbb{Z}$. They are administered to a pulse shaping filter where it gets shaped into a pulse $p(t)$ suitable to be transmitted on a communication channel. The pulse shapes considered in this thesis are sinc and raised cosine pulse. A set of random symbols are generated that are assumed to belong to a finite set (or *constellation*) $\mathcal{A}$. These symbols are also uncorrelated and uniformly distributed. The sequence of pulses are transmitted at the rate R_p, and they are separated by a pulse duration T_p such that they don't interfere with each other. The transmitter generates the signal .

$$s(t) = \sum_k a_k p(t - kT_p), \tag{3.1}$$

where $T_p > 0$. The pulses $p(t - kT_p)$ form an orthonormal set; that is,

$$\int_{-\infty}^{\infty} p(t - mT_p)p(t - nT_p)\ \mathrm{d}t = \begin{cases} 1, \text{ if } m = n, \\ 0, \text{ otherwise.} \end{cases} \tag{3.2}$$

with this scheme, symbols are transmitted at a rate $R_p = 1/T_p$. The received signal is represented as

$$r(t) = s(t) + n(t) \tag{3.3}$$

where $n(t)$ is white Gaussian noise with power spectral density $N_0/2$. The

symbols are estimated by calculating the sufficient statistics

$$r_k = \int_{-\infty}^{\infty} r(t)p(t - kT_p) \tag{3.4}$$

$$= a_k + n_k, \tag{3.5}$$

where the noise samples n_k are uncorrelated Gaussian random variables with zero mean and variance $\sigma_n^2 = N_0/2$.

If $p(t - kT_p)$ is an orthonormal set of pulses, then an integral of the product of two different pulses is zero, and an integral of the product of the same pulse is one. Which means at the sampling duration of a particular pulse, all the other pulses interfere at zero, implying there is no intersymbol interference.

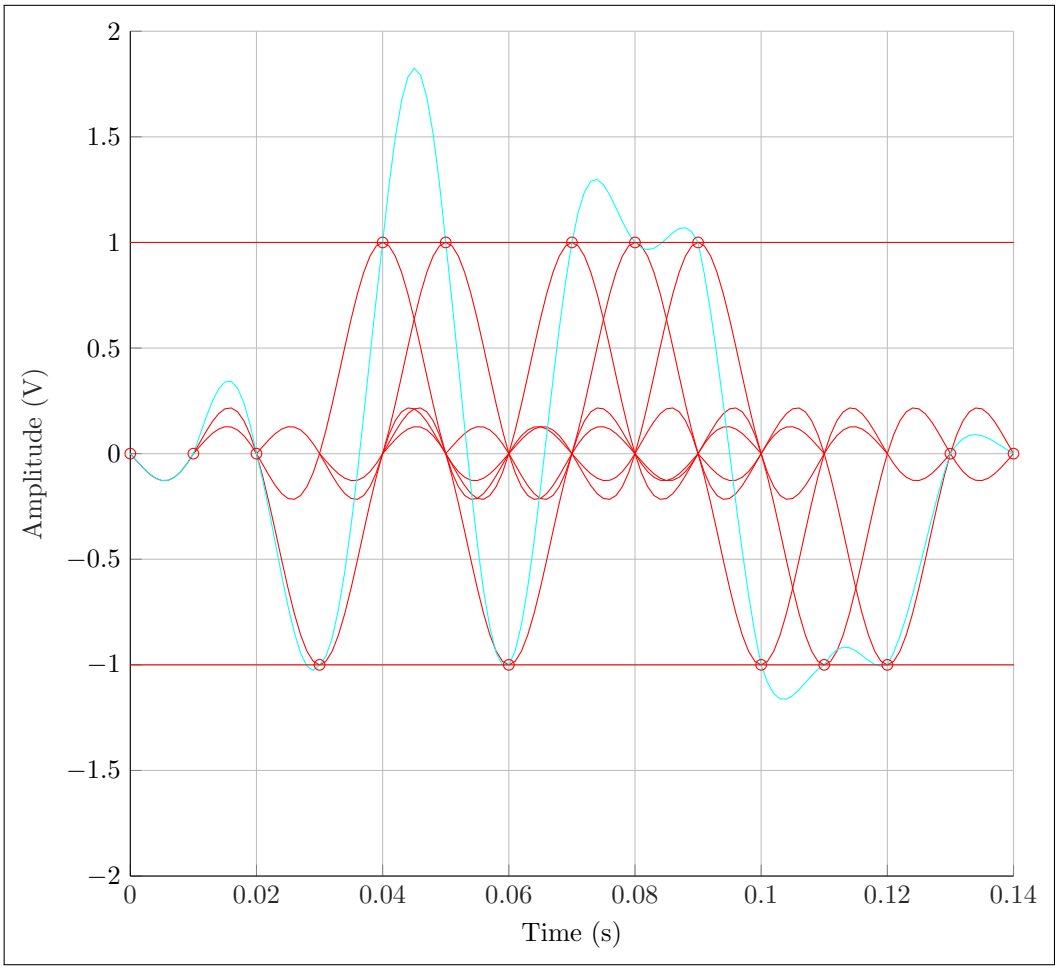


Figure 3.1: A train of 10 "sinc" pulses with $a_k = + - 1$, $R_b = 100B/s$, and pulses of duration 0.01 s. The red curves are individual pulses, while the cyan curve is $s(t)$, the sum of all pulses. The circles indicate the sampling points, where $s(kTp) = a_k$.

The Fig:3.1 represents a train of 10 *sinc* pulses with $a_k = [-111 - 1111 - 1 - 1 - 1]$. These pulses are transmitted at a bit rate of $100B/s$ with a pulse duration of $0.01s$. The red curves represent the individual sinc pulses and the cyan curve is the $s(t)$ which is the sum of all pulses. These pulses are orthogonal to each other, and they are known as Nyquist pulses. The

red circles indicate the sampling points which shows there is no ISI. The communication technique described above is known as Nyquist signaling, and it is optimal, in the sense that the probability of error is minimized. On the other hand, the receiver can be implemented with much less complexity using a filter matched to p(t) and sampled at kT_p Effectively the receiver can focus on one pulse and a_k symbol at once without interfering from others. Note that the interval of T_p is crucial here, and if the pulses are not separated by precisely nT, they will not have the zero-ISI property.

According to Nyquist criterion, it is also necessary that the pulse rate should be less than twice the channel bandwidth $R_p \leq 2B$, where B is the bandwidth of $s(t)$. Equality is achieved when $p(t)$ is a sinc function. The signal $s(t)$ needs to be carrier modulated to a suitable frequency to be transmitted within the channel bandwidth. The carriers perform this modulation since they have been allocated to a particular frequency. Since we do not have that restriction, we work with directly with the baseband signal for simulations.

3.2 Faster than Nyquist Signaling

When the pulses are transmitted at an interval T_p where $R_p = 1/T_p$, and if they are not orthogonal to each other, and they tend to have ISI, where the neighboring pulses are not invisible to each other and they overlap at one's sampling duration[29]. In that case, the received symbols become

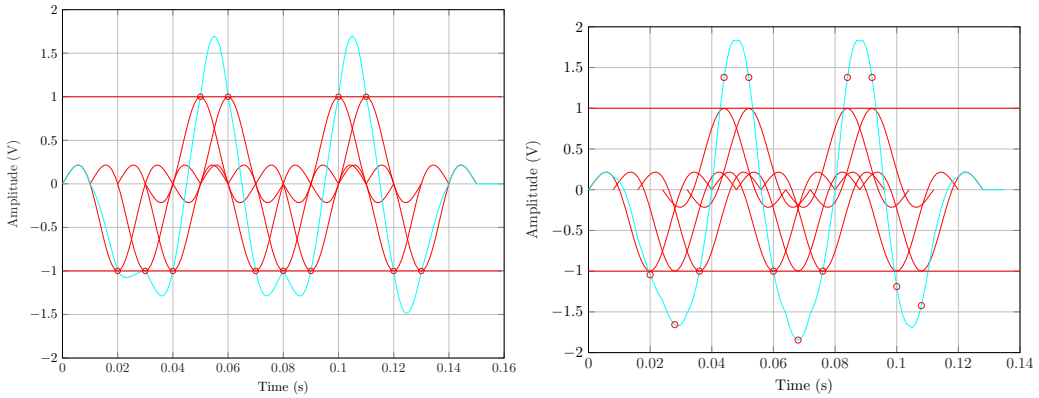
$$r_k = a_k + n_k + \text{ISI}, \tag{3.6}$$

The effect of ISI is usually as increases in the probability of symbol error. However, the interference does not always degrade the system performance. For example, partial response signaling systems introduce ISI in a controlled manner, to shape the spectral density of $s(t)$ [30]. The ISI can either be a constructive interference or destructive interference depending on the polarity of the pulses interfering.

In 1975, Mazo[2] discovered that in some instances the orthogonality could be lost without increasing the probability of error. He proposed that if the pulses transmitted at the rate faster than Nyquist and squeezing them in time by introducing a multiplier, ultimately a time compression factor. In this case, the transmitted signal will be represented as

$$s_{FTN}(t) = \sum_{k} a_k p(t - \tau k T_p), \tag{3.7}$$

where $\mu \leq \tau \leq 1$ is the *time-compression factor*, and μ is the Mazo limit, which depends on $p(t)$. Note that we can use this signal to transmit at a rate R_p/τ which is potentially larger than $2B$. In Fig. (3.2), a same set of sinc pulses representing the following sequence $[-1, -1, 1, 1, -1, -1, -1, 1, 1, -1]$ is transmitted with $\tau = 1$ which has no ISI and with $\tau = 0.8$ that has ISI.The blue line represents the actual transmitted signal $s(t)$. In both the cases $D = 4$ (number of lobes). The fact that we can transmit at a higher rate than Nyquist with the same probability of error is surprising.In this scenario, with $R_p = 100$ and $\tau = 0.8$, the pulses in the FTN system arrive at 20% faster than the traditional system. The set of modulation techniques that compress symbols in time, frequency or both is known as faster than Nyquist signaling [20].



(a) Nyquist pulses ($\tau = 1$) with $D = 4$. There is no ISI.

(b) non-orthogonal pulses ($\tau = 0.8$) with $D = 4$.

Figure 3.2: Same set of pulses transmitted at Nyquist rate and Faster than Nyquist

3.2.1 Error Probability

The FTN Signal $s_{FTN}(t)$ still has the shape of $P(f)$, even though $1/\tau$ more bits are carried in the same bandwidth. Euclidean distance or the minimum distance d_{min} is the parameter that determines the probability of symbol error E_s. The d_{min} and E_s plays an enormous role in understanding FTN systems. The signal $s(t)$ generated by the sequence of bits a_k. Let $s_i(t)$ and $s_j(t)$ be two signals whose symbols are same up to n_0 and different thereafter at least at the position n_{0+1}. Then d^2_{min} is the least square Euclidean distance between any such pair,

$$(1/2E_b) \int |s_i(t) - s_j(t)|^2 dt, i = j \tag{3.8}$$

The error rate for the symbols with the best detection tends to

$$Q\sqrt{d_{min}^2 E_b/N_0} \tag{3.9}$$

As SNR grows, the square minimum distance d_{min}^2 in eq. (3.9) with binary orthogonal pulses is always 2, no matter what the pulse shape. This quantity is called the matched filter bound, and it and the corresponding error rate $Q\sqrt{2E_b/N_0}$

Chapter 4

FTN Transmitter and Receiver

Even though we live in a digital era, all of the present-day communication systems use analog wave-forms to communicate digital information from one place to another. The communication systems are designed with some constraints to consider, like signal power, spectral efficiency, bandwidth, the probability of symbol and bit error and complexity to name a few. The information from the source to the transmission channel undergoes various conversions to ensure the recovery of the information on the receiver end. The Figure 4.1 shows all the blocks of a communication system both on the sender and the receiver side.

The crucial role of a transmitter in a communication system is to guide the information source all the way to transform to the format that the communication channel could carry them to the destination. Any information source can be represented as binary digits by an encoder. In this thesis, we will not be discussing about the encoder, as we will presume that the information source is already encoded in binary digits by randomly generating it

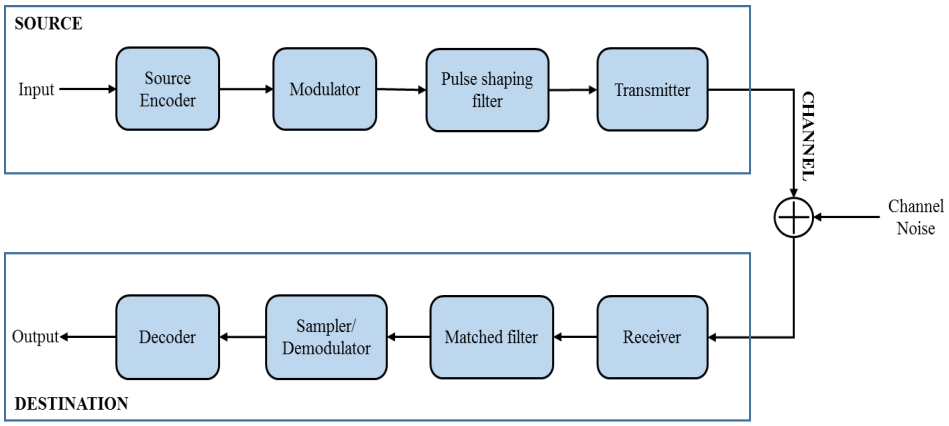


Figure 4.1: Block diagram of a Communication System

which are equally distributed and independent. The binary digits are passed into a pulse shaping filter to transform them into a waveform of pulses that is better suitable to be transmitted into the communication channel. Then the pulses are modulated to different amplitudes and phases based on the modulation scheme before transmitting the signal into the channel. In this chapter, we will discuss about the building blocks of an FTN transmitter, how the signal is transmitted in the AWGN channel and its bandwidth efficiency and channel capacity.

4.1 Pulse Shaping Filter

Pulse shaping is a process of changing the binary information into a waveform of transmitted pulses that is suitable for the communication channel

available. Considering the situation for increasing data rate for a system such as this with a fixed bandwidth without degrading the accuracy of the signal, we see the bandwidth constraint to be constant. One possible way is to modify the pulses effectively such that it requires less bandwidth than the contemporary means. The pulse shape determines the signal's spectrum. So, always deciding the pulse shaping filter for the transmitter is a trade-off between spectral efficiency and the complexity. In a system with band-limited channels during the increased modulation rate of a transmitted signal, it is highly probable to have distortion or ISI due to bandwidth limitation. Since we are going to induce ISI ourselves, we can't afford this. So it requires fixing the transmitted waveform into the shape such that the signal remains in the prescribed bandwidth. Usually, these pulse shaping filters at the transmitter are used in reference to the corresponding matched filter at the receiver end for optimal performance. There are many pulse shaping filters in practice. In this thesis, we will be using Sinc shaped filter and Raised cosine filter for very specific purposes which will be discussed further.

4.1.1 Sinc Shaped filter

The Sinc function $sinc(x)$ is one of the common pulse shapes which is also called as "Sampling function". The full name of this function is "Sine cardinal", but it is commonly referred to as "Sinc". In digital communication and information theory normalized version of this function is used where it is represented as

$$sinc(x) = sin(\pi x)/(\pi x) \tag{4.1}$$

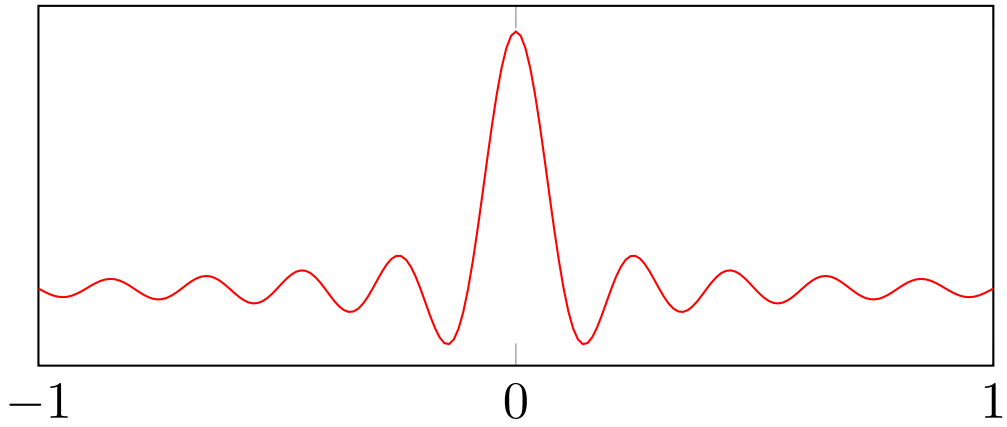


Figure 4.2: Sinc Pulse

In other words, the Sinc pulse is a sine wave that decays in amplitude as $1/x$ as shown in the figure 4.2. The Fourier transform of a Sinc pulse is equivalent to a rectangular shape in the time domain. Due to this, it is also called as boxcar filter. The phase components of this pulse are all zeros, and it is symmetrical in the time domain. Theoretically, this is the best pulse shaping filter, but it cannot be implemented precisely. Moreover, it is a non-causal filter with slow decaying tails.

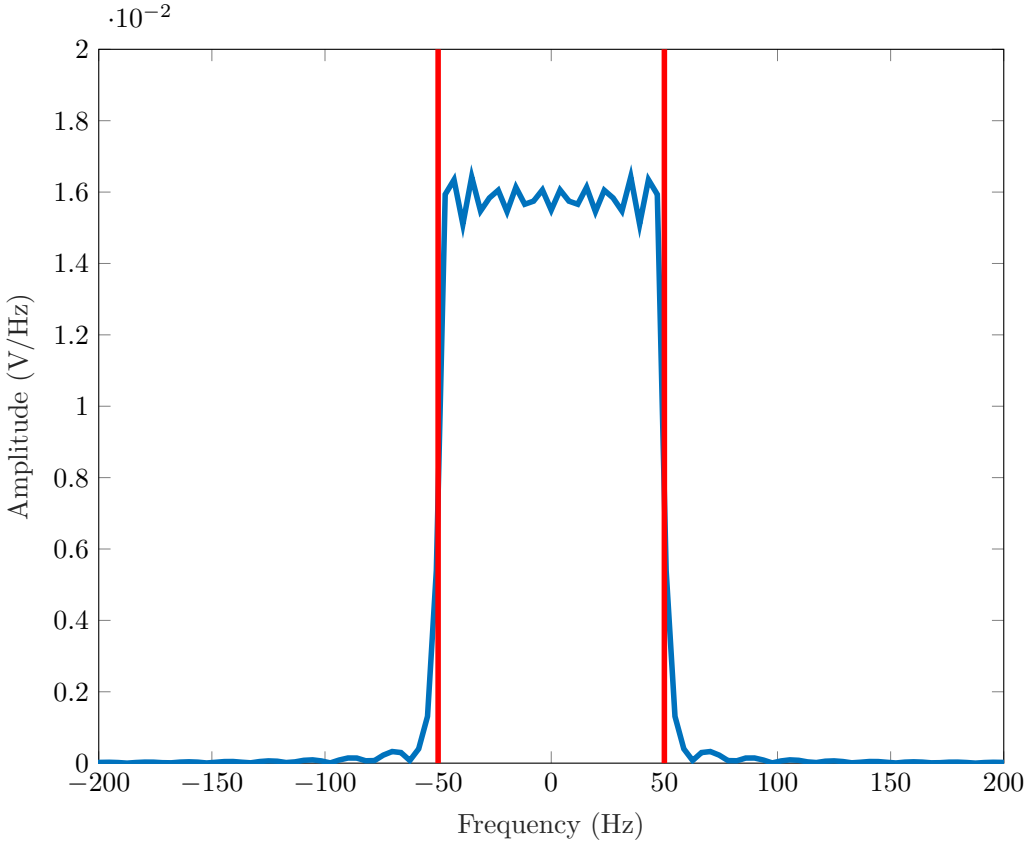


Figure 4.3: Magnitude spectrum of "Sinc" Pulse. The blue lines represent a "Sinc" pulses transmitted in time domain. The red line represents the channel bandwidth utilized by this transmission.

The Fig:4.2 represents a sinc pulse in the time domain, which has impulse response from the pulse shaping filter centered at 0 and decaying tails symmetric on both sides. Fig:4.3 represents the sinc pulses in a frequency domain with the 50Hz channel bandwidth as shown in the figure, as we know $R_p = 2B$.

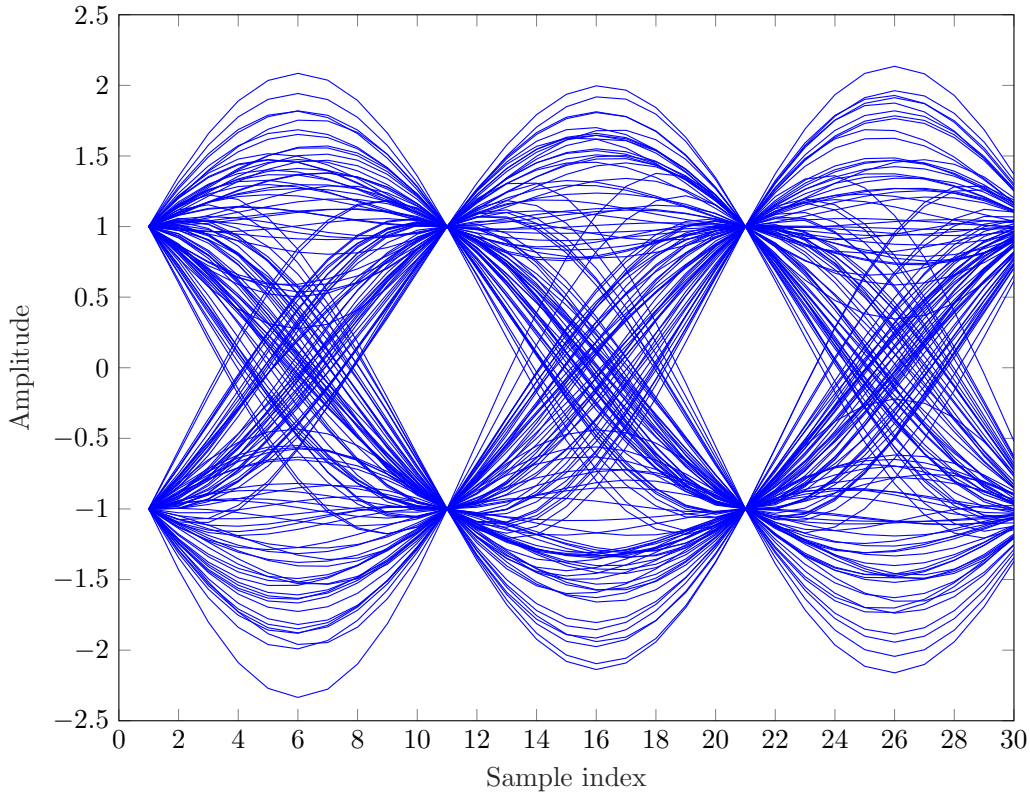


Figure 4.4: Eye Diagram of Sinc Pulse

The Fig:4.4 represents an Eye diagram of the sinc pulse. An eye diagram is a pattern of series of pulses from different time instance plotted over and over again to provide a holistic view of the signal. It is mostly used to evaluate the integrity of the signal and effects of channel noise and ISI. As shown in the figure, if the signal converges at certain points, then its assured that the signal did not experience any ISI.

The Sinc pulses have to be transmitted at a precise time interval, in order to satisfy Nyquist criterion and achieve bandwidth efficiency. If not the results will be catastrophic with ISI and problems like phase error and

synchronization. In this thesis, we use sinc pulse, and we also control the number of lobes of the sinc pulse with a variable D as we needed a controlled environment for testing the influence of ISI

4.1.2 Raised Cosine filter

The raised cosine pulse has the shape similar to that of the sinc pulse, but with a configurable, fast decaying tails. The name raised cosine refers to the shape of the pulse in the frequency domain, not in the time domain. Raised cosine pulse basically can be bandwidth-efficient with more bits/sec/Hz and power efficient. Unlike sinc pulse, Rc pulse does not take a sharp rectangular shape in the frequency domain, with an everlasting tail in the time domain, it has configurable time, and frequency domain shapes based on its roll-off factor β.

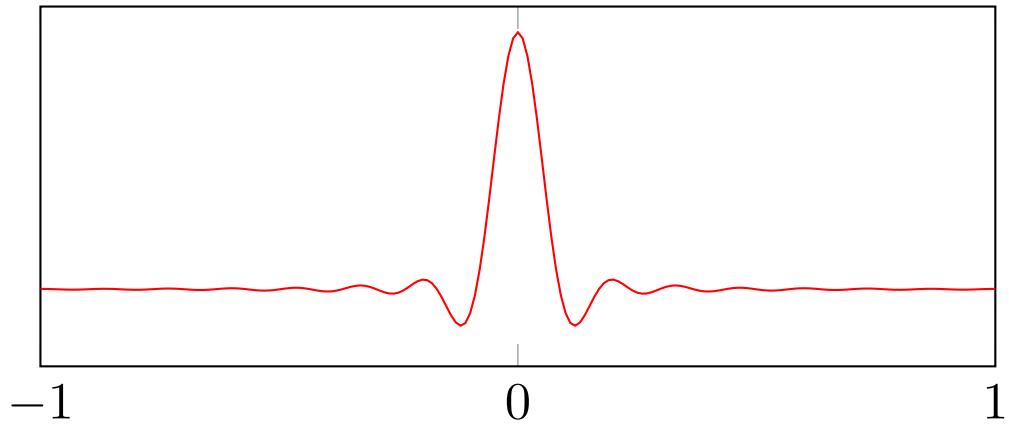


Figure 4.5: Raised cosine pulse $\beta = 0.5$

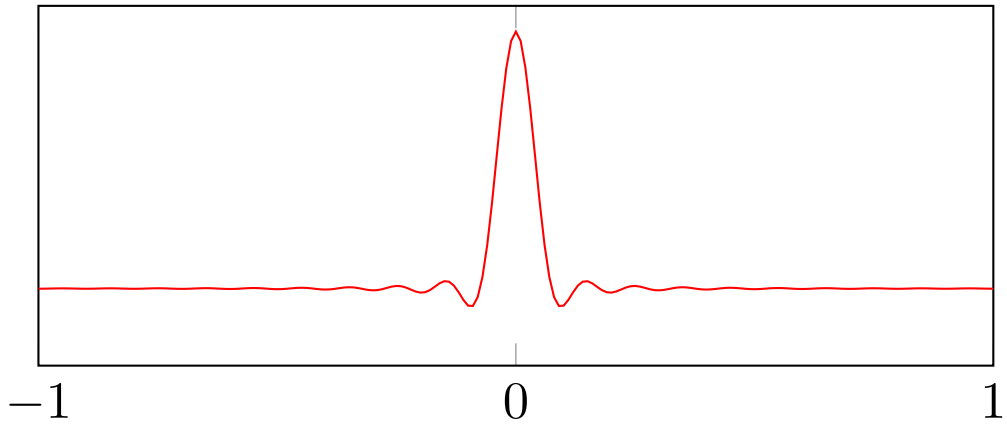


Figure 4.6: Raised cosine pulse $\beta = 1$

As depicted in the Figures 4.5 and 4.6, β governs the bandwidth occupied by the pulse and the rate at which the tails of the pulse decay. A value of $\beta = 0$ offers the narrowest bandwidth, but the slowest rate of decaying tails in the time domain, almost representing the characteristics of a sinc pulse, on the other hand when $\beta = 1$, it requires twice the bandwidth and taking the shape of a perfect cosine pulse in the frequency domain, but the tails decay rapidly in the time domain, which provides a significant advantage in case if Inter-symbol interference.

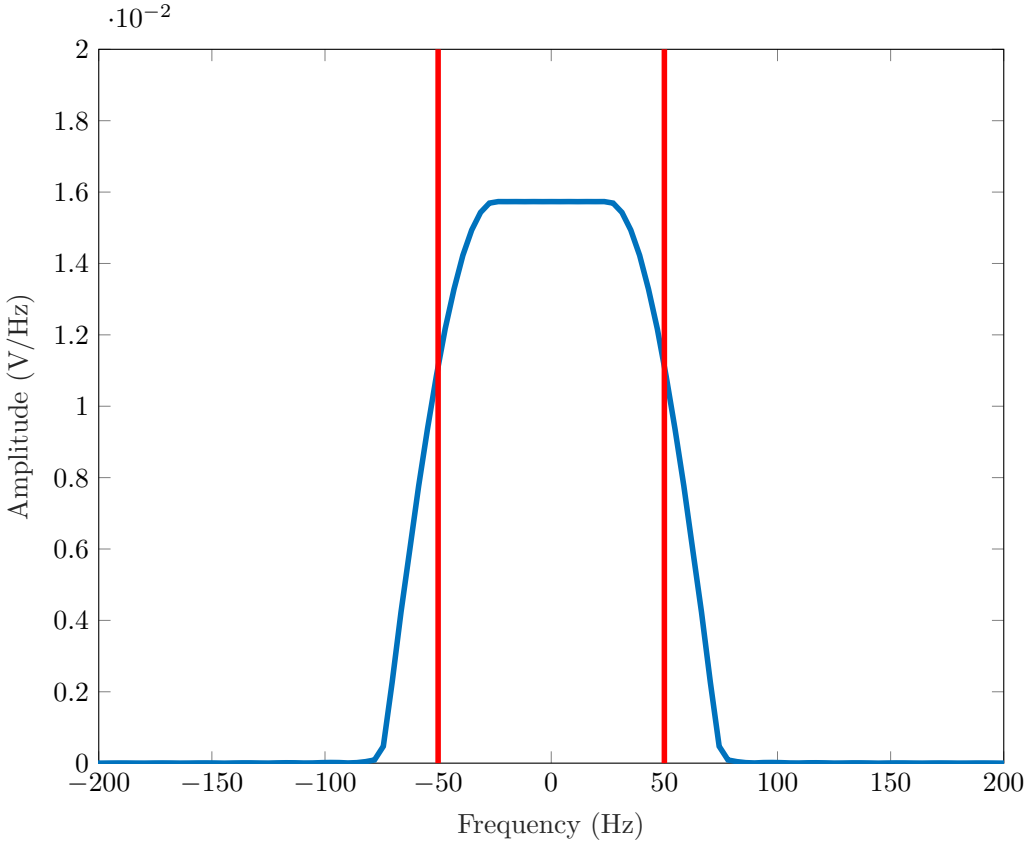


Figure 4.7: Magnitude spectrum Raised cosine pulse with roll-off factor $\beta = 0.5$. The blue lines represent a "rc" pulses transmitted in time domain with the utilization of 25% excess bandwidth and the red line corresponds to the $R_p/2$

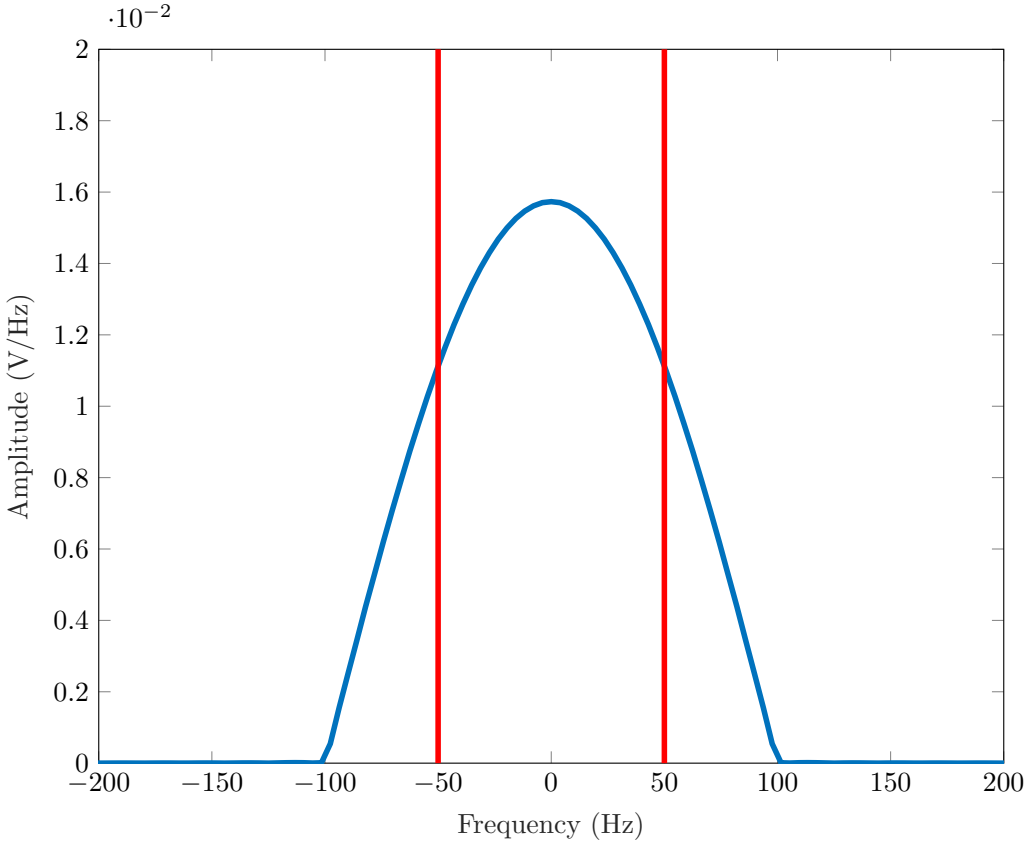


Figure 4.8: Magnitude spectrum of Raised cosine pulse with roll-off factor $\beta = 1$. The blue lines represent a "rc" pulses transmitted in time domain with the utilization of 50% excess bandwidth and the red line corresponds to the $R_p/2$

Basically, roll-off factor gives the measure of the excess bandwidth occupied by a digital filter as compared to that of the theoretical minimum Nyquist bandwidth. Bandwidth utilization is the most crucial parameter in wireless communication. In real-time communication, the signals usually tend to occupy the bandwidth more than specified by Nyquist. Officially this excess bandwidth is referred to as a roll-off factor. The roll-off factor is represented as

$$\beta = \Delta B/(R_p/2) \tag{4.2}$$

$$\Delta B = (\beta R_p)/2 \tag{4.3}$$

where ΔB is the excess bandwidth required due to the roll-off factor and R_p is the pulse rate. In this thesis we use raised cosine pulse as the pulse shaping filter with configurable beta values, depending on the bandwidth available and the system variable are adjusted accordingly to accommodate the changes and produced the train of rc pulses to be transmitted. Fig:4.5 is a raised cosine pulse in the time domain with roll-off factor $\beta = 0.5$ and it is clearly visible that the tails are decaying fast than the sinc pulse. In this case of β and symbol rate $R_s = 100$, the excess bandwidth required is

$$\Delta B = (0.5 * 100)/2 \tag{4.4}$$

$$\Delta B = 25Hz \tag{4.5}$$

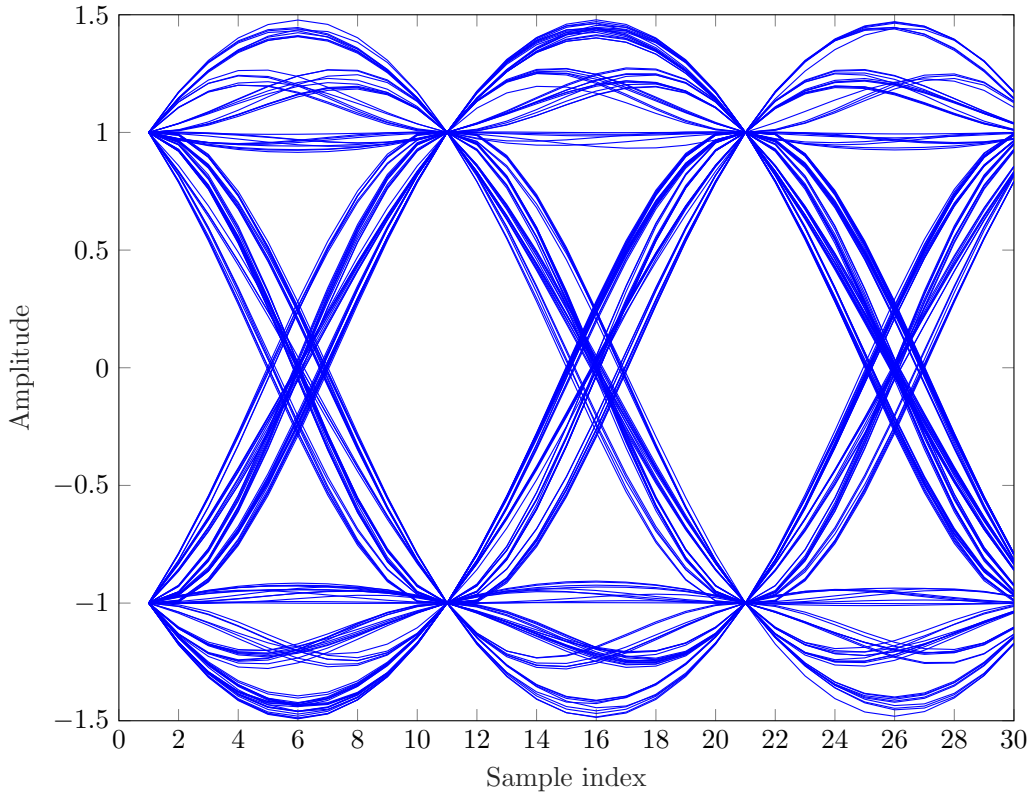


Figure 4.9: Eye Diagram of Raised cosine pulse $\beta = 0.5$

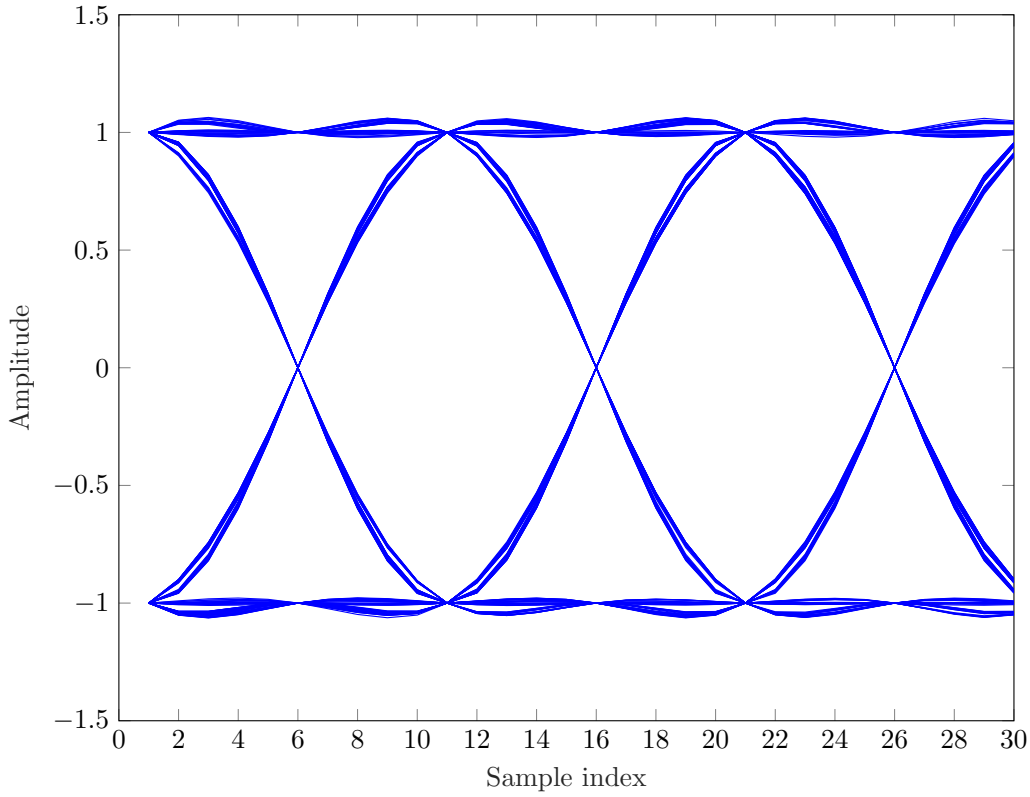


Figure 4.10: Eye Diagram of Raised cosine pulse $\beta = 1$

Since the pulse rate itself requires 50Hz of bandwidth for this transmission, along with this excess bandwidth of 25Hz, the Fig:4.7 represents the above scenario in frequency domain utilizing 75Hz bandwidth. Furthermore, when the roll-off factor is increased the tail starts to decay faster. In Fig:4.6, the raised cosine pulse is represented with $\beta = 1$. The decay in the side lobes is the key in this thesis because, it reduces the influence of ISI in the resulting signal and this pulse requires an excess bandwidth $\Delta B = 50Hz$, resulting in the total bandwidth of 100Hz. The frequency domain representation of this scenario is shown in Fig:4.8 and followed by the eye diagram of this raised

cosine pulse in Fig:4.10.

4.2 Modulator

In digital communication, modulation is a process of varying one or more properties of a periodic waveform. In this thesis, we mostly deal with the amplitude. Amplitude modulation is a most commonly used technique for transmitting information using radio carrier wave. In AM, the amplitudes of the pulse are defined in a set called *constellation* $\mathcal{A}$, which is in turn defined by the Modulation $\mathcal{M}$, which is the number of elements in the constellation. Based on the modulation parameter the bit density is defined as the number of bits a pulse carries based on the amplitude modulation, and it is an exponential function and its unit is bits/Hz-s. We have two such modulation techniques implemented in this thesis, which are Binary Phase Shift Keying (BPSK), 4-Pulse Amplitude Modulation.

4.2.1 Binary Phase Shift Keying

BPSK is a digital modulation technique which transmits data by changing the phase of the signal at a precise time. It is the simplest form of phase shift keying which uses two phases that are separated by 180. It is also called as 2-PSK.The BPSK constellation used in this thesis is [-1,1], and the Modulation $\mathcal{M} = 2$ and it modulates at 1 bit/symbol as shown in the Fig:3.1.

Bit Error Probability of BPSK

The probability of bit error of BPSK under additive white Gaussian noise can be represented as

$$P_b = Q(\sqrt{(2E_b)/N_0}) \tag{4.6}$$

where

$$Q(x) = \int_x^\infty 1/2\pi e^{-u^2/2} du \tag{4.7}$$

4.2.2 4-Pulse amplitude modulation

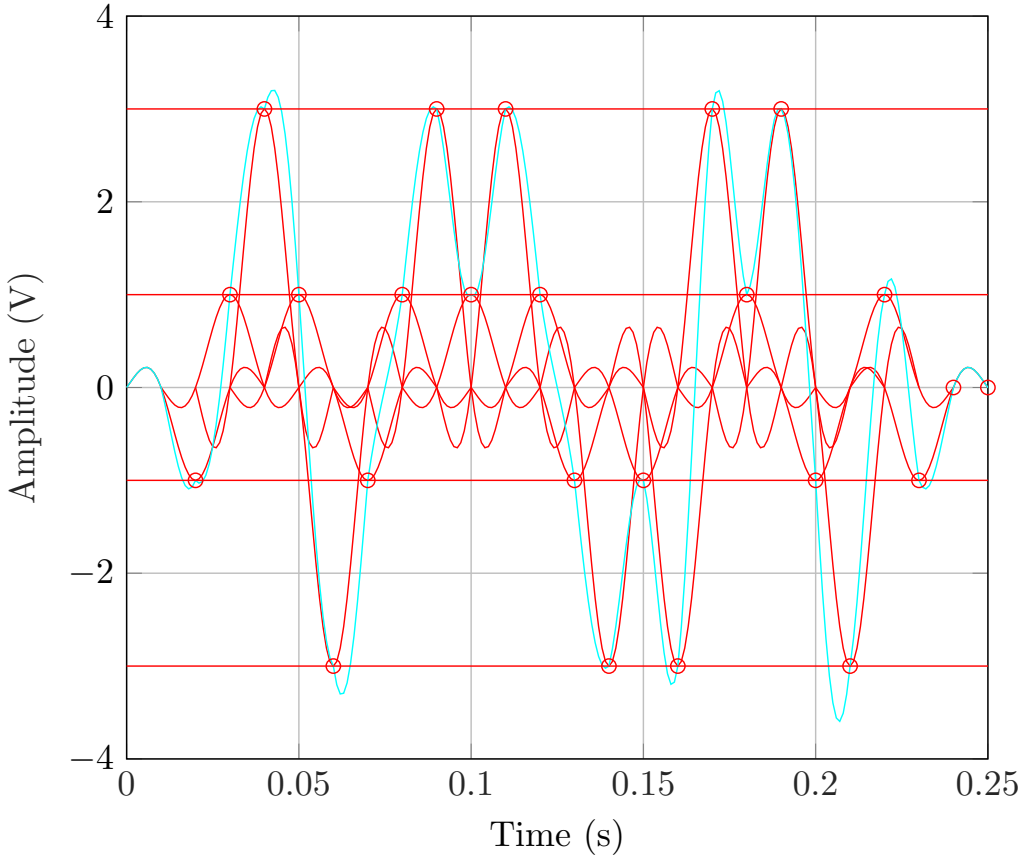


Figure 4.11: A train of 20 "sinc" pulses with 4-PAM constellation $\mathcal{A} = [-3, -1, 1, 3]$, $R_b = 100B/s$, and pulses of duration 0.01 s and $D = 4$. The red curves are individual pulses, while the cyan curve is $s(t)$, the sum of all pulses. The circles indicate the sampling points, where $s(kTp) = a_k$.

Pulse amplitude modulation is a modulation technique where the amplitude of the pulses is modified to transmit information. 4-PAM, as the name suggests it has 4 different amplitudes which in turn is the modulation index M. The constellation used for 4-PAM in this thesis is $\mathcal{A}$= [-3, -1, 1, 3] . This

modulation technique carries two bits per pulse, so it has a better data rate. A sequence of Sinc pulses modulated to 4-PAM with the above constellation is shown in Fig: 4.11

Bit Error Probability of 4-PAM

The probability of bit error of 4PAM under additive white Gaussian noise can be represented as

$$P_e = Q(\sqrt{(2E_p/5N_0)}) \tag{4.8}$$

4.3 AWGN Channel

Additive White Gaussian noise is the simplest noise model used in communication theory due to its nature. As its name suggests, this noise could be added to any information system that might be intrinsic, and it has uniform power across the frequency spectrum, and it is uniformly distributed in the time domain with an average value of zero. It has power equivalent to the variance of the Gaussian density. If N_o is the noise power, the variance σ^2 is represented as

$$\sigma_n^2 = N_0/2 \tag{4.9}$$

So the probability of error can be calculated as

$$P_e = Q(\sqrt{a_k^2/\sigma_n^2}) = Q(\sqrt{SNR}) \tag{4.10}$$

4.3.1 Channel Capacity and Bandwidth Efficiency

Channel capacity C is a measurable rate in bits/sec which is the maximum amount of information an AWGN channel can carry with low error probability. Shannon derived an equation to calculate the channel capacity. He stated that channel capacity is an ultimate limit that it is possible to approach the maximum capacity by sophisticated systems. However, it is impossible to carry information above the capacity with low error probability.

If it is a band-limited system with bandwidth W Hz, power is limited in Watts, and the noise power spectral density is N_o in watts/Hz, then the capacity for a AWGN channel is given by,

$$C = W \log_2(1 + (P/N_0W)) \tag{4.11}$$

The capacity C increases monotonically with W and reaches its maximum value of P/N_0log_2e as $W \to \infty$. If data rate is defined as R

$$R \leq Wlog_2(1 + (p/N_0W)) \tag{4.12}$$

4.4 Matched Filter

Matched filters are optimum linear time-invariant filters that increases the SNR and decreases the error probability. Matched filters are implemented using a known signal as the reference signal so that it could be compared to the unknown signal coming out of the AWGN channel added with noise. This is done by using a time-reversed form of the reference signal and the convoluted received signal. This filter tends to provide maximum possible

instantaneous SNR for the signal added with Gaussian noise[31].

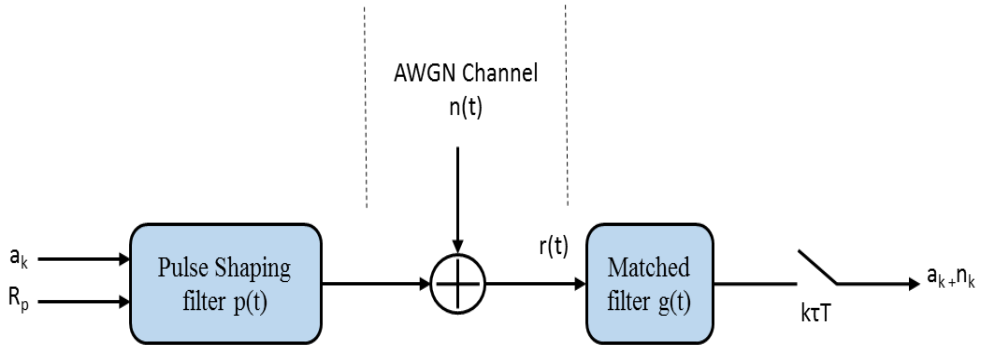


Figure 4.12: Block Diagram of a Matched Filter

According to Eq:4.13, the received signal is the addition of the transmitted signal and the noise added in the AWGN channel. The idea is to minimize the effect of the noise in the received signal to recover the transmitted signal, which is done in an optimal manned by the matched filter. Usually, the output signal power is desired to be higher than the noise power which results in maximizing the SNR. The filter gives a sharp peak response to the desired pulse at the input which helps to determine the amplitude of the pulse at a particular sampling duration which will be discussed in the next section.

The general matched filter representation in shown in Figure 4.12. For the transmitted signal $s(t)$ and $n(t)$

$$r(t) = s(t) + n(t) \tag{4.13}$$

The impulse response of the pulse shaping filter and the matched filter being $p(t)$ and $g(t)$ respectively, and output of matched filter being $y(t)$ is represented as

$$y(t) = r(t) * h(t) \tag{4.14}$$

Since the filter is linear, the output can be represented as

$$y(t) = s(t) * h(t) + n(t) * h(t) = x_0(t) + n_0(t) \tag{4.15}$$

Generally Signal to Noise ratio is defined as

$$SNR = a_k^2 / w_k^2 \tag{4.16}$$

and it is depended on g(t), If $g(t) = p(t)$, then SNR is maximum.

4.5 Sampler

Sampling is a crucial process in the receiver that determines the performance of the communication system. In Nyquist signaling, the received signal is sampled every T_p duration. But in FTN signaling, the signal is sampled at τT_p duration as the pulses are accelerated by the time acceleration factor τ. In Figure:3.2, the small red bubbles represent the samples of the signal and you can notice that, in the first figure the signal is sampled at every $T_p = 0.01$ duration, while the same set of pulses are transmitted in the second figure with $\tau = 0.8$, so the signal is sampled at $\tau T_p = 0.008$ duration.

Chapter 5

FTN Decoder

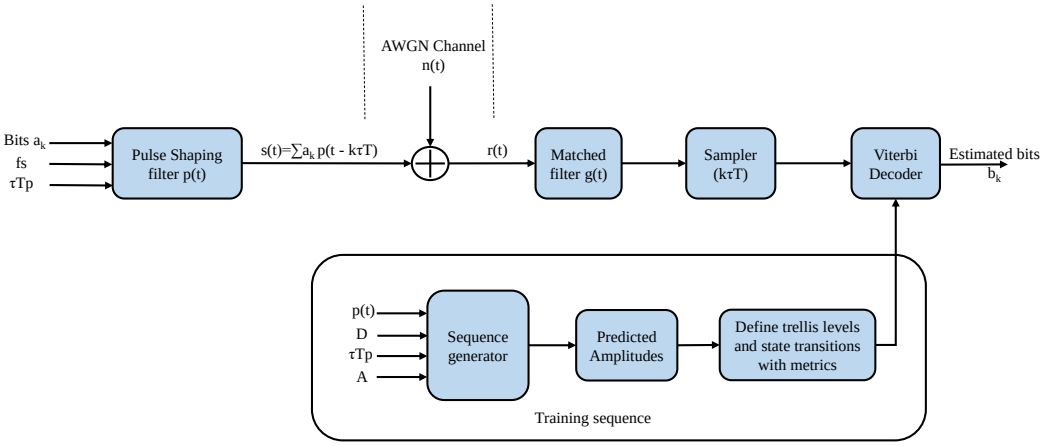


Figure 5.1: FTN Decoder using trellis decoding.

In the previous two chapters, we discussed how digital information is transmitted across an AWGN channel and received by a receiver, both in a traditional sense and Faster than Nyquist. When the sinc or raised cosine pulses are transmitted at Nyquist rate without inter-symbol interference, the signal is only influenced by the white Gaussian noise and maybe few more factors,

but we do not consider them in this thesis. In this case, the received signal can be sampled at T_p duration. These estimated information bits can be decoded from the sampled channel output by a decision rule, where you can set up a threshold based on the average of the minimum distance.

When the pulses are transmitted at a rate faster than Nyquist, the received signal has to be sampled at τT_p duration, and the samples are not only influenced by the noise, but also with the ISI which tends to encode the samples into a trellis structure. The received samples r_k contain information not only about a_k, but also about some of the symbols that are transmitted before and after it. Due to this dependency and the trellis structure, the estimated information bits can be decoded using a trellis decoder like Viterbi or BCJR. If $tau \geq \mu$, the Euclidean distance between symbol sequences is the same as in the Nyquist signaling and it is the distance which determines the probability of error. We use Viterbi decoder in this thesis, in order to decode the bits, due to its elegance and its less complexity. Before diving into the way, the Viterbi decoder used in this context, a little background on Viterbi decoder is provided.

5.1 Viterbi Decoder

The Viterbi decoder uses Viterbi algorithm for decoding a bitstream that has been encoded using a trellis code. It was proposed by Andrew Viterbi in 1967 [10]. The algorithm was initially designed to decode the convolution code, but later on, many inventions were made on it and its application started to grow. It is used for finding the most likely sequence of hidden states called

the Viterbi path.

The Viterbi decoding algorithm uses two variables to decode the estimated information bits: the branch metric and the path metric. The branch metric is a measure of the distance between what was transmitted and what was received, and it is defined for each transition in the trellis. The trellis is unfolded in each time instance a sample is received, and the number of states of the trellis is defined by 2^{k-1}, where k is the constraint length of the convolution code. The transitions tend from the current state to the next state based on the input bit, and a branch metric is associated with each transition. In every time instant, the euclidean distance between the received bit and the expected bit is calculated and, it is called the branch metric of a transition.

The path metric is associated with a state in the trellis or a value associated with each node. As the trellis proceeds, the path metric corresponds to the euclidean distance with respect to the received bit sequence over the most likely path from the initial state to the current state in the trellis. By most likely path, we mean the path with the smallest euclidean distance between the initial state and the current state, measured over all possible paths between the two states. Depending on the number of inputs, the number of transitions converges to a particular node. The node metric is calculated by the modulus of sum of the previous node metric and the branch metric of the transition. The path with the smallest euclidean distance minimizes the total number of bit errors and is most likely when the BER is low. According to the Viterbi algorithm, the receiver can compute the path metric for a state and time incrementally using the path metrics of previously computed states

and the branch metrics.

In case the receiver has computed the branch metric for each state s at time step I, the value of the branch metric is the total number of bit errors detected when comparing the received bit to the most likely transmitted message considering all the bits that could have been sent by the transmitter until time i. Among all the possible states at time i, the most likely state is the one with the smallest branch metric. If there is more than one such state, they are all equally good possibilities. In order to determine the path metric at the time $i + 1$ for each state s, we have to notice that if the transmitter is at state s at time i+1, then it must have been in only one of two possible states at the time i. More details about the branch metric and node metric calculation will be explained in the coming sections with an example.

5.1.1 Determining the most likely path

The main loop of the algorithm contains two major steps, the first one is to calculate the branch metric for the transitions, and the next is to compute the path metrics as described before. Let's see a step by step approach of determining the path metrics. 1) Add the branch metric to the path metric for the old state. 2) Compare the sums for the paths arriving at the next state (The number of paths approaching a particular state depends on the number of incoming bits) 3) Select the path with the smallest value, breaking ties arbitrarily. This path corresponds to the one with fewest errors.

Once all the path metrics are determined for every time instant, the path metric of all the states at the last time instant is compared to trace the most likelihood path. The state with the minimum path metric is selected

and traced back to the state with the minimum branch metric that led to the current states path metric, and it is traced back so forth unto the first state. The survivor path is one that has a chance of being the most likelihood path. The Viterbi algorithm is so practical as the number of survivor paths is much smaller than the total number of paths in the trellis. In any case, if the paths tend to have the same metric, it is considered most likely the ties would break, and the metric will change after a certain time step based on the decoder's future knowledge.

5.1.2 Decoding FTN signals using Viterbi decoder

Based on the background in the Viterbi decoder, we know all we need to decode the transmitted information sequence is the trellis and the branch metric. As discussed earlier, the transmitted bit sequence $\{a_k\}$ constrained to the *constellation* $\mathcal{A}$, is transmitted using a pulse shape $p(t)$ at τT_p interval. The transmitted FTN signal is represented as

$$s_{FTN}(t) = \sum_k a_k p(t - \tau k T_p), \tag{5.1}$$

When the signal comes out of the AWGN channel and passed through the matched filter, it is sampled at τT_p intervals and the channel output is determined. Our significant contribution in this thesis is to present an algorithm to generate the trellis of an FTN signaling scheme automatically. The algorithm that is implemented for simulation will be explained in detail in the next chapter. In this section, we will discuss an example of how the trellis is generated and the process of decoding the FTN signals based on

that trellis.

In order to generate the trellis, all we need is the pulse shape, the signaling rate and the constellation. In the Figure:5.3, we have a sequence of 11 bits $a_k = [1, 1, 1, -1, -1, -1, -1, 1, -1, 1, 1$ (shown in red, -1 added in either side for padding) constituting the *constellation*) $\mathcal{A} = -1, 1$ with modulation index $M = 2$. The pulse shape used in this example is a raised cosine pulse with $\beta = 1$ and their side lobes are truncated with $D = 2$. As you can notice in the figure that the pulses are transmitted faster than Nyquist rate with $\tau = 0.9$. The pulse interval is supposed to be 0.01 in Nyquist signaling, but it is $\tau T_p = 0.01 * 0.9 = 0.009$, due to which the pulses are interfered with the adjacent pulses at their sampling duration τT_p causing ISI. In the figure, at the duration 0.02, the pulse is interferes with 2 pulses transmitted at 0.01 and 0.03. The plot in cyan represents the signal $s(t)$ and the blue lines are plotted to mark the possible amplitudes the signal could take during the sampling duration. Due to the interference of two pulses, the amplitude of the signal at a sampling duration could have 8 possible values ($M^(D + 1)$, $2^(2 + 1) = 8$. With this short sequence of pulses, we have covered all possible combinations.

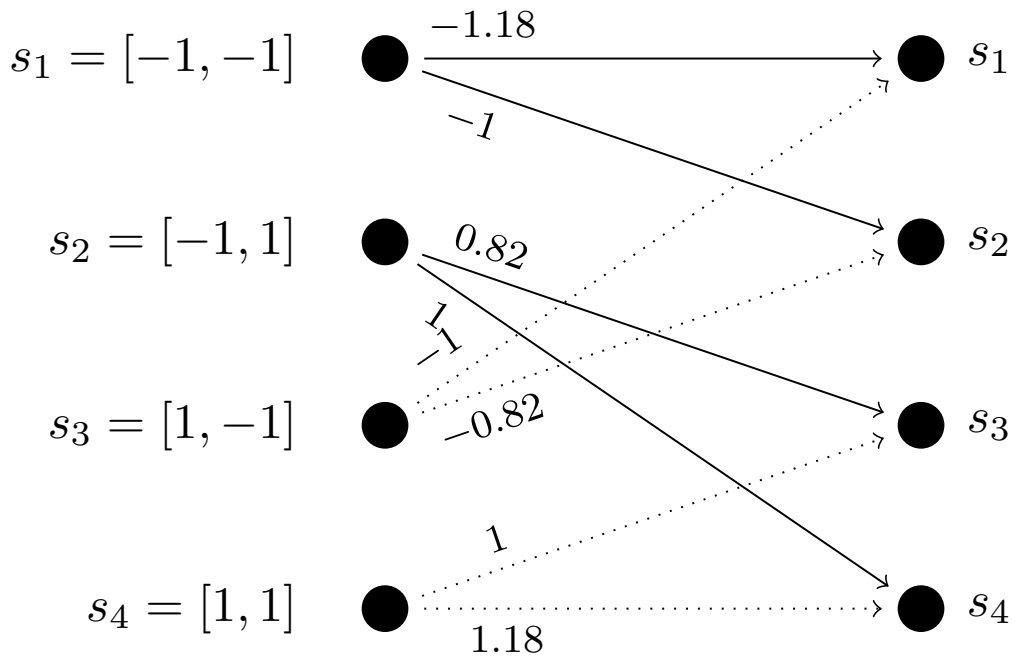


Figure 5.2: Trellis of a $D = 2$ FTN signal, using a BPSK constellation and raised-cosine pulses with roll-off factor 0.5. The solid lines represent an input "-1" and the dotted lines an input "1".

With that being said, the information is extracted from this plot to generate the trellis. The main idea behind the algorithm is to calculate all possible combinations of the transmitted pulses and incoming pulses along with the amplitudes variations of the center pulse caused by the ISI of the adjacent pulses. As discussed earlier, all we need to generate a trellis is the number of states, current state, input, next state and branch metrics. The trellis represented in the Figure:5.2 is generated from the pulses represented in the Figure:5.3. Consider a sample at time t that is interfered with pulses t-1 and t+1, the pulses t-1 and t is the current state and t+1 is the input and t and t+1 becomes the next state and the amplitude sampled at the time t is the

branch metric.

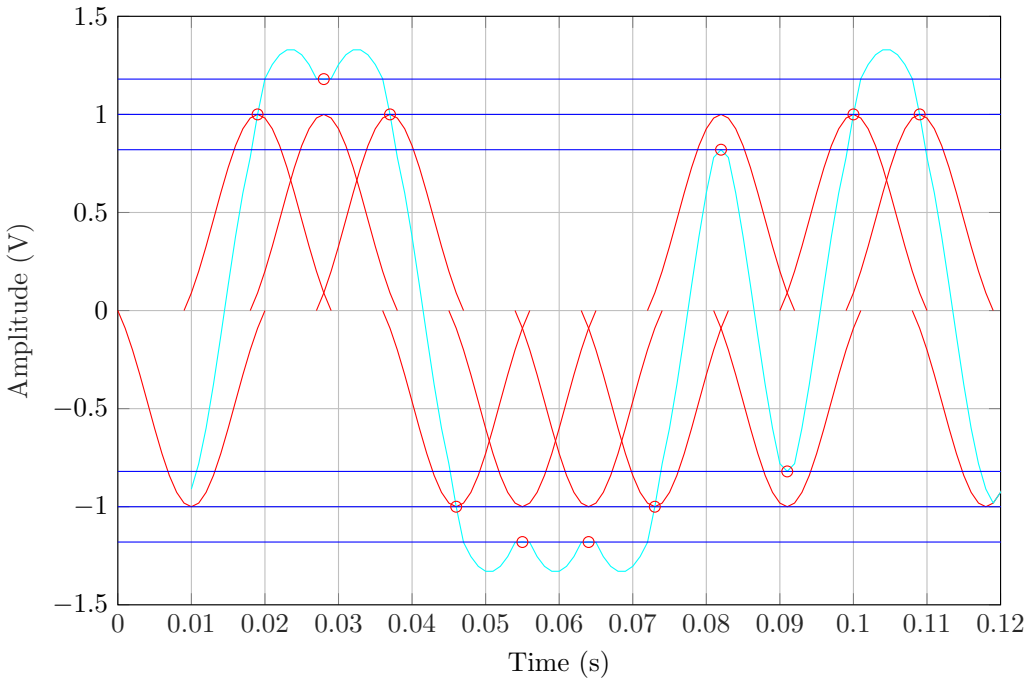


Figure 5.3: Sinc pulses to demonstrate trellis generation and Viterbi decoding

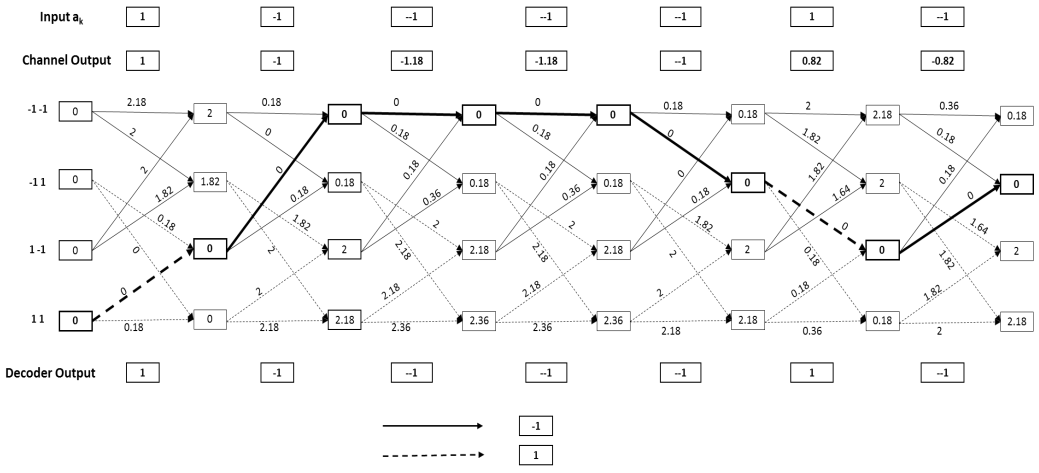


Figure 5.4: Most likely path computation using Viterbi decoder

Let us see an example that leads us to the generation of the trellis from the plot. In Figure:5.3, at time t=0.02, the pulse transmitted is +1 preceded by -1 at $(t-1)$, so the current state becomes [-1,1], and the input is +1 that is transmitted at t+1, so the next state is [1,1] and the sample captured at time t is 1 which is the branch metric. In Figure:5.2, the current state corresponding to this example is s2 and the transition that is leading to s4 which is the next state, the solid line represents the input 1 from the time t which is unlike the traditional Viterbi algorithm. Similarly, let us see another example with the pulse transmitted at t=0.09 interfered with the pulses t-1 and t+1. The current state is [1,-1] which is state s3 in the trellis and since the incoming pulse is 1, the next state is [-1,1], so the transition leading from s3 to s2 is a dotted line since the pulse transmitted at t is -1 and the sample of $s(t)$ at time t is -0.82 which is the branch metric of this transition.

Figure:5.4 is the demonstration of trellis decoding for the signal transmitted in figure:5.3. The trellis is represented only for a segment of the signal from the time instance 0.04 to 0.1. The process of trellis decoding comprises of three steps, calculating the branch metrics, calculating the node metrics and finally, tracing the survival path to decode the bits. These steps will be discussed in detail in the following sections

Calculating the branch metrics

The trellis shown in the Figure:5.4 is unfolded for every time instant mentioned earlier. The branch metric is calculated for every transition of every time instant. The channel output in the figure represents the samples taken from the received signal. The channel output is subtracted with the branch

metric from the model trellis shown in the Figure:5.2 and the modulus of the result is fed to the corresponding transitions in the decoding trellis. Since the signal represented here is only influenced by ISI and not any noise, at least one of the branch metric must be zero as the euclidean distance will be zero. In the last time instant, the transition leading from state s3 to s2 has a branch metric 0, as the current state is [1,-1] and the next state is [-1,1], the branch metric in the model trellis, and the channel output is -0.82 whose euclidean distance is 0.

Calculating the node metrics

Once the branch metrics are computed for all the transition in the trellis, the node metrics are calculated. In this context, the node corresponds to the states and the four nodes in every time instant have two transitions converging at each one of them. To calculate the node metric for a particular node, the minimum value of the sum of the node metric of the previous state and the branch metric of the transition leading from the previous state and the current state is computed. For example, consider the state s1 in the last time instant has two transitions from previous state s1 and s3. The sum of node metric and the transition leading from s1 is 2.54 and the one leading from s3 is 0.18 which is minimal, hence assigning that as the node metric of the current state.

Tracing the survival path and decoding the estimated bits

The process of tracing the survival path can be performed once all the node metrics are computed or even for a segment of time which will reduce the

processing requirements and memory capacity. It starts by picking the least of the node metrics in the last time instant and working our way backward by traversing through the transition that led to the node metric of the node. The nodes and the branches are traced back to the beginning of the trellis. For example, the node metric of state s2 in the last state is the least. It was computed by picking the branch that was led from s3 from the previous state and this node metric was in turn computed by selecting the branch led from the state s2 from the previous state and likewise. Once the survival path is identified, the final step is to decode the pulses in every time instant. A mentioned earlier the solid line represents -1 and the dotted represents 1. The decoder then identifies the branches of the survivor path in every time instance, and the decoder output is computed. The input a_k and the estimated decoder bits b_k is them compared to calculate the signal to noise ratio.

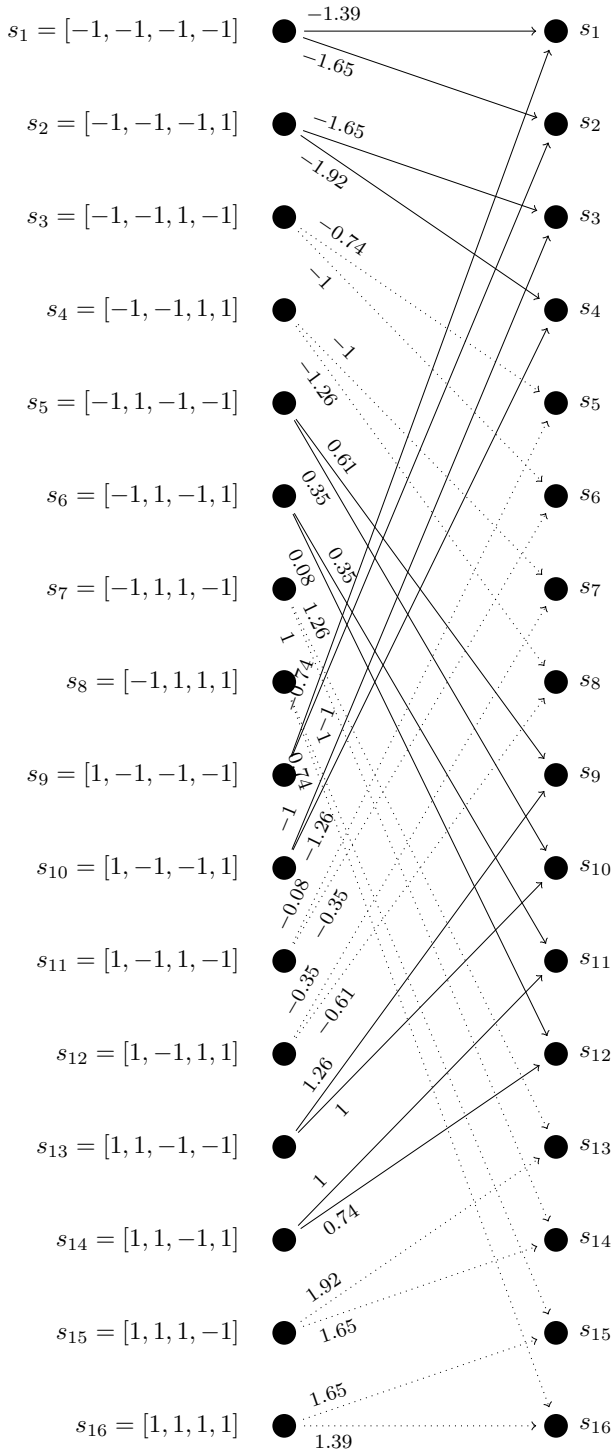


Figure 5.5: Trellis of a $D = 4$ FTN signal, using a BPSK constellation and raised-cosine pulses with roll-off factor 0.5. The solid lines represent an input "-1" and the dotted lines an input "1".

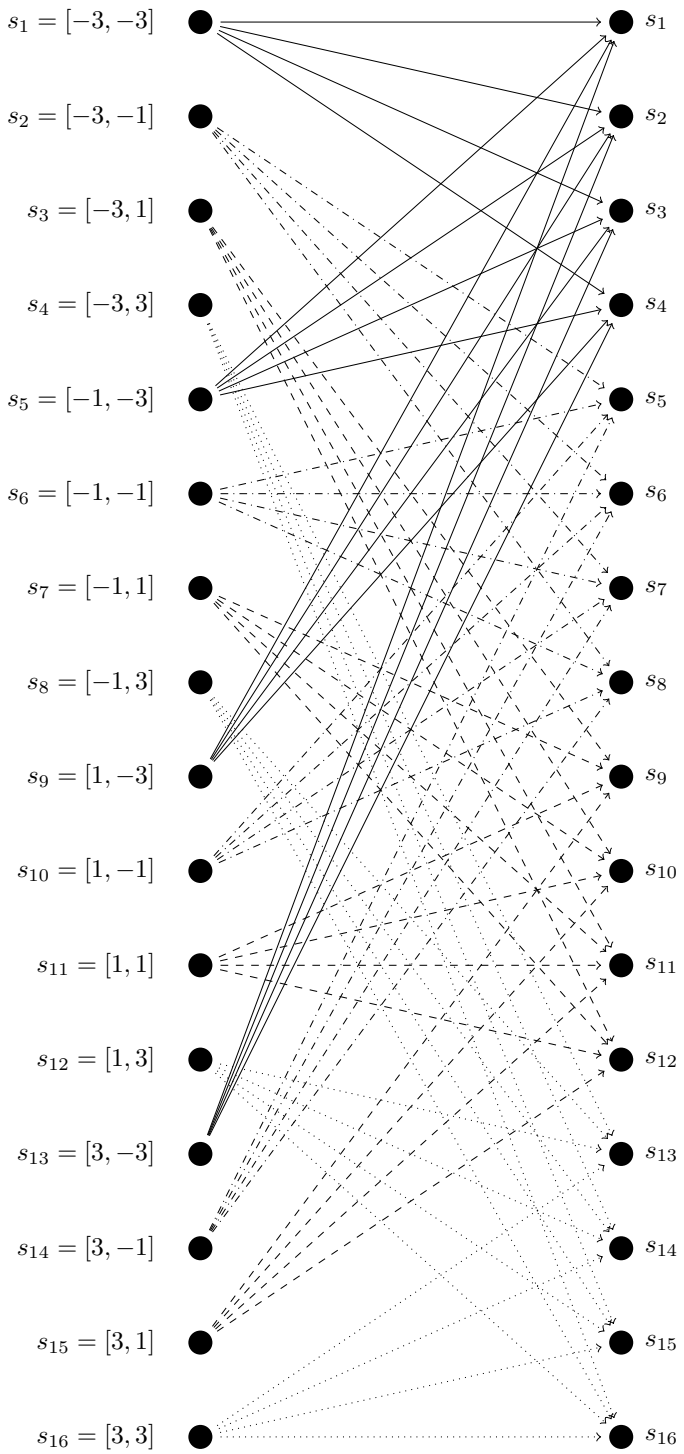


Figure 5.6: Trellis of a $D = 2$ FTN signal, using a 4-PAM constellation and raised-cosine pulses with roll-off factor 0.5. The solid lines represent an input "-1" and the dotted lines an input "1".Branch metrics are not displayed due to space constrain

In Figure:5.5, the trellis of a D = 4 FTN signal, using a BPSK constellation using a raised cosine pulse with $\beta = 0.5$ is represented. Due to the value of D, four pulses two from the front and two from the back will be interfering with the concerned pulse which results in a 16 level trellis two transitions leaving and converging at every state. The branch metrics are the samples from the s(t) captured at the center pulse at τT_p duration. Similarly Figure:5.6 represents the trellis for 4-PAM constellation using the similar raised-cosine pulses and D = 2. In this case, two pulses one from each side will be interfering with the center pulse, but due to the number of elements in the constellation, there will be four possible inputs for every state. Similarly, the branch metrics are sampled from the $s(t)$ captured at the center pulse at τT_p duration.

5.1.3 Determining the Euclidean distance for Nyquist and Faster than Nyquist signaling

In this section, we are going to consider a sequence of bits transmitted, both in Nyquist signaling and Faster than Nyquist signaling and calculate the euclidean distance between the transmitted and received bits. This example is performed to prove the statement we made earlier that, the euclidean distance is same for Nyquist signaling and Faster than Nyquist siganling up to a certain τ values for different pulses.

In this example, we considered a transmitted sequence $a_k = [-1 - 1 - 1111 - 1 - 1]$, in which the first and the last -1 is used for padding. In table:5.1, the first three rows represents the transmitted sequence and the next three for the received sequence b_k. Since we considered $D = 2$ and

$\tau = 0.9$, the trellis presented in Fig:5.2 corresponds to this scenario. By keeping the trellis as reference, the current states have been mapped in the second row for every corresponding transmitted bit including the previous and the next bit. For instance the first state is s1 since the current state is $[-1 - 1]$ considering the padded bit and the input is -1, so the next state is obviously $[-1 - 1]$. In the second instant the current state is $[-1 - 1]$ and the input is 1, which leads to the state s2 and so forth. In the third row, the branch metrics are determined from the trellis based on the transition from one state to another.

Table 5.1: Calculating Euclidean distance for Nyquist and Faster than Nyquist signaling

Transmitted sequence a_k	-1	-1	1	1	1	-1	
Actual states	s1	s2	s4	s4	s3	s1	
Branch metric of the transmitted sequence	-1.18	-1	1	1.18	1	-1	
Received sequence b_k	-1	-1	-1	1	1	-1	
Decoded states	s1	s1	s2	s4	s3	S1	
Branch metric of the received sequence	-1.18	-1.18	-1	1	1	-1	
Euclidean distance of FTN Signaling	0	-0.18	2	0.18	0	0	2
Euclidean distance of Nyquist Signaling	0	0	2	0	0	0	2

In the received sequence, we have deliberately introduced an error in the third bit. considering that the decoded states will have a change of survivor path. The decoded states and the appropriate branch metrics are determined from the trellis by similar methods we used on the transmitted sequence. The euclidean distance of FTN signaling is calculated by measuring the displacement in the branch metrics in each state. Due to the error the transitions

between the second, third and fourth instances have changed which resulted in the displacement that is captured in the row 7. The calculation of euclidean distance of Nyquist signaling is straight forward, by measuring the displacement between the transmitted and the received sequence. As you can notice the cumulative euclidean distance of both Nyquist and Faster than Nyquist signaling are same.

Chapter 6

Algorithm

Algorithm 1 Trellis generation for FTN signals

Input: D, $\mathcal{A}$, τ, T_p, interpolator $p(t)$
Output: Trellis matrix $\mathbf{T}$

Let $M = |A|$
Let $nst = M^D$ {number of trellis states}
Let $rowindex = 1$
Let $\mathbf{S} = \mathcal{A} \times \mathcal{A} \times \ldots \times \mathcal{A}$ {Repeated D times}
Let $\mathbf{t}_{ISI} = \tau\left(\frac{-D}{2} : \frac{D}{2}\right)$
for $sindex = 1$ **to** nst **do**
    $\mathbf{cs} = \mathbf{S}[sindex]$
    for $i = 0$ **to** $M - 1$ **do**
        $\mathbf{T}[rowindex + i, 1] = \mathbf{cs}$
    end for
    for $cindex, input$ in enumerate($\mathcal{A}$) **do**
        $\mathbf{T}[rowindex + cindex - 1, 2] = input$
        $\mathbf{ns} = [\mathbf{cs}[2 : end], input]$ {Next state}
        $\mathbf{T}[rowindex + cindex - 1, 3] = \mathbf{ns}$
        {Branch metric calculation}
        $\mathbf{a} = [\mathbf{cs}, input]$
        $bm = 0$
        for $(tindex, t)$ in enumerate($\mathbf{t}_{ISI}$) **do**
            $bm = bm + \mathbf{a}[tindex]p(t)$
        end for
        $\mathbf{T}[rowindex + cindex - 1, 4] \leftarrow bm$
    end for
    $rowindex = rowindex + M$
end for

The algorithm presented in Alg:1 used the following conventions.Indexable arrays or vectors are written in bold face. The constellation $\mathcal{A}$ is considered indexable. The iterator enumerate($\mathbf{a}$) returns a tuple consisting of the index and the current value of $\mathbf{a}$. The symbol : is used to represent a range in the usual way. Text between brackets {} are comments. The pulse $p(t)$ is implemented as an interpolator that can return the pulse amplitude for any t.

The generation of the trellis is the most important part of this thesis and the algorithm presented above facilitates that purpose. In this chapter we will be discussing the logic behind the algorithm and its implementation. The algorithm takes four important variables as input, first of all the constellation $\mathcal{A}$ which is determined by the modulation technique and the minimum euclidean distance. The pulse shape $p(t)$ with pulse duration T_p is the second parameter. The variable D represents the number of side lobes of the pulse and it ideally determines the structure of the trellis and finally the time compression factor τ. The output of the algorithm will be a matrix with four columns, the current state, the next state, input and the branch metrics.

The below table lists the number of trellis states and number of branch metrics each combination of these variables results in.

Table 6.1: Calculating number of trellis states and number of branch metrics

Modulation index M	Number of Interfering pulses D	Number of Trellis states $T=M^D$	Number of branch metrics $B=M^{D+1}$
2	2	4	8
2	4	16	32
2	6	64	128
4	2	16	64
4	4	256	1024
4	6	4096	16384

We have simulated this algorithm in Julia and Matlab and the Alg:1 represents the Julia implementation. In the next section we will see a couple of examples of the code execution on both Julia and Matlab implementations that will lead to the generation of the branch matrix. As mentioned above, this branch matrix helps us to generate the trellis presented in the previous chapter.

6.1 Matlab Implementation

In this section, we will see an example of the Matlab implementation step by step, right from declaring the input variables with appropriate values, followed by execution of the main function that leads to the output branch matrix.

In this example, we are considering a basic scenario of $D = 2$, $\tau = 0.9$ and a BPSK *constellation* $\mathcal{A} = [-1, 1]$. The table:6.2 initializes all the necessary input variables required for the generation of the branchmatrix.

Table 6.2: Matlab Input variables

Variables	**Value**	**Description**
Constellation A	[-1,1]	Possible amplitudes of the symbols
M	2	Modulation parameter (No of elements in the constellation)
D	2	Number of interfering pulses
β	0.5	Roll-off factor
τ	0.9	Time acceleration factor
f_s	1000	Sampling frequency
T_p	0.01	Pulse duration
Number of states = M^D	4	Number of states of the trellis.
Number_of_branchmetric = M^{D+1}	8	Each state has two inputs (-1,1) that leads to the next state.
[p] = rcpulse(β,D,Tp,Ts,'rc')	pulse(p)- raised cosine	The function rcpulse takes β,D,Tp,Ts as inputs and generates a pulse p
p = p./max(p)	-	The pulse p is normalized to amplitude 1

branchmatrix(1,1) = cellstr('Currentstate');
branchmatrix(1,2) = cellstr('Input');
branchmatrix(1,3) = cellstr('Nextstate');
branchmatrix(1,4) = cellstr('Branch value');
	branchmatrix(1,4)	The output branch matrix is initialized. It produces a empty table with the titles in the first row

Once all the variables are initialized, the main logic of the code begins with a for loop.

$fork = 0:1:(numberofbranchmetric) - 1$, where $k = 0:1:7$, iterates for 8 times in this case. In the next section we will see two iterations of k as example, that will generate two rows in the branch matrix and then the final branch matrix output is presented

6.1.1 Iteration:1, where $k = 0$

Table 6.3: Iteration 1

Code	Result	Description
b=deci2bin(k,L)	b=deci2bin(0,3) = 000	Binary representation with L bits
a_k = b*2-1	[-1 -1 -1]	Converting b to [-1 1], input sequence
impulses = zeros(1,length(a_k)*f_s*T_p*τ) impulses(1:fs*T_p*τ:end) = a_k	impulses(1x27)	Impulses is initialized with sequence of zeros and the input sequence ak is inserted every (f_s*T_p*τ) position
s = conv(impulses,p)	Signal (s)	The impulses is convoluted with the pulse p to generate the signal s
Outputs = s(start_samp:f_s*T_p*τ:end); Outputs = Outputs(1:length(a_k))	[-1.09, -1.18, -1.09]	The transmitted signal is sampled at τT_p duration
branch_value(k+1)=Outputs((D+2)/2)	Outputs(2) = -1.18	The sample value of the centre pulse is the branch metric for a_k input sequence
branchmatrix(k+2,1) = cellstr(num2str(b(1,1:D)));	branchmatrix(2,1)= -1 -1	Current state
branchmatrix(k+2,2) = cellstr(num2str(b(D+1)));	branchmatrix(2,2)= -1	Input
branchmatrix(k+2,3) = cellstr(num2str(b(1,2:D+1)));	branchmatrix(2,3)= -1 -1	Next state
branchmatrix(k+2,4) = cellstr(num2str(branch_value(k+1)));	branchmatrix(2,4)= -1.18	Branch metric

In the end of the first iteration, the first row of the branchmatrix will be populated with the data generated in the last four rows of table:6.3.

6.1.2 Iteration:2, where $k = 1$

Table 6.4: Iteration 2

Code	Result	Description
b=deci2bin(k,L)	b=deci2bin(1,3) = 001	Binary representation with L bits
a_k = b*2-1	[-1 -1 1]	Converting b to [-1 1], input sequence
impulses = zeros(1,length(a_k)*f_s*T_p*τ) impulses(1:fs*T_p*τ:end) = a_k	impulses(1x27)	Impulses is initialized with sequence of zeros and the input sequence ak is inserted every (f_s*T_p*τ) position
s = conv(impulses,p)	Signal (s)	The impulses is convoluted with the pulse p to generate the signal s
Outputs = s(start_samp:f_s*T_p*τ:end); Outputs = Outputs(1:length(a_k))	[-1.09, -1, 0.91]	The transmitted signal is sampled at τT_p duration
branch_value(k+1)=Outputs((D+2)/2)	Outputs(2) = -1	The sample value of the centre pulse is the branch metric for a_k input sequence
branchmatrix(k+2,1) = cellstr(num2str(b(1,1:D)));	branchmatrix(2,1)= -1 -1	Current state
branchmatrix(k+2,2) = cellstr(num2str(b(D+1)));	branchmatrix(2,2)= 1	Input
branchmatrix(k+2,3) = cellstr(num2str(b(1,2:D+1)));	branchmatrix(2,3)= -1 1	Next state
branchmatrix(k+2,4) = cellstr(num2str(branch_value(k+1)));	branchmatrix(2,4)= -1	Branch metric

In the end of the second iteration, the second row of the branchmatrix will be populated with the data generated in the last four rows of table:6.4.

6.1.3 Branch Matrix

The for loop continues the remaining iteration to completely populate the branchmatrix. The final output of the branchmatrix for this example is shown in the table:6.5.

Table 6.5: Output Branch Matrix

Current State	Input	Next state	Branch metrics
-1 -1	-1	-1 -1	-1.18
-1 -1	1	-1 1	-1
-1 1	-1	1 -1	0.82
-1 1	1	1 1	1
1 -1	-1	-1 -1	-1
1 -1	1	-1 1	-0.82
1 1	-1	1 -1	1
1 1	1	1 1	1.18

This information is vital for the creation of the trellis represented in Fig:5.2. The trellis created using this branchmatrix is fed to the Viterbi decoder, which in-turn uses it to decode the channel output that was transmitted with the same input parameters.

6.2 Julia Implementation

In this section, we will see an example with the Julia implementation. The variables and the code used in this example will correspond to the algorithm presented in Alg:1. In order to arrive at the same output branch metric, the values assigned to the input variables are similar to that we used in the Matlab implementation. The variables are declared in the Table:6.6 and followed by the code implementation in the Table:6.7.

Table 6.6: Julia Input variables

Variables	**Value**	**Description**
Constellation A	[-1,1]	Possible amplitudes of the symbols
M	2	Modulation parameter (No of elements in the constellation)
D	2	Number of interfering pulses
β	0.5	Roll-off factor
τ	0.9	Time acceleration factor
f_s	1000	Sampling frequency
T_p	0.01	Pulse duration
nst = M^D	4	Number of states in the trellis.
rowindex	1	Enumerate the rows in the Trellis matrix T
S = A x A x .....x A (Repeated D times)	[-1 -1]	Every iteration produces the next two bit combination constrained to the elements in the constellation
t_{ISI} = τ(-D/2 : D/2)	(-0.9:0.9)	Calculate ISI timing instants
[p] = rcpulse(β,D,Tp,Ts,'rc')	pulse (p) – raised cosine	The function rcpulse takes β,D,Tp,Ts as inputs and generates a pulse p
p = p./max(p)	-	The pulse p is normalized to amplitude 1
Trellis matrix T	T(8,4)	The output Trellis matrix is initialized. It creates a empty table T to feed the current state, input, next state and the branch metrics

The main logic is slightly different in this case, the main for loop sindex iterates based on the number of states, which is four in this example, feeding two rows at once to the Trellis matrix T. Each row is handled by a nested for that iterates based on the *constellation* $\mathcal{A} = [-11]$.

Table 6.7: Julia Code Execution

Code	Result	Description
for sindex = 1 to nst do	Sindex = 1	The for loop iterates for four times creating two rows in the Trellis T for every iteration
cs = S[sindex]	cs = [-1 -1]	Computing current state

for i = 0 to M-1 do
T[rowindex + i, 1] = cs
end for
	T[1,1] = [-1 -1]
	T[2,1] = [-1 -1]
		This for loop iterates twice and feeds the current state for the first two rows of Trellis matrix T

for cindex,input in enumerate(A) do	Iteration:1 (cindex = 1, input = -1) Iteration:2 (cindex = 2, input = 1)	This for loop iterates twice and feeds the input, next state and the branch metric for first two rows of T
Iteration :1 (cindex = 1, input = -1)		
T[rowindex + cindex -1, 2] = input	T[1,2] = -1	Feeding Input to Trellis T
ns = [cs[2:end],input]	ns = [-1 -1]	Computing the next state
T[rowindex + cindex -1, 3] = ns	T[1,3] = [-1 -1]	Feeding next state to Trellis T
a = [cs, input]	a = [-1 -1 -1]	Merging current state and input

for (tindex,t) in enumerate(t_{ISI}) do
bm = a[tindex]p(t)
end for
	bm = -1.18	Generation of branch metrics.

T[rowindex + cindex -1, 4] = bm	T[1,4] = -1.18	Feeding branch metrics to Trellis T
Iteration :2 (cindex = 2, input = 1)		
T[rowindex + cindex -1, 2] = input	T[2,2] = 1	Feeding Input to Trellis T
ns = [cs[2:end],input]	ns = [-1 1]	Computing the next state
T[rowindex + cindex -1, 3] = ns	T[2,3] = [-1 1]	Feeding next state to Trellis T
a = [cs, input]	a = [-1 -1 1]	Merging current state and input

for (tindex,t) in enumerate(t_{ISI}) do
bm = a[tindex]p(t)
end for
	bm = -1	Generation of branch metrics.

T[rowindex + cindex -1, 4] = bm	T[1,4] = -1	Feeding branch metrics to Trellis T

In the end of one sindex iteration, the Trellis matrix T will have two rows populated and each row is generated by the cindex iteration as show in the Table:6.7. By the end of four sindex iterations, the Trellis matrix T will have 8 rows populated with values for current state, input, next state and the branch metrics. The Trellis matrix T generated by the Julia implementation is same as the branchmatrix presented in the Table:6.5.

Chapter 7

Simulation Results

We ran various simulations with different values of D (Number of lobes), τ (Time acceleration factor), β (Roll-off Factor) and M (Modulation Index) and made quite interesting observations. With the slight variation of any one of the above parameters, changes the entire simulation, like a new set of branch metrics are determined, and a new trellis is formed to decode the channel output. Below is the table that lists the above variables and the different values we used respectively.

Variable	Value
Modulation Index (M)	2(BPSK), 4(PAM-4)
Roll off factor (β)	0.5,0.75 (rc pulses)
Time Acceleration factor (τ)	0.7,0.8,0.9
Number of lobes (D)	2,4,6

Table 7.1: Simulations performed with different combination of variables

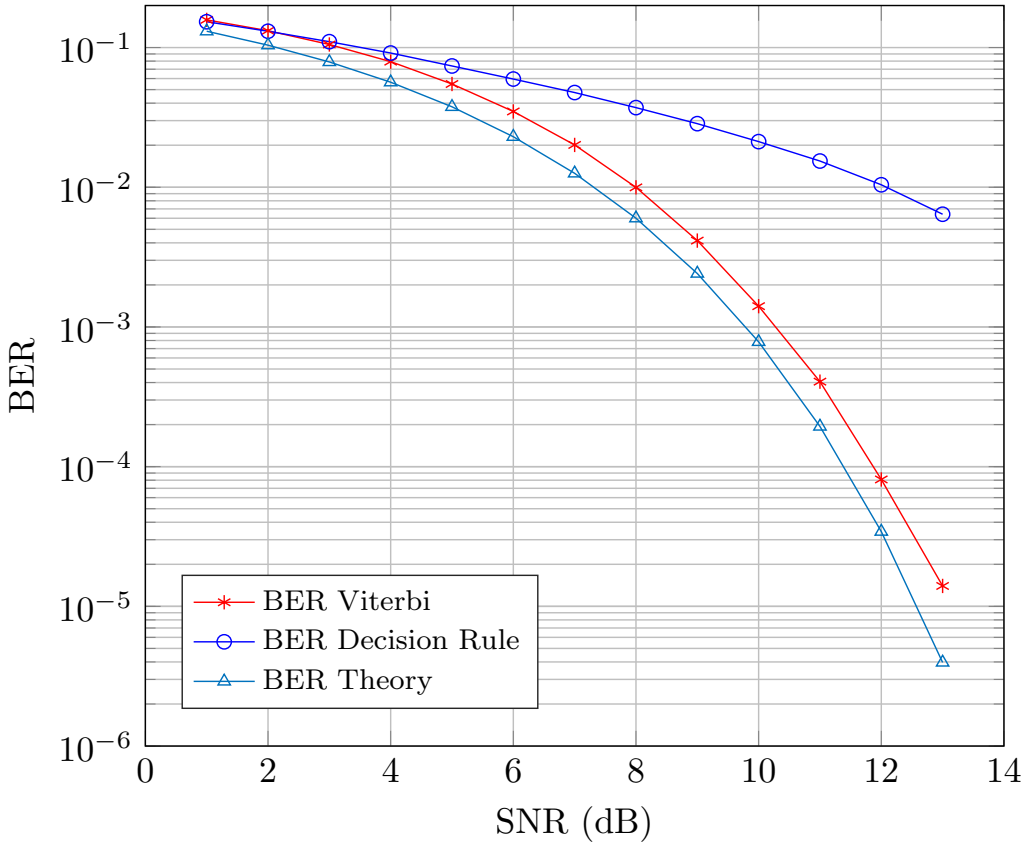


Figure 7.1: Bit error rate of an BPSK FTN signaling scheme with $D = 2$ and $\tau = 0.7$ and $\beta = 0.75$.

In Figure:7.1, we present the bit error rate of an FTN system with D = 2, $\tau = 0.7$ with BPSK constellation using a raised cosine pulse with roll-off factor $\beta = 0.75$. The plot in blue color corresponds to the case where no trellis decoding is involved on recovering the bits. Instead, the traditional decision rule method was performed on the received signal. The effect of ISI is quite evident in this case as it has resulted in a very large BER and poor SNR. The plot in cyan represents the theoretical bit error rate for the Nyquist BPSK. The plot in red corresponds to the BER of Viterbi decoded

FTN. The results are much better than the decision rule BER. It is also too close to the theoretical BER, but it must be considered that this system is operating at a rate $1/\tau = 1.42$ times faster than the Nyquist system.

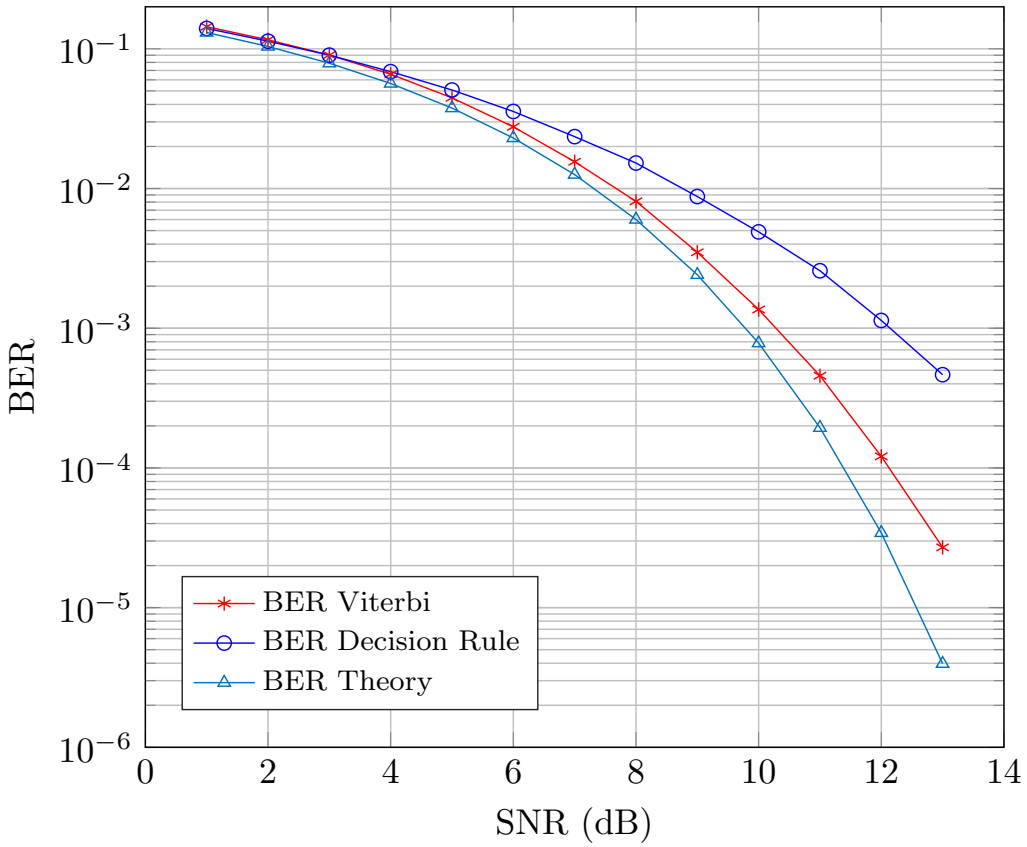


Figure 7.2: Bit error rate of an BPSK FTN signaling scheme with $D = 4$ and $\tau = 0.8$ and $\beta = 0.75$.

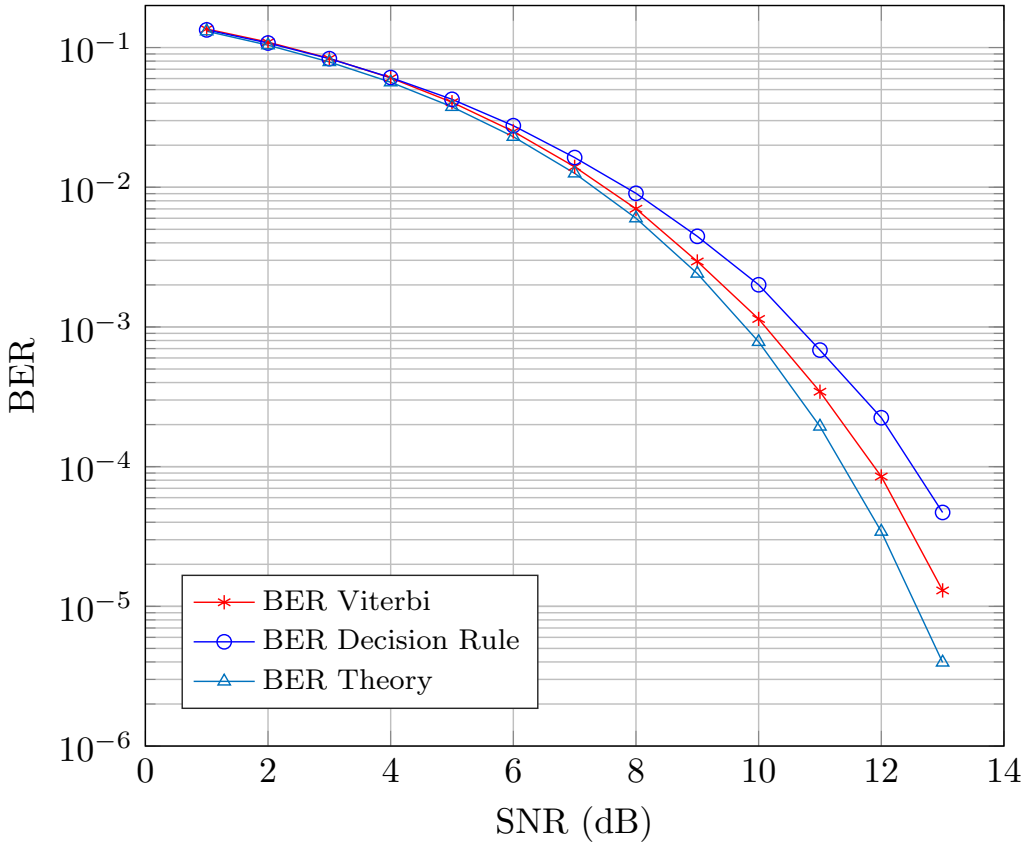


Figure 7.3: Bit error rate of an BPSK FTN signaling scheme with $D = 6$ and $\tau = 0.9$ and $\beta = 0.5$.

The Figure:7.2 represents the bit error rate of an FTN system with D = 4, $\tau = 0.8$ with BPSK constellation using a raised cosine pulse with roll-off factor $\beta = 0.75$. The colors of the plots remain the same. The decision rule has shown some improvement compared to the previous scenario, but still, the BER of the Viterbi decoding is consistent and still better than the decision rule. Figure:7.3 represents the bit error rate of an FTN system with D = 6, $\tau = 0.9$ with BPSK constellation using a raised cosine pulse with roll-off factor $\beta = 0.5$. The BER of the decision rule has shown some

more improvement, but the Viterbi decoder BER is still holding its position, However, we need to notice that even though the τ is increased to 0.9, the D = 6 which means 6 neighbouring pulses are interfering with the center pulse which increases the processing complexity of the decoder very high. In addition to that the roll-off factor β is reduced to 0.5 which means the side lobes will have higher amplitudes which make things even worse, but still, the Viterbi decoder has consistently performed. Plots with the other combinations of D, β, τ, are presented in the Appendix.

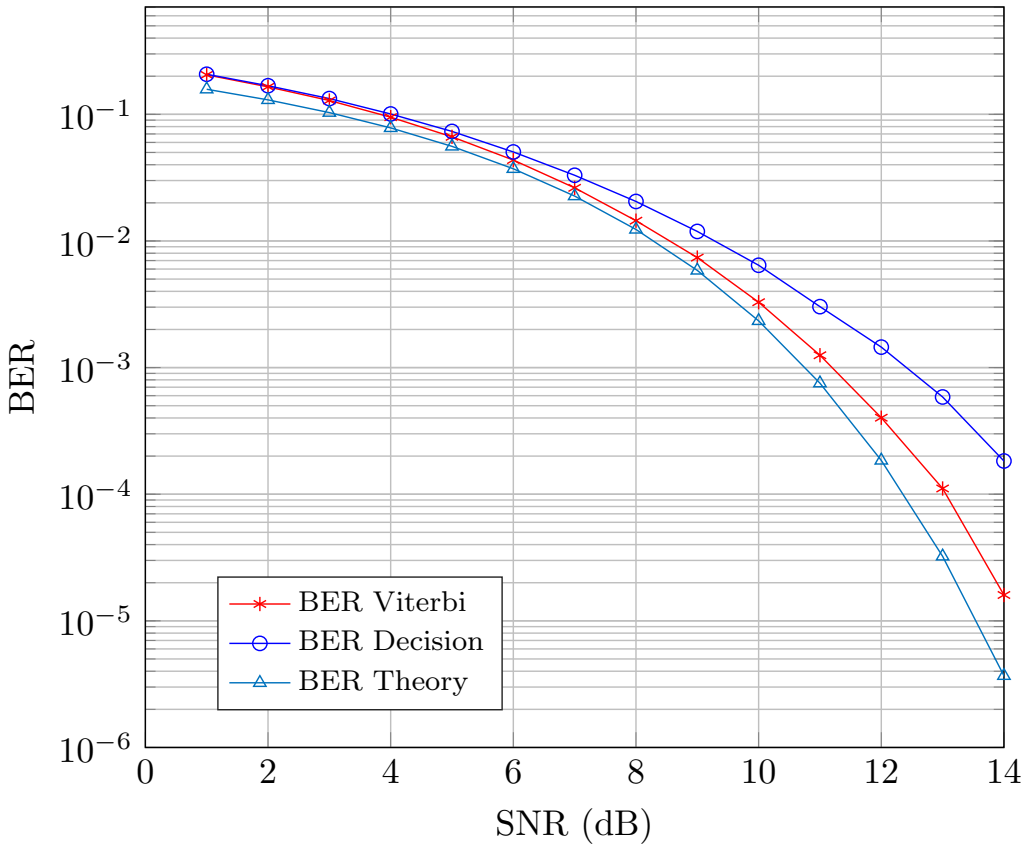


Figure 7.4: Bit error rate of an PAM-4 FTN signaling scheme with $D = 2$ and $\tau = 0.9$ and $\beta = 0.75$.

Now lets focus on the simulation results of a different constellation. The last 3 figures represent the plots for PAM-4 constellation with various D,τ and β values. The figure 7.4 represents the bit error rate of an FTN system with D = 2, $\tau = 0.8$ using a raised cosine pulse with roll-off factor $\beta = 0.5$. The performance of the Viterbi decoded output is considerably degraded to that of the theoretical BER and nonetheless compared to the BPSK systems which is obvious because of its complexity. However it carries twice the number of bits carried by the BPSK systems and it shows better performance than the decision rule decoded output, specifically at higher SNR values.

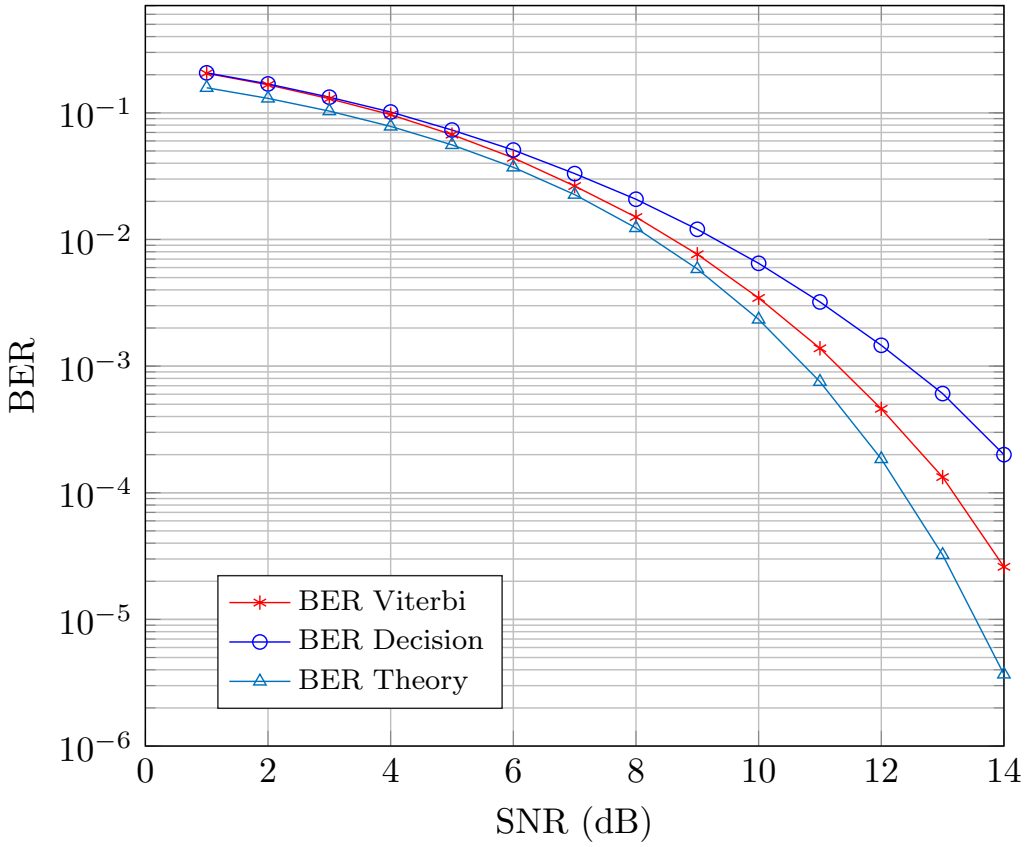


Figure 7.5: Bit error rate of an PAM-4 FTN signaling scheme with $D = 4$ and $\tau = 0.9$ and $\beta = 0.75$.

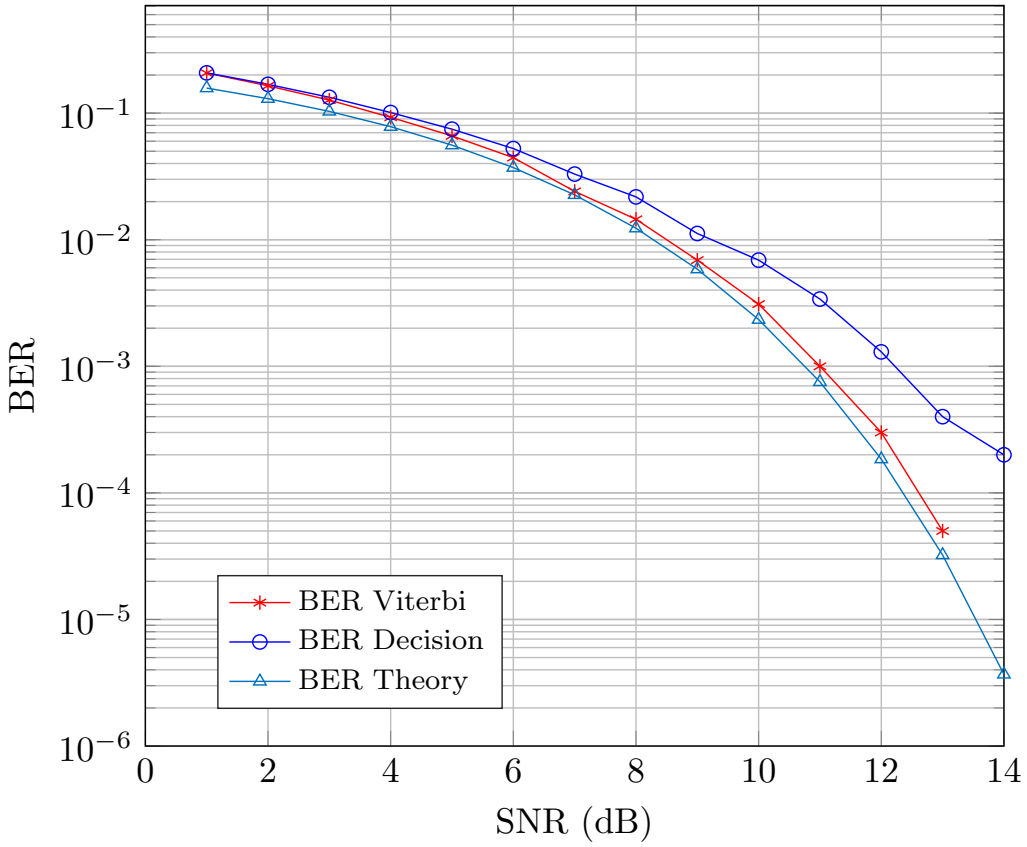


Figure 7.6: Bit error rate of an PAM-4 FTN signaling scheme with $D = 6$ and $\tau = 0.9$ and $\beta = 0.75$.

The figures 7.5 and 7.6 represent the bit error rate of an FTN systems with D = 4 and D = 6 with the same τ and β values. These systems also show similar kind of performance to that of the D = 2. The performance of this trellis decoding is also proportional to amount of computing resources and number if input bits. With the simulation results presented in the thesis, I hereby concur that trellis decoding for FTN system works it is worth exploring. I have also specified few ways to improve the performance of this system in the future work section of the next chapter.

Chapter 8

Conclusion

In this thesis, we have described an algorithm that automatically generates the trellis that describes the ISI in a faster than Nyquist communication system. The algorithm takes the pulse shape, the constellation and the time compression factor as inputs and provides us with a trellis with necessary states and branch metrics as output stored in a matrix that helps us in decoding the channel output from the FTN system. We constructed the entire communication system around the algorithm and demonstrated its efficiency by simulating various scenarios of test cases. We have shown examples of trellis generated by the algorithm for BPSK constellation with D = 2 and D = 4 and for 4-PAM constellation with D = 2. We have also presented simulation results, various plots distributed across various input parameters. All of them confirm that the FTN system along with our algorithm can operate at a bit error rate similar to that of Nyquist system while operating at a much higher data rate.

8.1 Difficulties Faced

I have encountered many difficulties over the course of this thesis and found ways to overcome them. The most important instance was to modify the traditional Viterbi algorithm to work for our need, as you know the Viterbi algorithm works well binary for both states and branch metrics, but in this thesis the Viterbi decoder has to work with integers for the states and decimal values for the branch metrics. It was not easy to handle the complexity that the decoder posed due to the increase in interference at higher D values. The increase in the trellis levels and the branch metric made it really difficult to debug in terms of synchronization and errors in the algorithm.

8.2 Future Work

The research in FTN signalling has caught the attention of many researchers and it will be much productive to pursue this further. This direction or research in FTN signalling has enormous potential for improvement. I have mentioned a few ideas for improvement to this thesis work below.

- Implementing the Viterbi decoding for other Modulation techniques like QPSK, QAM etc, and analyze the SNR performance.
- Reduce the complexity of the receiver by letting it pick the D value based on β, when the interference of the side lobes become negligible.
- Reduce the memory and processor utilization by implementing parallel computing and tracing the survivor path for a section of the trellis and clearing the memory.

- Hardware implementation of the algorithm to emulate real world applications.

Bibliography

[1] Nyquist.H. Certain factors affecting telegraph speed. *The Bell System Technical Journal*, 3(2):324–346, 1924.

[2] J. E. Mazo. Faster-Than-Nyquist Signaling. *Bell System Technical Journal*, 54(8), 1975. ISSN 0005-8580.

[3] D.W. Tufts. Nyquist's problem – The joint optimization of transmitter and receiver in pulse amplitude modulation. *Proceedings of the IEEE*, 53(3):248–259, March 1965. doi: 10.1109/PROC.1965.3682.

[4] H. J. Landau. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706, Oct 1967. ISSN 0018-9219. doi: 10.1109/PROC.1967.5962.

[5] D.W. Tufts. On "Sampling, data transmission, and the Nyquist rate"and adaptive communication using randomly time-varying channels. *Proceedings of the IEEE*, 56(5):889–889, May 1968. doi: 10.1109/PROC.1968.6437.

[6] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Infor-*

mation Theory, 13(2):260–269, April 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1054010.

[7] B. Saltzberg. Intersymbol interference error bounds with application to ideal bandlimited signaling. *IEEE Transactions on Information Theory*, 14(4):563–568, July 1968. ISSN 0018-9448. doi: 10.1109/TIT.1968.1054187.

[8] R.W.Lucky. Decision feedback and faster-than-nyquist transmission. *IEEE International symposium on Information Theory*, June 1970.

[9] G. Forney. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Transactions on Information Theory*, 18(3):363–378, May 1972. ISSN 0018-9448. doi: 10.1109/TIT.1972.1054829.

[10] Gd Forney. Viterbi Algorithm. *Proc. IEEE*, 61(3):268–278, 1973. doi: 10.1109/PROC.1973.9030.

[11] G. J. Foschini. Contrasting performance of faster binary signaling with qam. *AT T Bell Laboratories Technical Journal*, 63(8):1419–1445, Oct 1984. ISSN 0748-612X. doi: 10.1002/j.1538-7305.1984.tb00044.x.

[12] J. E. Mazo and H. J. Landau. On the minimum distance problem for faster-than-nyquist signaling. *IEEE Transactions on Information Theory*, 34(6):1420–1427, Nov 1988. ISSN 0018-9448. doi: 10.1109/18.21281.

[13] D. Hajela. On computing the minimum distance for faster than nyquist

signaling. *IEEE Transactions on Information Theory*, 36(2):289–295, March 1990. ISSN 0018-9448. doi: 10.1109/18.52475.

[14] Hajela.D. On faster than nyquist signaling: Further estimations on the minimum distance. *SIAM Journal on Applied Mathematics*, 52(3):900–8, 06 1992. Copyright - Copyright] © 1992 © Society for Industrial and Applied Mathematics; Last updated - 2012-01-20.

[15] A. D. Liveris and C. N. Georghiades. Exploiting faster-than-Nyquist signaling. *IEEE Trans. Commun.*, 51(9):1502–1511, September 2003. doi: 10.1109/TCOMM.2003.816943.

[16] Cheng-Kun Wang and Lin-Shan Lee. Practically realizable digital transmission significantly below the nyquist bandwidth. *IEEE Transactions on Communications*, 43(2/3/4):166–169, Feb 1995. ISSN 0090-6778. doi: 10.1109/26.380028.

[17] A. Hafeez and W. E. Stark. Decision feedback sequence estimation for unwhitened isi channels with applications to multiuser detection. *IEEE Journal on Selected Areas in Communications*, 16(9):1785–1795, Dec 1998. ISSN 0733-8716. doi: 10.1109/49.737647.

[18] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proceedings of ICC '93 - IEEE International Conference on Communications*, volume 2, pages 1064–1070 vol.2, May 1993. doi: 10.1109/ICC.1993.397441.

[19] Catherine Douillard, Michel Jézéquel, Claude Berrou, Département Electronique, Annie Picart, Pierre Didier, and Alain Glavieux. Iterative

correction of intersymbol interference: Turbo-equalization. *European Transactions on Telecommunications*, 6(5):507–511. doi: 10.1002/ett.4460060506. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4460060506.

[20] J. B. Anderson, F. Rusek, and V. Öwall. Faster-Than-Nyquist Signaling. *Proceedings of the IEEE*, 101(8):1817–1830, August 2013. doi: 10.1109/JPROC.2012.2233451.

[21] D Dasalukunte, F Rusek, and V Owall. An iterative decoder for multi-carrier faster-than-nyquist signaling systems. *2010 IEEE International Conference on Communications*, pages 1–5, 2010. doi: 10.1109/ICC.2010.5502554.

[22] A Prlja, J.B Anderson, and F Rusek. Receivers for faster-than-Nyquist signaling with and without turbo equalization. *2008 IEEE International Symposium on Information Theory, Toronto, ON*, pages 464–468, 2008. doi: 10.1109/ISIT.2008.4595029.

[23] J. B. Anderson. *Faster-than-Nyquist Signaling for 5G Communication*, chapter 2, pages 25–46. Wiley-IEEE Press, 2016. ISBN 9781119116493.

[24] John B. Anderson. A Review of Faster than Nyquist Signaling with an Extension to Four-Level Modulation. In *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (istc)*, pages 6–10. Ieee, 2016.

[25] Jungpil Yu, Joosung Park, Fredrik Rusek, Boris Kudryashov, and Irina Bocharova. High Order Modulation in Faster-than-Nyquist Signaling

Communication Systems. *2014 Ieee 80th Vehicular Technology Conference (vtc Fall)*, 2014.

[26] Y. G. Yoo and J. H. Cho. Asymptotic Optimality of Binary Faster-than-Nyquist Signaling. *IEEE Communications Letters*, 14(9):788–790, September 2010. ISSN 1089-7798. doi: 10.1109/LCOMM.2010.072910.100499.

[27] Ji Zhou, Yaojun Qiao, Zhanyu Yang, Qixiang Cheng, Qi Wang, Mengqi Guo, and Xizi Tang. Capacity limit for faster-than-Nyquist non-orthogonal frequency-division multiplexing signaling. *Sci Rep*, 7:3380, June 2017. doi: 10.1038/s41598-017-03571-6.

[28] M. El Hefnawy and H. Taoka. Overview of Faster-Than-Nyquist for Future Mobile Communication Systems. In *77th IEEE Vehicular Tech. Conf.* 2013.

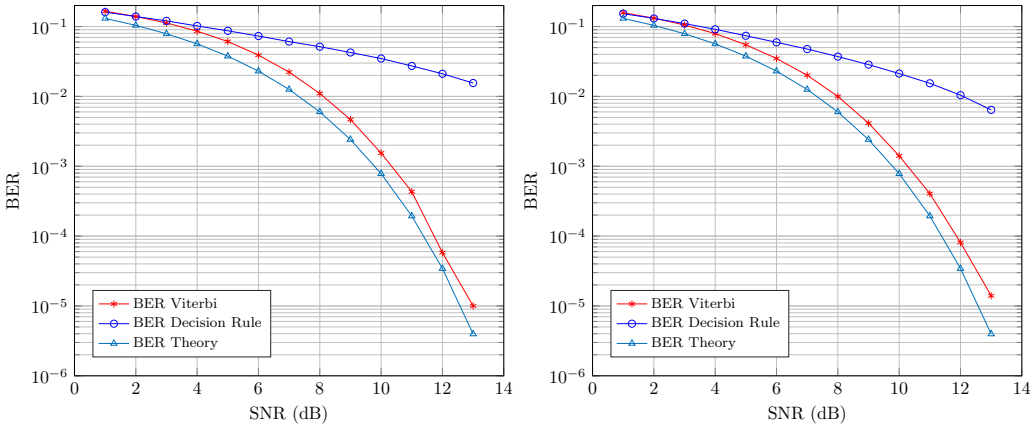
[29] M.R. Aaron and D.W. Tufts. Intersymbol interference and error probability. *IEEE Transactions on Information Theory*, 12(1):26–34, January 1966. doi: 10.1109/TIT.1966.1053842.

[30] P. Kabal and S. Pasupathy. Partial-response signaling. *IEEE Transactions on Communications*, 23(9):921–934, September 1975. doi: 10.1109/TCOM.1975.1092918.

[31] G. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6:311–329, 1960. doi: 10.1109/TIT.1960.1057571.

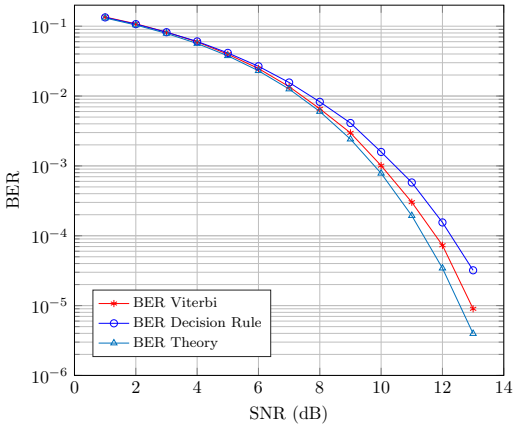
Appendix 1

Rest of the plots simulated for BPSK.

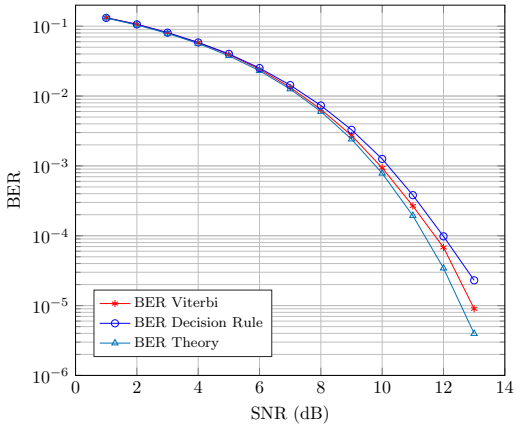


(a) Bit error rate of an BPSK FTN signaling scheme with $D = 2$ and $\tau = 0.7$ and $\beta = 0.5$.

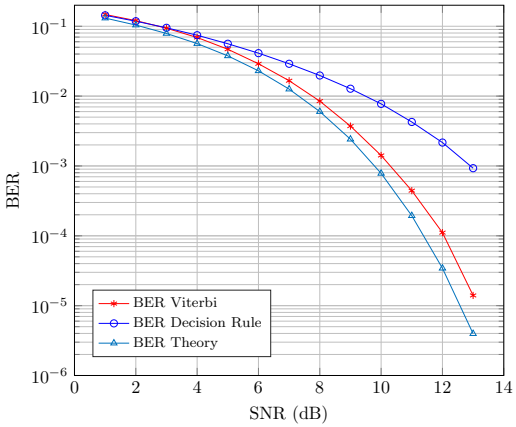
(b) Bit error rate of an BPSK FTN signaling scheme with $D = 2$ and $\tau = 0.7$ and $\beta = 0.75$.



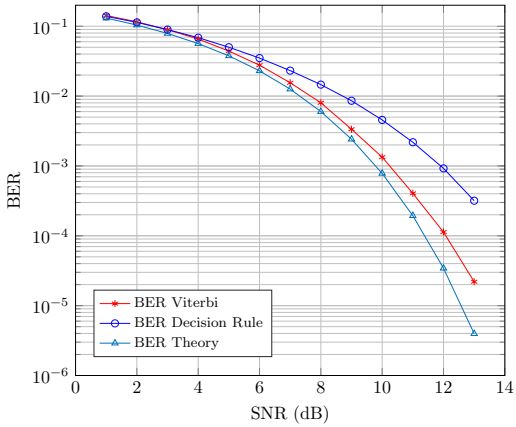
(a) Bit error rate of an BPSK FTN signaling scheme with $D = 2$ and $\tau = 0.9$ and $\beta = 0.5$.



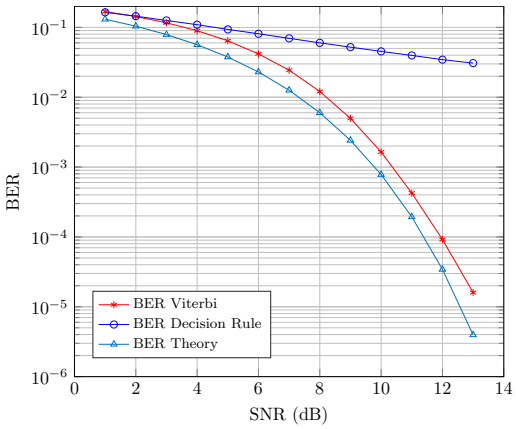
(b) Bit error rate of an BPSK FTN signaling scheme with $D = 2$ and $\tau = 0.9$ and $\beta = 0.75$.



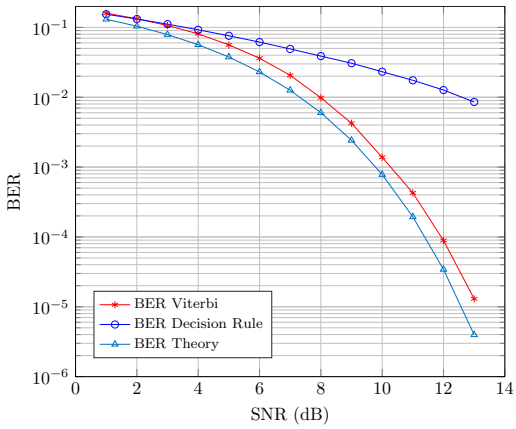
(a) Bit error rate of an BPSK FTN signaling scheme with $D = 2$ and $\tau = 0.8$ and $\beta = 0.5$.



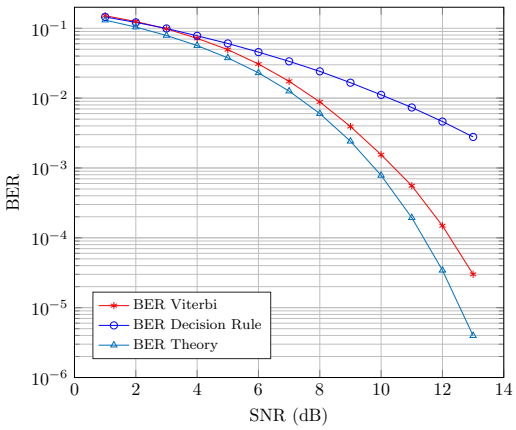
(b) Bit error rate of an BPSK FTN signaling scheme with $D = 2$ and $\tau = 0.8$ and $\beta = 0.75$.



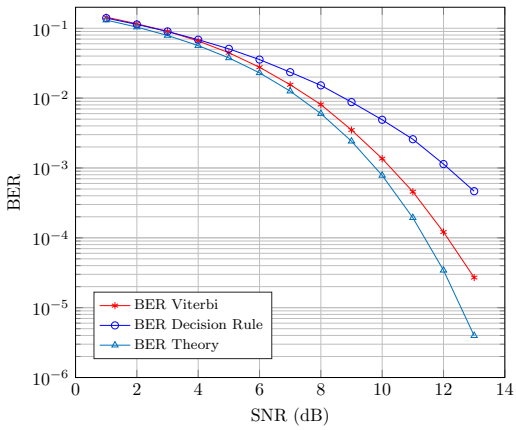
(a) Bit error rate of an BPSK FTN signaling scheme with $D = 4$ and $\tau = 0.7$ and $\beta = 0.5$.



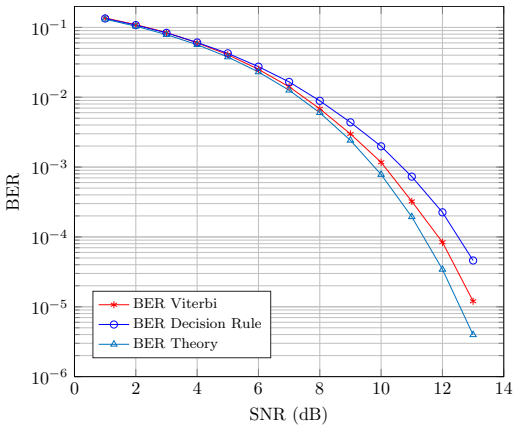
(b) Bit error rate of an BPSK FTN signaling scheme with $D = 4$ and $\tau = 0.7$ and $\beta = 0.75$.



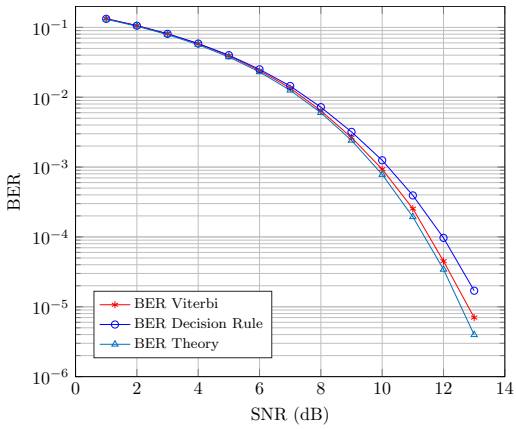
(a) Bit error rate of an BPSK FTN signaling scheme with $D = 4$ and $\tau = 0.8$ and $\beta = 0.5$.



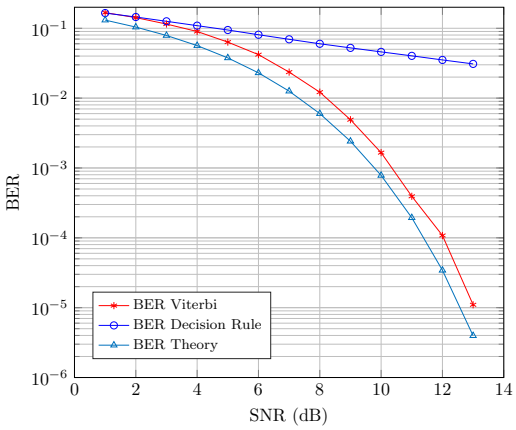
(b) Bit error rate of an BPSK FTN signaling scheme with $D = 4$ and $\tau = 0.8$ and $\beta = 0.75$.



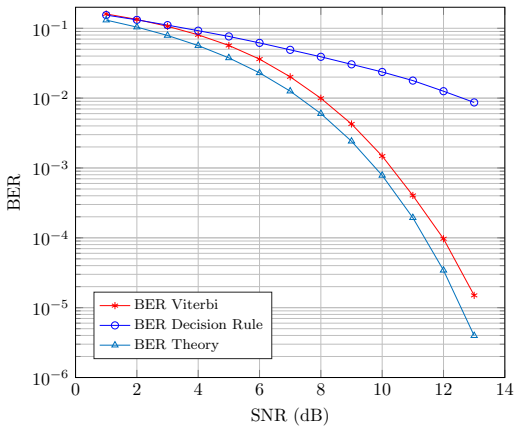
(a) Bit error rate of an BPSK FTN signaling scheme with $D = 4$ and $\tau = 0.9$ and $\beta = 0.5$.



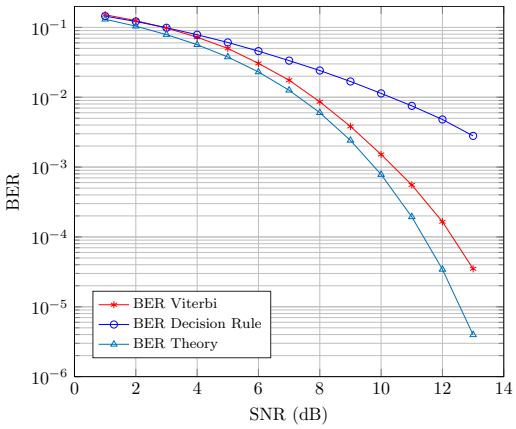
(b) Bit error rate of an BPSK FTN signaling scheme with $D = 4$ and $\tau = 0.9$ and $\beta = 0.75$.



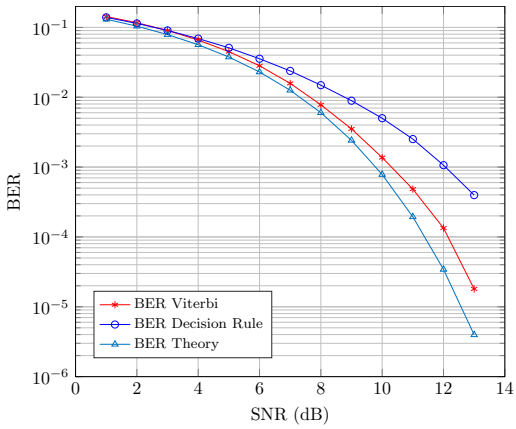
(a) Bit error rate of an BPSK FTN signaling scheme with $D = 6$ and $\tau = 0.7$ and $\beta = 0.5$.



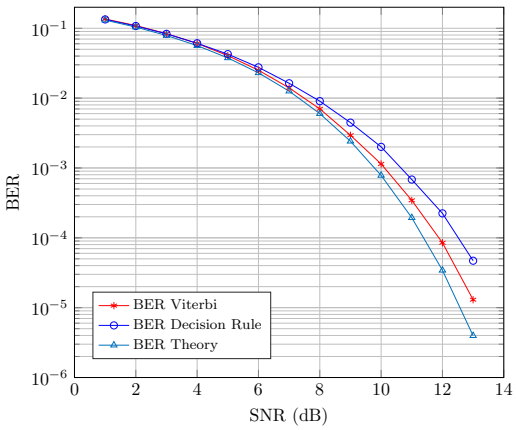
(b) Bit error rate of an BPSK FTN signaling scheme with $D = 6$ and $\tau = 0.7$ and $\beta = 0.75$.



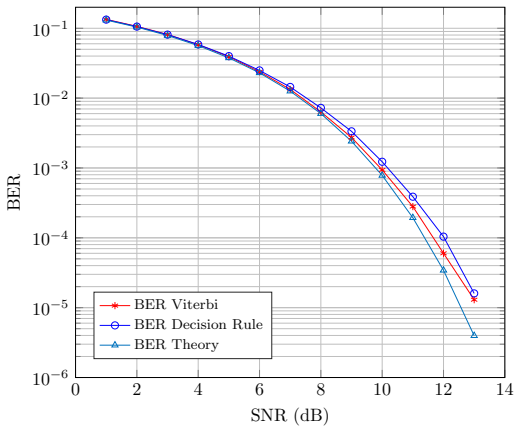
(a) Bit error rate of an BPSK FTN signaling scheme with $D = 6$ and $\tau = 0.8$ and $\beta = 0.5$.



(b) Bit error rate of an BPSK FTN signaling scheme with $D = 6$ and $\tau = 0.8$ and $\beta = 0.75$.



(a) Bit error rate of an BPSK FTN signaling scheme with $D = 6$ and $\tau = 0.9$ and $\beta = 0.5$.



(b) Bit error rate of an BPSK FTN signaling scheme with $D = 6$ and $\tau = 0.9$ and $\beta = 0.75$.

Appendix 2

Matlab Code to derive the states and branch metrics for the trellis

```
function [branch_value,branchmatrix]=branch_metric (fs,
    Ts,Tp,Tt,D,beta,start_samp,const,M)
%%
L = (D+1);  % Generate bit based on interference
number_of_branchmetric = M^L;
[p] = rcpulse(beta,D,Tp,Ts,'rc'); % Generates necessary
     pulse (sinc or rc)
p = p./max(p);
%%
branchmatrix(1,1) = cellstr('Currentstate');
branchmatrix(1,2) = cellstr('Input');
branchmatrix(1,3) = cellstr('Nextstate');
branchmatrix(1,4) = cellstr('Branch value');
%%
if M~=2
     X = pam_counter(L,M);
end
```

```
16 for k=0:1:(number_of_branchmetric)-1      % No of
     Amplitude possibilities Based on Interference
17 %k=0;
18 if (M == 2)
19     b=deci2bin(k,L);        % Binary Values for every K
         Value
20     a_k = b*2-1;            % Converting Binary into
         Bipolar
21 else
22     for pam = 1:1:L
23      a_k(1,pam) = const(X(k+1,pam)) ;
24      b=a_k;
25     end
26 end
27 impulses = zeros(1,length(a_k)*fs*Tp*Tt);
28 impulses(1:fs*Tp*Tt:end) = a_k;
29 %% Generate the transmitted signal
30 % Signal s is the analog signal that can be transmitted
     .
31 s = conv(impulses,p);
32 %plot (t1,s,'c');
33 %% Training Sequence
34 Outputs = s(start_samp:fs*Tp*Tt:end); % bits sampled at
     ntT time
35 Outputs = Outputs(1:length(a_k));
```

```
36
37 branch_value(k+1)=Outputs((D+2)/2);      % Assigning the
       Center Sampled value to the Branch metric
38 branchmatrix(k+2,1)=cellstr( num2str(b(1,1:D)));
39 branchmatrix(k+2,2)= cellstr(num2str(b(D+1)));
40 branchmatrix(k+2,3)= cellstr(num2str(b(1,2:D+1)));
41 branchmatrix(k+2,4)= cellstr(num2str(branch_value(k+1))
       );
42 end
```