Machine Learning Approaches for Faster-than-Nyquist (FTN) Signaling Detection

A thesis submitted to the
College of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical and Computer Engineering
University of Saskatchewan
Saskatoon

By

Sina Abbasi

©**Sina Abbasi**, Nov. 2022. All rights reserved.
Unless otherwise noted, copyright of the material in this thesis belongs to the author.

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering
University of Saskatchewan
57 Campus Drive
Saskatoon, Saskatchewan S7N 5A9
Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Abstract

There will be a significant demand on having a fast and reliable wireless communication systems in future. Since bandwidth and bit rate are tightly connected to each other, one approach will be increasing the bandwidth. However, the number of wireless devices are growing exponentially, and we don't have infinite bandwidth to allocate. On the other hand, increasing the bit rate for a given bandwidth, i.e., improving the spectral efficiency (SE), is another promising approach to have a fast and reliable wireless communication systems. Faster-than-Nyquist (FTN) is one of the candidates to improve the SE while this improvement comes at the expense of complexity of removing the introduced inter-symbol interference (ISI). In this thesis, we propose two algorithms to decrease the computational complexity regarding removing the ISI in FTN signaling.

In the first main contribution of the thesis, we introduce an equivalent FTN signaling model based on orthonormal basis pulses to transform the non-orthogonal FTN signaling transmission to an orthogonal transmission carrying real-number constellations. Then we propose a deep learning (DL) based algorithm to decrease the computational complexity of the known list sphere decoding (LSD) algorithm. In essence, the LSD is one of the algorithm that can be used for the detection process of the FTN signaling; however, at huge computational complexity. Simulation results show the proposed DL-based LSD reduces computational complexity by orders of magnitude while maintaining close-to-optimal performance.

In the second main contribution of the thesis, we view the FTN signaling detection problem as a classification problem, where the received FTN signaling signal viewed as an unlabeled class sample that is an element of a set of all potential classes samples. Assuming receiving N samples, conventional detectors search over an N-dimensional space which is computationally expensive especially for large value of N. However, we propose a low-complexity classifier (LCC) that performs the classification in N_p dimensional space where $N_p \ll N$. The proposed LCC's ability to balance performance and complexity is demonstrated by simulation results.

Acknowledgements

I would like to extend my sincerest appreciation to my supervisor, Dr. Ebrahim Bedeer, for providing me with the invaluable opportunity to work as a Master's student in his laboratory. Not only did I gain a wealth of knowledge and skills related to research, but I also had the privilege of observing and learning from his exemplary professional attitude in various aspects. I am deeply grateful for his guidance and support throughout the entire process of completing my thesis, which would not have been possible without his invaluable contributions. I am truly blessed to have had such wonderful supervisors throughout my life, and I firmly believe that their guidance and support has played a crucial role in any success I have achieved thus far.

I would also like to express my heartfelt gratitude to my loving parents, whose constant love and support have been an immense source of motivation for me, and my dear sister, who has always been a supportive and encouraging presence in my life.

To my Mother.

Contents

List of Figures

List of Abbreviations

APP	A posteriori probability
AWGN	Additive white Gaussian noise
BER	Bit error rate
BPSK	Binary phase-shift keying
DL	Deep learning
DL-LSD	Deep learning based List Sphere Decoding
E_b/N_o	Energy-per-bit to noise power spectral density ratio
FTN	Faster-than-Nyquist
ISI	Intersymbol interference
LCC	Low-complexity classifier
LLR	Log likelihood ratio
LSD	List sphere decoding
ML	Machine learning
MLSE	Maximum likelihood sequence estimation
NN	Neural network
PHY	Physical layer
RNN	Recurrent neural network
rRC	Root raised cosine
SD	Sphere decoding
SE	Spectral Efficiency

1 Introduction

1.1 Motivation

During past decades a lot of technological breakthroughs happened such as Internet and cellular phones. Led by these breakthroughs, the need of a stable wireless communication emerged. Today's wireless communication is more challenging than ever due to two factors: first, the exponential growth of wireless devices like smartphones, tablets, computers, drones, and smart gadgets. Second, the use of heavy real-time data applications on each of these wireless devices, such as streaming. The reason is each of these heavy data application wireless devices needs a specific bandwidth to operate on while in reality we do not have infinite spectral to accommodate all these fast growing devices. Even though the current rollout of 5G wireless communication systems handles 20 Gbps peak rates for enhanced multimedia applications; it is anticipated that these applications will eventually grow to include augmented reality, 3DTV/holographic communications, multi-sense communications, and/or combinations of these. The peak rates needed by the emerging applications are predicted to be in the range of a few terabits per second (Tbps), which is more than what 5G systems are capable of. Supporting this demand needs larger bandwidth than what is currently offered in 5G; these new frequency bands, however, come with very high absorption [1] which is only useful for short-range (few meters) of communications. Therefore, improving the spectral efficiency (SE) in limited bandwidth is a promising approach to mitigate this challenge.

Simply said, the SE is defined as the bit rate for a given bandwidth and is measured by bit/s/Hz. Therefore, improving the SE means increasing the bit rate for a given bandwidth. Faster-than-Nyquist (FTN) signaling as a non-orthogonal transmission method is among the promising candidates to improve SE compared to Nyquist signaling. Nyquist demonstrated that there is no interference between the samples of the receiver's matched filter output when signaling at rates greater than $1/T$ of T-orthogonal pulses, i.e., pulses that are orthogonal to an nT shift of themselves for nonzero integer n; this type of signaling is called Nyquist signaling. On the other hand, FTN signaling sends pulses faster than Nyquist criteria, causing intersymbol interference (ISI), i.e., each symbol interferes with the others. For example, in

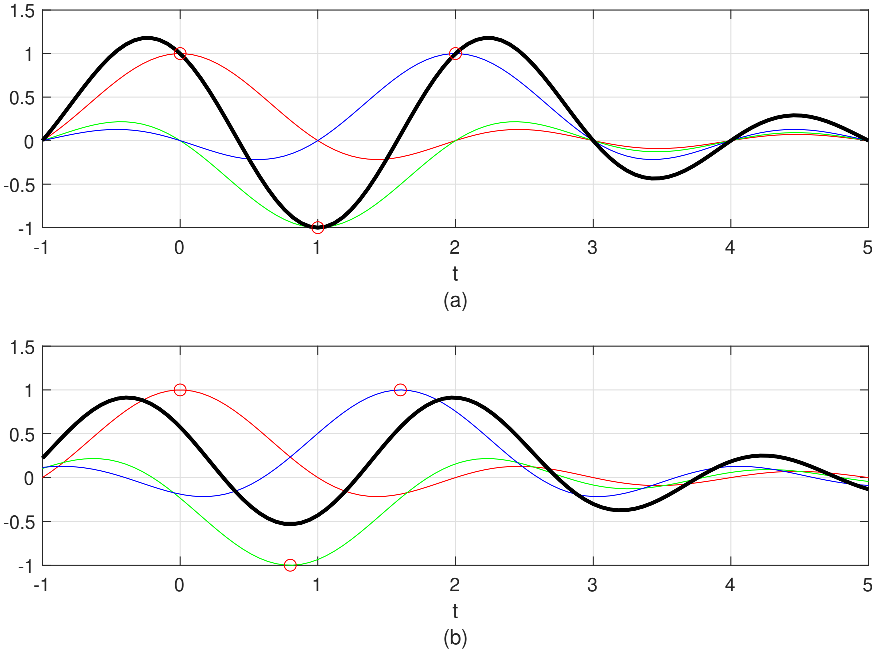

Figure 1.1: (a) Nyquist signaling $T = 1$. (b) FTN signaling $\tau T = 0.8$.

Fig. 1.1, the red, blue, and green colors represent symbols that are being sent, i.e., +1, -1, and +1, respectively. The black line is the summation of all these three symbols which is the actual transmit signal. As can be seen in Nyquist signaling Fig. 1.1.a, the transmit signal line, i.e., black line, has exactly the same value of actual symbols value, i.e. red dots, at each nT; however, in FTN signaling Fig. 1.1.b, the black line has a distance with red dots at each $n\tau T$. This distance is because of the ISI due to FTN signaling. Even though FTN signaling improves the SE, but this improvement comes at the cost of dealing with introduced intersymbol interference (ISI). Since removing the introduced ISI has a exponential computational complexity, in this thesis, we propose two algorithms to decrease the computational complexity due to FTN signaling.

Recently, deep learning (DL), one of the well-known machine learning (ML) techniques, has been shown to significantly enhance performance in several fields, including computer vision and natural language processing [2], [3]. Essentially ML algorithms comes useful when we are facing with problems without a clear mathematical definition, i.e., model deficit [4]–[6], or when the problem's solution has huge computational complexity which is not practical in a real-time application, i.e., algorithm deficit [7]–[9].

In the context of FTN signaling as mentioned earlier, we are dealing with significant computational complexity, either exponential of the transmit block length or exponential of ISI length, when we want to remove the introduced ISI and to detect the transmitted information bits at the receiver by using conventional detection and signal processing algorithms, e.g., list sphere decoding (LSD). Please note that the LSD is explained in Section 2.2 of this thesis. Motivated by the aforementioned application of ML, our research objectives in this thesis are focused on reducing the computational complexity of FTN detectors with the help of ML algorithms.

1.2 Research Objectives

There are two main objective in this research:

I. **Reducing the computational complexity of the LSD algorithm.** This part of thesis introduces a DL based solution to reduce the computational complexity of the LSD algorithm. We select DL among ML techniques because we can produce enormous training data sets, which is a great advantage in DL scenario. Additionally, the features we use as input into the model are quite vast, and many other ML models fail as the number of features increases due to the curse of dimensionality. Essentially, the LSD can be used for the FTN signaling detection process; however, at the cost of high computational complexity. We first present an alternative transmission model for FTN signaling using orthonormal basis functions to avoid the necessity of having a noise whitening filter at the receiver. Second, we provide a DL-based list sphere decoding (DL-LSD) technique that chooses and modifies the original LSD's initial radius in order to ensure that a pre-defined N_{L} lattice points are inside the hypersphere. To do this, a neural network (NN) is trained to provide an approximate initial radius with N_{L} lattice points. During the testing phase, If there are more than N_{L} lattice points inside the hypersphere, we keep the N_{L} points that are closest to the received FTN signaling vector; however, if the hypersphere has less than N_{L} points, we increase the approximate initial radius by a value that depends on the standard deviation of the distribution of the output radii from the training phase. The approximate log-likelihood ratio (LLR) value is then determined using the N_{L} points for the channel decoder.

II. **Low-complexity classification task for FTN signaling detection.** In the second main part of thesis, we look into the use of ML in reducing the detection complexity of FTN signaling. The FTN signaling detection problem is viewed as a classification problem, where the received signal is treated as an unlabeled class sample that is a part of a set of samples from every possible class. The set of all potential class samples, assuming we employ an off-the-shelf classifier, belongs to an N-dimensional space, where N is the transmission block length. Given that the detection process has an exponential relationship with N, an off-the-shelf classifier's computational cost can be very high, especially for large values of N. We propose a low-complexity classifier (LCC) to carry out the classification task in a N_p-dimension space, where $N_p \ll N$, by taking advantage of the ISI structure of FTN signaling. The proposed LCC consists of two stages: 1) off-line pre-classification that constructs the labeled classes samples in the N_p-dimensional space and 2) online classification where the detection of the received samples occurs. The proposed LCC is extended to produce soft-outputs as well. Further reduction of complexity when in calculating the soft-outputs comes from using the DL-LSD algorithm (proposed in Chapter 3).

1.3 Organization of the Thesis

This thesis is organized in a manuscript-based style. The obtained results are included in the form of accepted or under review manuscripts.

The thesis has overall five chapter. The motivation and research objectives are given in the first chapter. In Chapter 2, a background for Nyquist signaling and FTN signaling is discussed, and also is ended with a literature review. Chapter 3 introduced the deep learning-based LSD for FTN signaling detection. In Chapter 4, a low complexity classification approach for FTN signaling detection is proposed. Finally, a summary of the contribution of thesis and potential future work are given in the Chapter 5.

1.4 References

[1] "Propagation modeling for wireless communications in the terahertz band," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 96–101, Jun. 2018 (cit. on p. 2).

[2] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Apr. 2020 (cit. on p. 3).

[3] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020 (cit. on p. 3).

[4] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," in *Proceedings of the IEEE Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct. 2017, pp. 76–80 (cit. on p. 3).

[5] Z. Zhao, M. C. Vuran, F. Guo, and S. D. Scott, "Deep-waveform: A learned OFDM receiver based on deep complex-valued convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2407–2420, Jun. 2021 (cit. on p. 3).

[6] M. Honkala, D. Korpi, and J. M. Huttunen, "Deeprx: Fully convolutional deep learning receiver," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3925–3940, Feb. 2021 (cit. on p. 3).

[7] Y. Wang, M. Martonosi, and L.-S. Peh, "A supervised learning approach for routing optimizations in wireless sensor networks," in *Proceedings of the IEEE*, May 2006, pp. 79–86 (cit. on p. 4).

[8] H. Agirman-Tosun, Y. Liu, A. M. Haimovich, *et al.*, “Modulation classification of MIMO-OFDM signals by independent component analysis and support vector machines,” in *Proceedings of the IEEE*, Nov. 2011, pp. 1903–1907 (cit. on p. 4).

[9] G. De Veciana and A. Zakhor, “Neural net-based continuous phase modulation receivers,” *IEEE Transactions of Communication*, vol. 40, no. 8, pp. 1396–1408, Aug. 1992 (cit. on p. 4).

2 Background and Literature Review

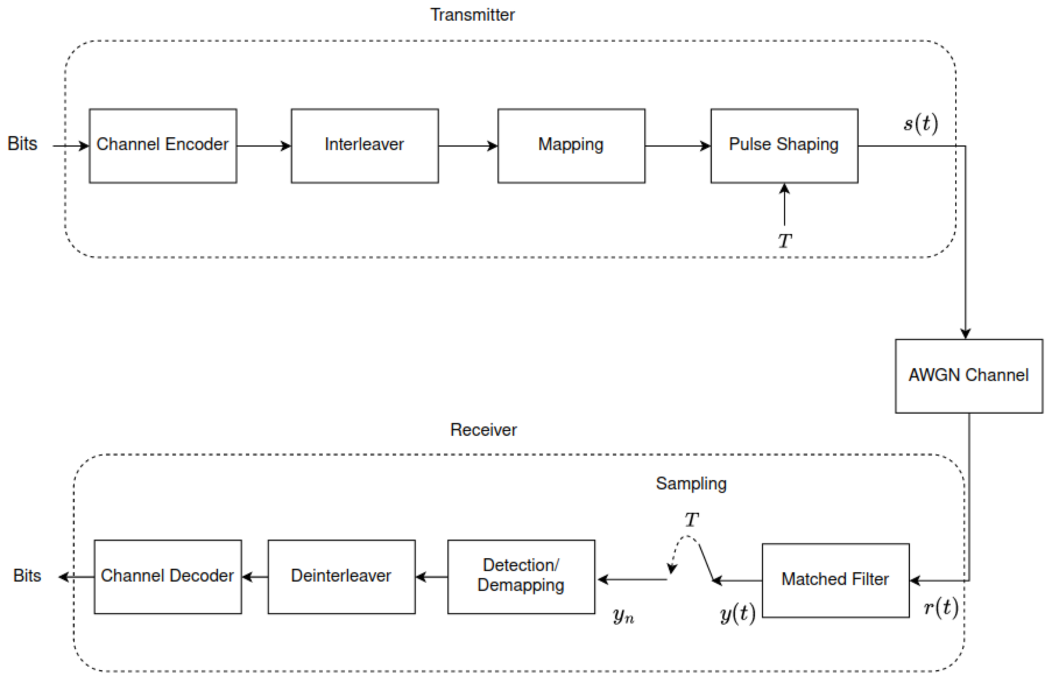

Figure 2.1: Basic communication system.

2.1 Communication System

An example of a basic communication system model is illustrated in Fig. 2.1 which mainly includes three essential parts: transmitter, channel, and receiver. In this study, we focused on the transmitter and receiver side of a communication system, and an additive white Gaussian noise (AWGN) channel is considered. First, we explain the concepts of noise and the AWGN channel, and then we discuss Nyquist signaling and the FTN signaling system model in detail.

2.1.1 Noise and AWGN channel

The physical medium used to carry a signal from a transmitter to a receiver is known as a communication channel. The atmosphere could serve as the channel in wireless communication (free space). The essential thing to notice here is that the transmitted signal is corrupted in a random manner by a variety of possible mechanisms, such as additive thermal noise generated by electronic devices; man-made noise, and/or fading noise. Additive noise is a typical issue in the transmission of signals which is mainly caused by thermal noise. Typically, electronic components at the receiver generate thermal noise by the thermal motions of electrons. This type of noise has a Gaussian distribution, and is usually called additive Gaussian noise [10].

Color is another characteristic of the noise that is being used. The color of the noise reflects the frequency content in the noise power spectral density. White noise is noise with a constant power spectral density that is frequency independent. Thermal noise sources are classified as white noises [11]. Therefore, an AWGN channel is mathematically modeled as a Gaussian distribution with zero mean and constant variance.

2.1.2 Nyquist Signaling

Transmitter

In the transmitter side, the information bits are going through the channel encoder and interleaver blocks. Basically, the goal of the channel encoder is adding redundancy to transmit bits with some sort of algorithms, e.g., convolutional coding, hamming coding, etc, where in this case the errors caused by noise during transmission can be mitigated at the receiver. Then, encoded bits pass into the mapping block where the bits are mapped to symbols. A lot of mapping schemes has being used based on communication system needs, and some popular ones are: pulse amplitude modulation (PAM), phase-shift keying (PSK), frequency-shift keying (FSK).

After mapping, for amplitude and phase modulation, the transmit symbols are modulated with pulse shape $h(t)$. The bandwidth of modulated signal is a function of the bandwidth of pulse shape $h(t)$; hence, $h(t)$ should be selected in such a way that ISI is not introduced; in other words, there is no interference between symbols at each T where $1/T$ is the transmit symbol rate. This criteria is called Nyquist criterion. There are a couple of candidate pulses that satisfy the Nyquist criterion, e.g., Raised Cosine pulses [12].

After the pulse shaping block, the transmit signals are ready to be sent with transmission rate $1/T$ based on Nyquist criterion. The following equation shows transmit signal $s(t)$:

$$s(t) = \sum_{n} a_n h(t - nT), \tag{2.1}$$

where $a_n, n = 1, ..., N$, are data symbols, and N is pre-defined transmission block size based on channel encoder structure.

Receiver

The transmit signal goes through the AWGN channel and perturbs by white noise. The received signal at the receiver is formulated as follow:

$$r(t) = s(t) + w(t), \tag{2.2}$$

where $w(t)$ is additive white Gaussian noise (AWGN) with zero mean and variance of σ^2. Then, $r(t)$ passes through a matched filter which is designed to maximize the signal-to-noise ratio (SNR). The received signal, $y(t)$, after the matched filter, $h(t)$, is formulated as follow:

$$y(t) = r(t) * h(t), \tag{2.3}$$

where $*$ defined the convolution operator. Then, $y(t)$ is sampled at the rate $1/T$ after the matched filter:

$$y_n = a_n + w_n, \tag{2.4}$$

where the detection process of transmit symbol a_n can be achieved on a symbol-by-symbol basis detection. By detection, we are referring to the process of identifying and recognizing the transmit symbols, i.e. a_n, at the receiver side. After the detection block, detected symbols go into the deinterleaver and channel decoder accordingly to get the information bits.

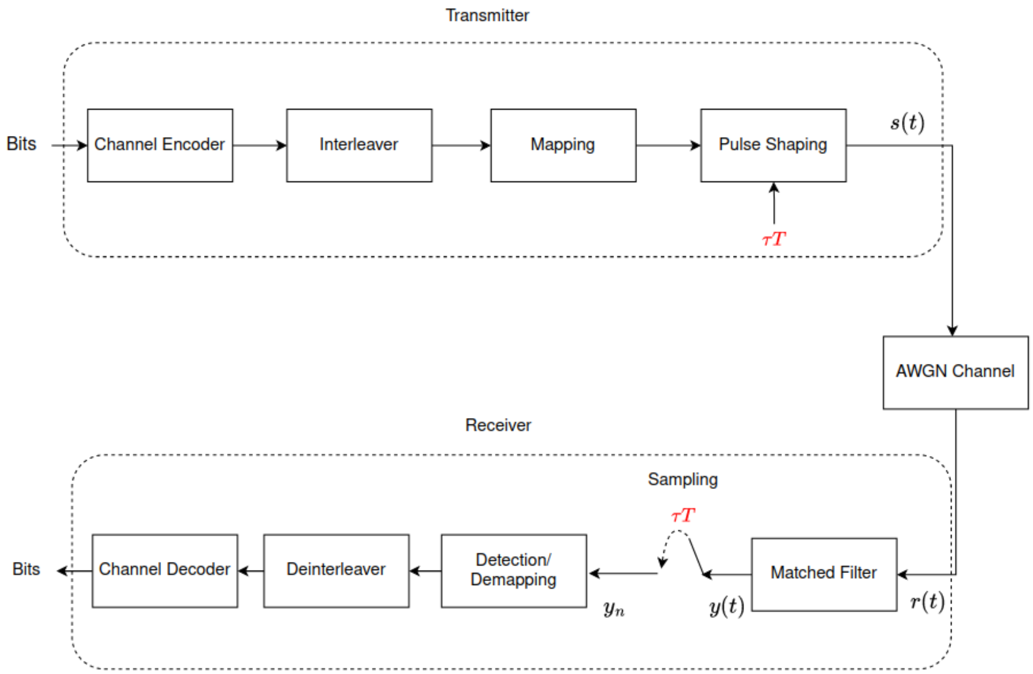

Figure 2.2: FTN signaling.

2.1.3 FTN Signaling

Transmitter

As shown in Fig. 2.2, since in FTN signaling we sent the transmit signal faster than Nyquist rate, i.e., $1/\tau T$. So the transmit signal is formulated as:

$$s(t) = \sum_n a_n h(t - n\tau T), \tag{2.5}$$

Receiver

After matched filter, the received signal is formulated as:

$$y(t) = \sum_n a_n g(t - n\tau T) + \bar{w}(t), \tag{2.6}$$

where $g(t) = \int h(x)h(x-t)dx$ and $\bar{w}(t) = \int w(x)h(x-t)dx$. Bedeer et al. has been shown that $\bar{w}(t)$ is a colored noise and we no longer have white noise [13]. After sampling at rate

$1/\tau T$ the n-th received symbol is:

$$\begin{aligned} y_n &= y(n\tau T), \\ &= \sum_{n'} a_{n'} g(n\tau T - n'\tau T) + \bar{w}(n\tau T), \\ &= a_n g(0) + \sum_{n', n' \neq n} a_{n'} g((n - n')\tau T) + \bar{w}_n, \end{aligned} \tag{2.7}$$

where the first part, i.e., $a_n g(0)$, is the desired symbol, and second part is due to the ISI, and last part is the sampled colored noise. We can write the received symbols as a vector as:

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{a} + \bar{\boldsymbol{w}}, \tag{2.8}$$

where $\boldsymbol{G}$ is the ISI matrix, where $G_{i,j} = g((i-j)\tau T)$ represents the ISI between data symbol i and j, and $\bar{\boldsymbol{w}}$ is the colored noise with zero-mean and covariance of $\sigma^2 \boldsymbol{G}$. If we define $\boldsymbol{z} = \boldsymbol{G}^{-1}\boldsymbol{y}$ and $\boldsymbol{\eta} = \boldsymbol{G}^{-1}\bar{\boldsymbol{w}}$, we can re-write (2.8) as:

$$\boldsymbol{z} = \boldsymbol{a} + \boldsymbol{\eta}, \tag{2.9}$$

where the maximum likelihood sequence estimator (MLSE) problem can be formulate as [14]:

$$\arg\min_{\boldsymbol{a}} (\boldsymbol{z} - \boldsymbol{a})^T \boldsymbol{G} (\boldsymbol{z} - \boldsymbol{a}). \tag{2.10}$$

Brute force search can solve this MLSE problem at the cost of huge computational complexity which prevents its real-world implementation.

2.2 List Sphere Decoding

In this section, we discuss one of the main detection algorithm in conventional communication systems named sphere decoding (SD). Originally, the SD algorithm was proposed by Fincke et al. [15], to solve the integer least-squares problem. An integer least-squares problem can be formulated as follow:

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a} \in \mathcal{D}} \|\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}\|_2^2, \tag{2.11}$$

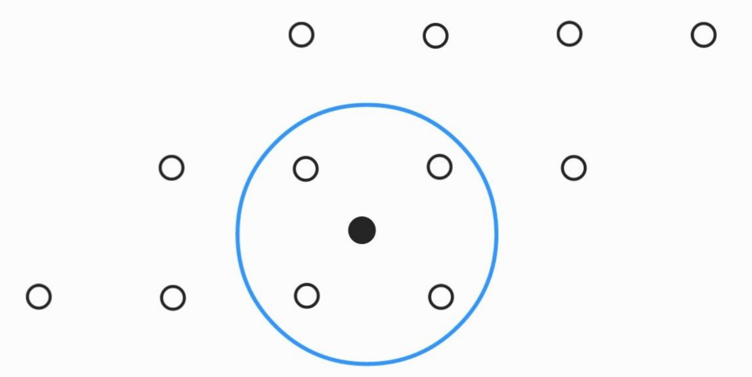

Figure 2.3: Idea of sphere decoding.

where $\mathcal{D}$ is a subset of the integer lattice $\mathbb{Z}^N$. In communication systems, this type of least-squares problem is used to detect the sequence of transmit symbols $\boldsymbol{a}$ when vector $\boldsymbol{y}$ is received at the receiver. In other words, the SD is trying to find the closest point in the skewed lattice $\boldsymbol{Ga}$ to the received vector $\boldsymbol{y}$. For example, in Fig. 2.3 the white points belong to the skewed lattice and the black point is the received vector where SD is trying to get the closest white point to the black point within its hypersphere, i.e., blue sphere. The main goal of SD is to search over all of the lattice points that are contained within a hypersphere with radius d that is centered on the received vector $\boldsymbol{y}$. Thus, in contrast to search over the entire lattice, i.e., brute force search, to obtain the MLSE solution, the search region of the SD is the set of points inside the hypersphere which results in reduced computational complexity. Obviously, the closest lattice point inside the hypersphere to vector $\boldsymbol{y}$ is also the closest lattice point of the entire lattice; then, the solution of the SD algorithm is the optimal solution of the MLSE.

In the SD algorithm, choosing the initial radius d is challenging; it should be chosen so that it is neither too huge nor too small. If it is too large, then the number of lattice points that fall inside the hypersphere will be in the order of entire lattice and the SD algorithm will have the same complexity as a brute force search solution. On the other hand, if the radius d is too small, it is possible that there is no lattice point inside the hypersphere. Hassibi et al. suggested choosing the radius as a scaled variance of the noise [16].

The SD finds all lattice points inside its sphere in the way that a lattice point is inside hypersphere centered at $\boldsymbol{y}$ with the radius of d, if and only if:

$$d^2 \geq \|\boldsymbol{y} - \boldsymbol{Ga}\|_2^2 \tag{2.12}$$

By $\boldsymbol{QR}$ factorization of matrix $\boldsymbol{G}$, i.e. $\boldsymbol{G} = \boldsymbol{QR}$, we break down the problem in lower dimension and rewrite (2.12) as:

$$\begin{aligned} d^2 &\geq \|\boldsymbol{z} - \boldsymbol{Ra}\|_2^2 \\ d^2 &\geq \sum_{i=1}^{N} \left(z_i - \sum_{j=i}^{N} R_{i,j} a_j \right)^2 , \end{aligned} \tag{2.13}$$

where $\boldsymbol{z} = \boldsymbol{Q^T y}$.This is the situation where the upper triangular property of R comes useful. In the inequality above, the right-hand side (RHS) can be expanded as follows:

$$d^2 \geq (z_N - R_{N,N} a_N)^2 + (z_{N-1} - R_{N-1,N-1} a_{N-1} - R_{N-1,N} a_N)^2 + ... \tag{2.14}$$

It can be seen that the first term only depends on a_N, the second term depends only on a_N, a_{N-1}, and so on so forth. Then, based on the first term, one of the necessary conditions is as:

$$d^2 \geq (z_N - R_{N,N} a_N)^2 , \tag{2.15}$$

which after expanding, its equivalent to following constraint for a_N:

$$\left\lceil \frac{z_N - d}{R_{N,N}} \right\rceil \leq a_N \leq \left\lfloor \frac{z_N + d}{R_{N,N}} \right\rfloor , \tag{2.16}$$

Of course, (2.16) is not enough to determine all symbols, we need to have such a constraint for every symbol. For every a_N that satisfies (2.16), we define $d_{N-1}^2 = d^2 - (z_N - R_{N,N} a_N)^2$ and then a stronger constraint with looking the first two terms of (2.14) can be driven for $(N-1)$-th symbol, i.e., a_{N-1}:

$$\left\lceil \frac{z_{N-1} - R_{N-1,N} a_N - d_{N-1}}{R_{N,N}} \right\rceil \leq a_{N-1} \leq \left\lfloor \frac{z_{N-1} - R_{N-1,N} a_N + d_{N-1}}{R_{N,N}} \right\rfloor . \tag{2.17}$$

Similar steps can be taken for a_{N-2} and so on until a_1 to acquire all lattice points belonging to (2.12).

After the SD algorithm finds all points inside the hypersphere, one can sort the points based on the closest distance to the vector $\boldsymbol{y}$ and make a list of these points, this type of SD is known as list sphere decoding (LSD). The list of these points is being used to produce a soft output that can be used by the channel decoder. This soft output, which is represented as a log-likelihood ratio (LLR) value, can be produced by maximizing a posteriori probability (APP) for a particular bit. The magnitude of the LLR describes the decision's reliability, whereas the sign of the LLR shows whether the provided bit is zero or one. The LLR for a bit x_k given the received vector $\boldsymbol{y}$ is written as:

$$L_D\left(x_k \mid \boldsymbol{y}\right)=\ln \frac{P\left(x_j=1 \mid \boldsymbol{y}\right)}{P\left(x_j=0 \mid \boldsymbol{y}\right)}, \tag{2.18}$$

where x_k is the kth bit of $N \times 1$ vector $\boldsymbol{x}$ of all bits in one transmit block. Let us assume that the list of closest points obtained from LSD is defined as $\boldsymbol{\mathcal{L}}$, then we use the Bayes theorem to re-write (2.18) and approximate it as:

$$\begin{aligned} &\tilde{L}_D\left(x_k \mid \boldsymbol{y}\right)= \\ &L_A\left(x_k\right)+\ln \frac{\sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{L}}_{k,1}} p(\boldsymbol{y} \mid \boldsymbol{x}) \cdot \exp \sum_{j \in \mathcal{J}_{k,\boldsymbol{x}}} L_A\left(x_j\right)}{\sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{L}}_{k,0}} p(\boldsymbol{y} \mid \boldsymbol{x}) \cdot \exp \sum_{j \in \mathcal{J}_{k,\boldsymbol{x}}} L_A\left(x_j\right)}, \end{aligned} \tag{2.19}$$

where $\boldsymbol{\mathcal{L}}_{k,1}=\{\boldsymbol{x} \in \boldsymbol{\mathcal{L}} \mid x_k=1\}$, $\boldsymbol{\mathcal{L}}_{k,0}=\{\boldsymbol{x} \in \boldsymbol{\mathcal{L}} \mid x_k=0\}$, and

$$L_A(x_j)=\ln \frac{P(x_j=+1)}{P(x_j=0)}, \tag{2.20}$$

and the likelihood function $p(\boldsymbol{y} \mid \boldsymbol{x})$ is given as follow:

$$p(\boldsymbol{y} \mid \boldsymbol{x})=\frac{\exp\left(-\frac{1}{2\sigma^2} \cdot \|\boldsymbol{y}-\boldsymbol{G}\boldsymbol{a}\|^2\right)}{\left(2\pi\sigma^2\right)^N}. \tag{2.21}$$

2.3 Mazo Limit

In (2.5) we have defined the transmit signal in FTN signaling. By considering all possible a_n sequences we can generate all possible $s(t)$ values. Lets assume that $s_i(t)$ and $s_j(t)$ are two of these odds. Then, the least square Euclidean distance between any such pair is defined as [17]:

$$d_{min}^2=\frac{1}{2}\int_{-\infty}^{\infty}|s_i(t)-s_j(t)|^2 dt, \quad i \neq j. \tag{2.22}$$

Regardless of pulse shaping, d_{min}^2 is always equal to 2 in binary orthogonal transmission. In 1975, Mazo [18] worked with a binary transmission with sinc pulses, and through his investigation, he found out an interesting observation that by speeding up sinc pulses with the acceleration factor of τ, $\tau \in [0.802, 1]$, the d_{min}^2 remains equal 2 while we have non-orthogonal and faster transmission, i.e., $1/0.802 = 25\%$ more bits can be carried, without losing any bit error rate performance. This is referred to as the Mazo limit.

For many years, Mazo's achievement went unnoticed, but after 1990, the attention of the research community surged. The most frequently utilized in applications, root raised cosine (rRC) pulses, were demonstrated to have the same effect. For example, in binary transmission using rRC pulses distance remains 2 at $\tau = 0.703$, resulting in a 42% increase in bit density. Mazo limit is also applicable to non-binary transmission [19] and non-linear modulation [20] as well where the bandwidth at which the d_{min}^2 initially drops is always substantially smaller than the Nyquist bandwidth, resulting in significant bandwidth savings.

Researches around FTN signaling have been extended beyond the Mazo limit with significant acceleration of transmit symbols but at the cost of computational complexity to detect the transmit signal. Also in this work, we have considered going beyond the Mazo limit.

2.4 Related Works

Since the thesis is focused on the detection of transmit bits at the receiver in the FTN signaling system model, we mainly discuss these algorithms in two following categories:

2.4.1 Non-ML FTN Signaling Detectors

Simple detectors perform poorly in terms of bit error rate and fail to detect FTN signals in severe ISI, i.e., the value of τ is small. Nonetheless, it has been demonstrated that the ISI that was introduced by the FTN signaling has a trellis structure [21]. Then detectors with trellis structure can be used as detectors for the received samples in FTN signaling. Among classic candidates of the trellis based detectors, we can name the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm that can be used to identify transmit sequences in FTN signaling. However, the number of states in the BCJR algorithm in FTN signaling is in exponential order of ISI length and then the computational complexity of using BCJR is becoming so expensive. A reduced-space version of the BCJR with the help of the M-algorithm, where it considers the M best state of BCJR, has been investigated by Anderson et al. [22]. More recently, an improved M-BCJR algorithm based on the Ungerboeck observation model was proposed by Li et al. [23], where trellis states are reserved at each trellis section by considering the effect of future symbols, and the detection method only exploring a fraction of the original ISI trellis. Both of these methods provide a flexible balance between performance and computational complexity, however, they can be computationally expensive especially when ISI length becomes large.

Bedeer et al. proposed a low complexity symbol-by-symbol based estimator for FTN signaling [24]. They proposed a successive symbol-by-symbol sequence estimator with go-back K sequence estimator (SSSgb$K$SE), where the estimator of the current symbol information re-estimates previous K symbols to reduce error propagation due symbol-by-symbol estimation. Their proposed algorithm has very low complexity and have a close-to-optimal performance when ISI length is very small, however, in mid or severe ISI it fails to provide satisfactory performance. Although the work introduce by Kulhandjian et al. [25], could outperform the SSSgb$K$SE with probabilistic data association (PDA) algorithms at the cost of a modest increase in computational complexity but the performance was not close to MLSE estimators, and the severe ISI was not investigated. Bedeer et al., in another study, investigate severe ISI scenario as well, where they proposed a generalized sphere decoding-based sequence estimation (SDSE) that uses a whitening noise filter after matching filter [13]. Even though the performance obtained by their approach is the same as the MLSE but it depends on having an exact whitening noise filter and is limited to binary modulation. However, they proposed a sub-optimal solution for high order of modulation with the help of semidefinite relaxation (SDR)-based sequence estimation for M-quadrature-amplitude modulation (QAM) [14] and M-ary phase shift keying (PSK) [26]. Their approach still requires an impractically large computational time for modulation orders larger than 16 QAM. Nonetheless, they overcame this issue in their recent work [27], where a detection algorithm was proposed based on alternating directions multiplier method (ADMM) for ultra high order of modulation, i.e., up to 64K-QAM.

Since in FTN signaling the ISI is known, with the help of the channel shortening idea, it is possible to derive the optimal relationship between the length of the shortened channel and the performance. Fan et al. aimed to solve a MLSE problem with the shortened ISI channel, and they came up with a trade-off between performance and computational complexity [28]. However, their work reduces the computational complexity when the ISI is not severe. The computational complexity of their approach is still exponentially related to the length of shortened ISI channel, and in mid range or severe ISI the shortened channel will have long length. Furthermore, a code-based channel shortening scheme was introduced by Li et al. [29], where a particular type of convolutional codes were used to absorb the channel memory. Strong concatenated codes can be easily designed based on these convolutional codes. The designed concatenated code, when combined with FTN signaling, improves the error performance. However, in severe ISI case, the memory length will increase which leads to higher computational complexity, otherwise, without increasing the length of channel memory we lose performance significantly.

A Gaussian message-passing algorithm based on a factor graph was proposed for an FTN receiver by Wen et al. [30], where in order to simplify detection, an auto-regressive method was used to approximate the FTN-specific correlated noise. Their approach can achieve reduced complexity while causing small performance degradation. A similar message-passing method based on a factor graph was developed for FTN signaling by Li et al. [31], which was integrated with space-time multi-mode index modulation and also with non-orthogonal multiple access (NOMA) [32], where it was demonstrated that the bandwidth efficiency is further improved compared to conventional NOMA system by exploiting FTN signaling based on numerical results. Besides, FTN signaling has been applied as a successful paradigm to improve the spectral efficiency for multi-carrier transmission as well, where Wang et al. [33] proposed a transceiver that is compatible with the existing orthogonal frequency division multiplexing (OFDM). We direct the reader to [17] for a description of the most important FTN signaling detection methods and [34] for a more recent survey.

2.4.2 ML FTN Signaling Detectors

The applications of ML have been extended to design FTN signaling systems in very few works. Song et al. proposed in their study an efficient DL-based architecture for FTN receivers that can replace either the signal detection block or both the signal detection and channel decoding blocks for uncoded and coded FTN signaling, respectively [35]. Their proposed DL-based FTN receivers showed near optimal performance for non-severe ISI operating regions. However, their work without considering the training processes still has a greater number of multiplication operations in comparison to the optimal solution. Even though there is no gain in terms of computational complexity, but they claimed that their solution supports parallel implementation and it can reduce the detection delay.

Liu et al. proposed a DL-based sum-product algorithm for FTN signaling that operates on a modified factor graph and concatenates a neural network function node to the variable nodes to approximate the optimal error rate performance [36]. The underlying neural network structure, however, is designed particularly for binary transmission, which significantly restricts its range of use.

It goes without saying that, most of the non-ML detectors in the previous section do not support parallel implementation. The development of ML technology, the continuous progress of signal processing chips, and support for parallel implementations enable ML solutions to have the advantages of having faster training times, better performance as they allow the algorithm to make more efficient use of available resources which can reduce the time required to reach a solution, and scalability as they allows the algorithm to make use of additional resources as needed.

2.5 References

[10] E. A. Lee and D. G. Messerschmitt, "Digital communication," *Springer Science Business Media*, 2012 (cit. on p. 10).

[11] R. Mancini, “Op amps for everyone: Design reference,” *Newnes*, 2003 (cit. on p. 10).

[12] A. Goldsmith, “Wireless communications,” *Cambridge University Press*, 2005 (cit. on p. 10).

[13] E. Bedeer, H. Yanikomeroglu, and M. H. Ahmed, “Reduced complexity optimal detection of binary faster-than-Nyquist signaling,” in *Proceeding of the IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6 (cit. on pp. 12, 18, 26, 37, 47).

[14] E. Bedeer, M. H. Ahmed, and H. Yanikomeroglu, “Low-complexity detection of high-order QAM faster-than-Nyquist signaling,” *IEEE Access*, vol. 5, pp. 14 579–14 588, Jul. 2017 (cit. on pp. 13, 18).

[15] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of computation*, vol. 44, no. 170, pp. 463–471, 1985 (cit. on p. 13).

[16] B. Hassibi and H. Vikalo, “On the sphere-decoding algorithm I. Expected complexity,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Jul. 2005 (cit. on p. 14).

[17] J. B. Anderson, F. Rusek, and V. Öwall, “Faster-than-Nyquist signaling,” *Proceedings of the IEEE*, vol. 101, no. 8, pp. 1817–1830, Mar. 2013 (cit. on pp. 16, 19).

[18] J. E. Mazo, “Faster-than-Nyquist signaling,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1451–1462, Oct. 1975 (cit. on pp. 17, 26, 47).

[19] C.-K. Wang and L.-S. Lee, “Practically realizable digital transmission significantly below the Nyquist bandwidth,” *IEEE Transactions on Communications*, vol. 43, no. 2, pp. 166–169, Dec. 1995 (cit. on p. 17).

[20] N. Seshadri, “Error performance of trellis modulation codes on channels with severe intersymbol interference,” Ph.D. dissertation, Rensselaer Polytechnic Institute, Sep. 1986 (cit. on p. 17).

[21] A. D. Liveris and C. N. Georghiades, “Exploiting faster-than-nyquist signaling,” *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1502–1511, Sep. 2003 (cit. on p. 17).

[22] J. B. Anderson, A. Prlja, and F. Rusek, “New reduced state space BCJR algorithms for the ISI channel,” in *Proceeding of the IEEE International Symposium on Information Theory*, Jun. 2009, pp. 889–893 (cit. on pp. 18, 47).

[23] S. Li, B. Bai, J. Zhou, P. Chen, and Z. Yu, “Reduced-complexity equalization for faster-than-Nyquist signaling: New methods based on Ungerboeck observation model,” *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1190–1204, Nov. 2017 (cit. on pp. 18, 28).

[24] E. Bedeer, M. H. Ahmed, and H. Yanikomeroglu, “A very low complexity successive symbol-by-symbol sequence estimator for faster-than-Nyquist signaling,” *IEEE Access*, vol. 5, pp. 7414–7422, Mar. 2017 (cit. on pp. 18, 26, 47).

[25] M. Kulhandjian, E. Bedeer, H. Kulhandjian, C. D’Amours, and H. Yanikomeroglu, “Low-complexity detection for faster-than-Nyquist signaling based on probabilistic data association,” *IEEE Communications Letters*, vol. 24, no. 4, pp. 762–766, Dec. 2019 (cit. on pp. 18, 26).

[26] E. Bedeer, H. Yanikomeroglu, and M. H. Ahmed, “Low-complexity detection of M-ary PSK faster-than-Nyquist signaling,” in *Proceeding of the IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, Apr. 2019, pp. 1–5 (cit. on p. 18).

[27] A. Ibrahim, E. Bedeer, and H. Yanikomeroglu, “A novel low complexity faster-than-Nyquist (FTN) signaling detector for ultra high-order QAM,” *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2566–2580, Nov. 2021 (cit. on pp. 18, 37).

[28] J. Fan, Y. Ren, Y. Zhang, and X. Luo, “MLSE equalizer with channel shortening for faster-than-Nyquist signaling,” *IEEE Photonics Technology Letters*, vol. 30, no. 9, pp. 793–796, Mar. 2018 (cit. on p. 19).

[29] S. Li, J. Yuan, B. Bai, and N. Benvenuto, “Code-based channel shortening for faster-than-Nyquist signaling: Reduced-complexity detection and code design,” *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 3996–4011, Apr. 2020 (cit. on p. 19).

[30] X. Wen, W. Yuan, D. Yang, N. Wu, and J. Kuang, “Low complexity message passing receiver for faster-than-Nyquist signaling in nonlinear channels,” *IEEE Access*, vol. 6, pp. 68 233–68 241, Oct. 2018 (cit. on p. 19).

[31] S. Li, N. Wu, Q. Shi, and Q. Guo, “FTN signaling-aided space-time multi-mode index modulation systems with a GMP-based receiver,” *IEEE Access*, vol. 7, pp. 162 898–162 912, Nov. 2019 (cit. on p. 19).

[32] W. Yuan, N. Wu, A. Zhang, X. Huang, Y. Li, and L. Hanzo, “Iterative receiver design for FTN signaling aided sparse code multiple access,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 915–928, Nov. 2019 (cit. on p. 19).

[33] K. Wang, A. Liu, X. Liang, S. Peng, and Q. Zhang, “A faster-than-Nyquist (FTN)-based multicarrier system,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 947–951, Aug. 2018 (cit. on p. 19).

[34] T. Ishihara, S. Sugiura, and L. Hanzo, “The evolution of faster-than-Nyquist signaling,” *IEEE Access*, vol. 9, pp. 86 535–86 564, Jun. 2021 (cit. on pp. 19, 26).

[35] P. Song, F. Gong, Q. Li, G. Li, and H. Ding, “Receiver design for faster-than-Nyquist signaling: Deep-learning-based architectures,” *IEEE Access*, vol. 8, pp. 68 866–68 873, Apr. 2020 (cit. on p. 20).

[36] B. Liu, S. Li, Y. Xie, and J. Yuan, “A novel sum-product detection algorithm for faster-than-Nyquist signaling: A deep learning approach,” *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5975–5987, Jun. 2021 (cit. on pp. 20, 26).

3 Deep Learning-based List Sphere Decoding for Faster-than-Nyquist (FTN) Signaling Detection

Sina Abbasi and Ebrahim Bedeer[1]

Abstract

Faster-than-Nyquist (FTN) signaling is a candidate non-orthonormal transmission technique to improve the spectral efficiency (SE) of future communication systems. However, such improvements of the SE are at the cost of additional computational complexity to remove the intentionally introduced intersymbol interference. In this paper, we investigate the use of deep learning (DL) to reduce the detection complexity of FTN signaling. To eliminate the need of having a noise whitening filter at the receiver, we first present an equivalent FTN signaling model based on using a set of orthonormal basis functions and identify its operation region. Second, we propose a DL-based list sphere decoding (DL-LSD) algorithm that selects and updates the initial radius of the original LSD to guarantee a pre-defined number N_{L} of lattice points inside the hypersphere. This is achieved by training a neural network to output an approximate initial radius that includes N_{L} lattice points. At the testing phase, if the hypersphere has more than N_{L} lattice points, we keep the N_{L} closest points to the point corresponding to the received FTN signal; however, if the hypersphere has less than N_{L} points, we increase the approximate initial radius by a value that depends on the standard deviation of the distribution of the output radii from the training phase. Then, the approximate value of the log-likelihood ratio (LLR) is calculated based on the obtained N_{L} points. Simulation results show that the computational complexity of the proposed DL-LSD is lower than its counterpart of the original LSD by orders of magnitude.

3.1 Introduction

There are increasing demands to improve the spectral efficiency (SE) to meet the requirements of future communication systems. Faster-than-Nyquist (FTN) signaling is a promising

[1] The work in this Chapter has been accepted in Proceeding of the IEEE 95th Vehicular Technology Conference: (VTC2022-Spring).

candidate technology that can increase the data rate without increasing the transmission bandwidth [37]. In FTN signaling, the data symbols are transmitted at a rate of $1/(\tau T)$, $\tau \leq 1$, when compared to the Nyquist rate of $1/T$ when using T-orthogonal pulses, and hence, inter-symbol interference (ISI) is intentionally introduced.

The early contribution of Mazo[2] [18] showed that increasing the data rate by accelerating the sinc pulses carrying binary phase shift keying (BPSK) symbols up to $\tau = 0.802$ will not deteriorate the asymptotic error rate when compared to Nyquist signaling that operates in the same bandwidth. However, such improvement of the SE is at the cost of prohibitive (at Mazo's time) computational complexity to remove the introduced ISI. In the past decade, there have been several research works based on conventional signal processing and estimation theory that detect the transmit data symbols of FTN signaling at reduced computational complexity, e.g., [13], [24], [25], [38]. We refer the reader to [37] for a summary of key FTN signaling detection techniques and to [34] for a more recent survey.

Recently, the application of deep learning (DL) to physical layer problems shows promising results mainly when there is a lack of appropriate mathematical models, i.e., model deficit, or a lack of low complexity algorithms, i.e., algorithm deficit [39]. Given the fast development of artificial intelligence chips, it is expected that DL will find more applications in physical layers problems.

The applications of DL have been extended to design FTN signaling systems in [36], [40]. In particular, Song et al. proposed an efficient DL-based architecture for FTN receivers that can replace either the signal detection block or both the signal detection and channel decoding blocks for uncoded and coded FTN signaling, respectively [40]. Their proposed DL-based FTN receivers showed near optimal performance for non-severe ISI operating regions. Liu et al. proposed a DL-based sum-product algorithm for FTN signaling that operates on a modified factor graph and concatenates a neural network function node to the variable nodes to approximate the optimal error rate performance [36].

Against the aforementioned literature, in this paper, we investigate the use of DL to reduce

[2] Mazo's contribution was an experimental work.

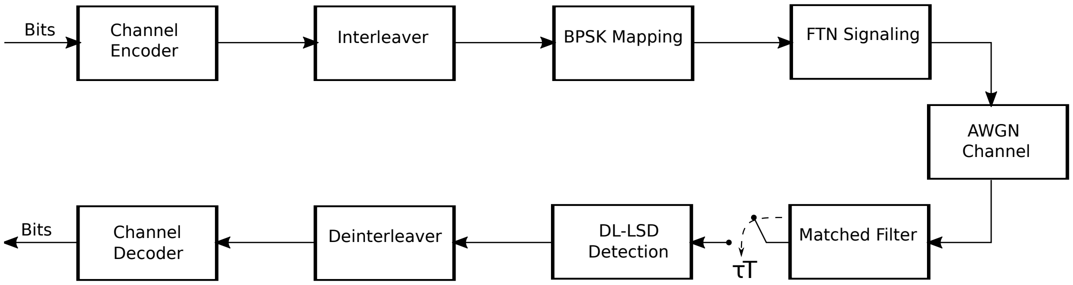

Figure 3.1: Block diagram of an FTN signaling system.

the detection complexity of FTN signaling. To eliminate the need of having a noise whitening filter at the receiver, we first present an equivalent transmission model for FTN signaling with the help of orthonormal basis functions, and we show its operation region. Second, we propose a DL-based list sphere decoding (DL-LSD) algorithm that selects and updates the initial radius of the original LSD to guarantee a pre-defined number N_{L} of lattice points inside the hypersphere. This is achieved by training a neural network to output an approximate initial radius that includes N_{L} lattice points. During the testing phase, if the hypersphere has more than N_{L} lattice points, we keep the N_{L} closest points to the point corresponding to the received FTN signal; however, if the hypersphere has less than N_{L} points, we increase the approximate initial radius by a value that depends on the standard deviation of the distribution of the output radii from the training phase. Then, the approximate value of the log-likelihood ratio (LLR) is calculated based on the N_{L} points. Simulation results show that the average number of flops of the proposed DL-LSD algorithm is three order and one order of magnitude lower than its counterpart of the original LSD, with a selection of the initial radius based on the noise variance, at low and high E_b/N_0 values, respectively.

The rest of the paper is organized as follows. In Section 3.2, we present an equivalent transmission model for FTN signaling based on using a sum of orthonormal basis; while in Section 3.3 we discuss the proposed DL-LSD algorithm. Simulation results are presented in Section 3.4, and the paper is concluded in Section 3.5.

3.2 System Model and Problem Formulation

Figure 1 shows a block diagram of an FTN signaling communication system. At the transmitter side, information bits are encoded, interleaved, and then mapped to data symbols where each symbol is carried by a unit-energy pulse $h(t)$. The widely used FTN signaling model expresses the transmit signal $s(t)$ as:

$$s(t) = \sum_{n} a_n h(t - n\tau T), \tag{3.1}$$

where $0 < \tau \leq 1$ is the time acceleration factor, T is the symbol duration, and $a_n, n = 1, ..., N,$[3] is the binary phase shift keying (BPSK) data symbol with average bit energy E_b. In our work, we assume that $h(t)$ is a T-orthogonal root raised cosine (rRC) pulse with a roll-off factor β_h. However, such transmission of non-orthogonal pulses in additive white Gaussian noise (AWGN) will require additional discrete-time whitening filter at the receiver to process the colored noise samples after the matched filter. Designing an exact causal and stable discrete-time whitening filter can be challenging at small values of τ [41]. To avoid using a whitening filter, one possibility is to use models based on the Ungerboeck observation model that deals directly with the colored noise, e.g. [23]. Another possibility which we adopt in this work is to use an equivalent FTN signaling model that uses a set of orthonormal basis function to whiten the noise samples after the matched filter. This model appeared in [42], [43] but has not received enough attention in the state-of-the-art literature, and it will be discussed here in detail for completeness of the presentation.

In the equivalent FTN signaling model based on orthonormal basis functions, the T-orthogonal pulse $h(t)$ is approximated as a sum of τT-orthonormal pulses $v(t - n\tau T)$ as:

$$h(t) \approx \sum_{n} h_n v(t - n\tau T). \tag{3.2}$$

In Lemma 1, we discuss how to find the constant coefficient h_n such that the approximation in (3.2) is valid.

[3] N is a pre-defined transmission block size based on channel encoder structure.

Lemma 1. *For a T-orthogonal $h(t)$ pulse, where $H(f) = 0$, $|f| > W$ and $W < 0.5/(\tau T)$, let a τT-orthonormal pulse $v(t)$ have the Fourier transform:*

$$V(f) = \begin{cases} C_o, & |f| < W, \\ 0, & |f| > \frac{1}{\tau T} - W, \end{cases} \tag{3.3}$$

where C_o is a constant. Then $h(t)$ may be expressed as $h(t) = \sum_n h_n v(t - n\tau T)$, where:

$$h_n = \frac{\tau T}{C_o} h(n\tau T). \tag{3.4}$$

Proof: see Appendix.

As one can see from Lemma 1, $h(t)$ can be approximated as a sum of τT-orthogonal basis functions $v(t - n\tau T)$ weighted by the scaled samples of $h(t)$ in (3.4), if $W < 0.5/(\tau T)$ and $V(f)$ is constant for $|f| < W$. For example and as shown in Fig. 3.2 (a), when $h(t)$ is a T-orthonormal rRC with a roll-off factor $\beta_h = 0.35$ with a bandwidth $W = 0.5(1 + \beta_h)/T$, it can be represented as a sum of 20 rRC τT-orthonormal pulses, i.e., $\sum_{n=1}^{20} h_n v(t - n\tau T)$, with a roll-off factor $\beta_v = 0.12$ and $\tau = 0.6$, if $W < 0.5/(\tau T)$, which yields:

$$\tau < \frac{1}{1 + \beta_h}. \tag{3.5}$$

Hence, the condition in (3.5) defines the operation region of the FTN signaling equivalent model. On the other hand, in Fig. 3.2 (b), $\tau = 0.9$ does not satisfy (3.5), and hence, the approximation is not accurate.

Given Lemma 1 and substituting (3.2) in (3.1), the equivalent FTN signaling transmit signal using the orthonormal basis function is expressed as:

$$s(t) = \sum_n b_n v(t - n\tau T), \tag{3.6}$$

where $b_n = \sum_l a_{n-l} h_l$ and h_l is given in (3.4). Assuming AWGN channel, the received signal is passed through a filter matched to the orthonormal basis $v(t)$, and for a real and symmetric $v(t)$, it is given as:

$$y(t) = (s(t) + w(t)) * v(t), \tag{3.7}$$

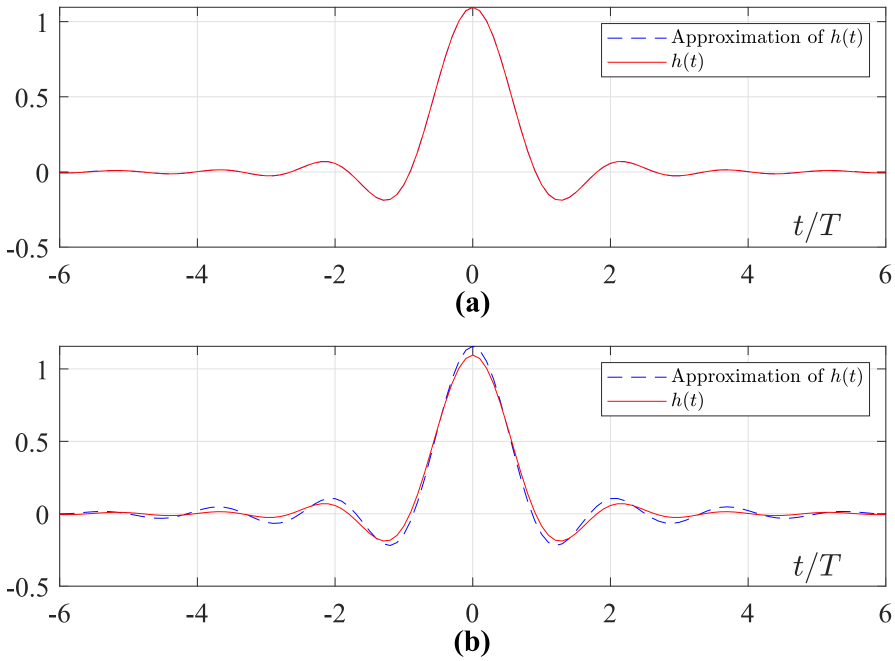

Figure 3.2: (a) $\tau = 0.6$, (b) $\tau = 0.9$. The solid line is the exact $h(t)$ pulse and the dashed line is its approximation based on (3.2).

where $w(t)$ is additive white Gaussian noise (AWGN) with zero mean and variance of σ^2 and $*$ denotes the convolution. Then, this signal is sampled every τT and is written as:

$$y_n = b_n + w_n. \tag{3.8}$$

The received sampled FTN signal can be expressed in a matrix form as: as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{a} + \boldsymbol{w}, \tag{3.9}$$

where $\boldsymbol{a}$ and $\boldsymbol{w}$ are the transmit data symbol and white noise vectors, respectively.

The received vector $\boldsymbol{y}$ needs to be processed by an FTN signaling detector to produce a soft output that can be used by the channel decoder. This soft output can be obtained from maximizing a posteriori probability (APP) for a given bit, and it is expressed as a log-likelihood ratio (LLR) value. The LLR for a bit x_k given the received vector $\boldsymbol{y}$ is written as:

$$L_D\left(x_k \mid \boldsymbol{y}\right) = \log \frac{P\left(\operatorname{map}(x_j) = +1 \mid \boldsymbol{y}\right)}{P\left(\operatorname{map}(x_j) = -1 \mid \boldsymbol{y}\right)}, \tag{3.10}$$

where x_k is the kth bit of $N \times 1$ vector $\boldsymbol{x}$ of all bits in one transmit block. We map the binary bits of 0 and 1 to -1 and $+1$, respectively. Assuming that $x_k, k = 0, ..., N-1$, are statistically independent, we use the Bayes theorem to re-write (3.10) as [44]:

$$L_D\left(x_k \mid \boldsymbol{y}\right) = \tag{3.11}$$
$$L_A\left(x_k\right) + \ln \frac{\sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_{k,+1}} p(\boldsymbol{y} \mid \boldsymbol{x}) \cdot \exp \sum_{j \in \mathcal{J}_{k,\boldsymbol{x}}} L_A\left(x_j\right)}{\sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_{k,-1}} p(\boldsymbol{y} \mid \boldsymbol{x}) \cdot \exp \sum_{j \in \mathcal{J}_{k,\boldsymbol{x}}} L_A\left(x_j\right)},$$

where $\boldsymbol{\mathcal{X}}$ is the set of all 2^N possible lattice points $\boldsymbol{x}$, $\boldsymbol{\mathcal{X}}_{k,+1} = \{\boldsymbol{x} \mid x_k = +1\}$, $\boldsymbol{\mathcal{X}}_{k,-1} = \{\boldsymbol{x} \mid x_k = -1\}$, $\mathcal{J}_{k,\boldsymbol{x}} = \{j | j = 0, ..., N-1, j \neq k, x_j = 1\}$, and

$$L_A(x_j) = \ln \frac{P(\text{map}(x_j) = +1)}{P(\text{map}(x_j) = -1)}, \tag{3.12}$$

and the likelihood function $p(\boldsymbol{y} \mid \boldsymbol{x})$ is given as follow:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2\sigma^2} \cdot \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{a}\|^2\right)}{\left(2\pi\sigma^2\right)^N}. \tag{3.13}$$

3.3 Proposed DL-LSD Algorithm

3.3.1 Review of the LSD Algorithm

Calculation of the LLR value for each bit in (3.11) needs to consider the whole possible lattice points in $\boldsymbol{\mathcal{X}}$, which has the size of 2^N of the skewed lattice points. Since for each bit x_k we iterate over all lattice points in $\boldsymbol{\mathcal{X}}$ and the calculation inside the exp function takes $O(N)$, and each transmit block has N bits in total; then, the computational complexity of the LLR values of one transmit block is at the order of $O(2^N N^2)$. For example, when the transmission block has $N = 25$ symbols; then the set $\boldsymbol{\mathcal{X}}$ has 2^{25} N-dimensional points. Accordingly, calculating the (3.11) for all bits within the transmit block requires $25 \times 2^{25} \approx 2 \times 10^{10}$ operations.

One can see from (3.13) that the conditional probability $p(\boldsymbol{y} \mid \boldsymbol{x})$ has an exponential relation with the distance of the skew lattice points to $\boldsymbol{y}$, i.e., $\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{a}\|^2$. That said, to reduce the complexity of the calculations of the LLR values in (3.11), we can consider a pre-defined number of points close to $\boldsymbol{y}$ rather all possible points in $\boldsymbol{\mathcal{X}}$. Finding the closest number of pre-defined points to $\boldsymbol{y}$ can be obtained by modifying the SD to what is called the LSD [44]. The LSD finds the first N_{L} closest lattice points in the skew lattice $\boldsymbol{H}\boldsymbol{a}$ to the vector $\boldsymbol{y}$ corresponding to the received FTN signaling, and then, forms the candidate list $\boldsymbol{\mathcal{L}}$.

To form the candidate list $\boldsymbol{\mathcal{L}}$, the SD is modified as follows. When a lattice point is found inside the hypersphere, the initial radius of the hypersphere is not reduced to the distance of that lattice point; rather, we add this lattice point to our list $\boldsymbol{\mathcal{L}}$. However, if the size of $\boldsymbol{\mathcal{L}}$ became $N_{\mathrm{L}}+1$, the lattice point with the largest distance to the vector $\boldsymbol{y}$ in $\boldsymbol{\mathcal{L}}$ is removed and the radius is updated to the largest distance to the vector $\boldsymbol{y}$ among all the remaining N_{L} lattice points in $\boldsymbol{\mathcal{L}}$. At the end and instead of using all the lattice points in $\boldsymbol{\mathcal{X}}$, the LSD algorithm finds the N_{L} closest points to the vector $\boldsymbol{y}$ that are to be used in the calculations of the approximate LLR values as follows:

$$\begin{aligned}&\tilde{L}_D\left(x_k \mid \boldsymbol{y}\right)= \\ &\tilde{L}_A\left(x_k\right)+\ln \frac{\sum_{\boldsymbol{x}\in\boldsymbol{\mathcal{L}}_{k,+1}} p(\boldsymbol{y}\mid\boldsymbol{x})\cdot\exp\sum_{j\in\mathcal{J}_{k,\boldsymbol{x}}}\tilde{L}_A\left(x_j\right)}{\sum_{\boldsymbol{x}\in\boldsymbol{\mathcal{L}}_{k,-1}} p(\boldsymbol{y}\mid\boldsymbol{x})\cdot\exp\sum_{j\in\mathcal{J}_{k,\boldsymbol{x}}}\tilde{L}_A\left(x_j\right)},\end{aligned} \tag{3.14}$$

where $\boldsymbol{\mathcal{L}}_{k,+1}=\{\boldsymbol{x}\in\boldsymbol{\mathcal{L}}\mid x_k=+1\}$, $\boldsymbol{\mathcal{L}}_{k,-1}=\{\boldsymbol{x}\in\boldsymbol{\mathcal{L}}\mid x_k=-1\}$. Also, $\tilde{L}_A$ is obtained similar to L_A but by considering lattice points inside $\boldsymbol{\mathcal{L}}$ instead of the whole lattice $\boldsymbol{\mathcal{X}}$.

As can be seen from (3.14), the computational complexity to approximate the LLR value of each transmit block of symbols reduces to $O(N_L N^2)$ because we consider the N_{L} elements in $\boldsymbol{\mathcal{L}}$ instead of whole 2^N lattice points. For example, if we consider $N_L = 32$ and $N = 25$, the calculation of (3.14) for all bits within the transmit block requires $25\cdot 25^2 \approx 1.5\times 10^3$ operations which is way less than 2×10^{10} required to calculate the exact LLRs.

On one hand, selecting the initial radius of the LSD to be of large value will lead to a comparable complexity to the exhaustive search due to the large number of lattice points inside the hypersphere. On the other hand, selecting the initial radius to be of small value may not guarantee to have N_{L} lattice points, and hence, degrade the approximation quality of the LLR values in (3.14). Hence, it is clear from the previous discussion that the selection of the initial radius of the LSD to have N_{L} lattice points is crucial to reduce its tree search complexity while maintaining an acceptable approximation of the LLR values. That said, we propose a DL-LSD algorithm to find the proper initial radius that guarantees to have N_{L} lattice points.

3.3.2 The Training Phase of the Proposed DL-LSD Algorithm

The intuition behind our proposed DL-LSD algorithm is estimating the initial radius to guarantee a pre-defined number N_{L} of lattice points inside the hypersphere. Similar idea for estimating the initial radius that guarantees at least one point inside the hypersphere has been proposed by Mostafa et al. [45]. This radius estimation problem is a non-linear regression problem, and neural networks (NNs) have shown success in solving such problems [46]. That said, we propose to use NNs to predict the initial radius that guarantees to include a pre-defined number N_{L} of lattice points to the received FTN signaling vector $\boldsymbol{y}$.The training data are obtained from the implementation of the LSD. Then, we feed the NN with the received vector $\boldsymbol{y}$ as an input, and we consider the distance of the furthest point in $\boldsymbol{\mathcal{L}}$ from $\boldsymbol{y}$ as the desired radius for training the output of the NN. Therefore, the set of input-output pairs $\{\boldsymbol{y}^{(i)}, R^{(i)}\}$ is used to train our NN, where $R^{(i)}$ is the largest radius in $\boldsymbol{\mathcal{L}}^{(i)}$, and $i = 1, ..., S$, where S is the size of training data set. The NN, f, predicts the initial radius $\hat{R}$ at its output layer as:

$$\hat{R}^{(i)} = f(\boldsymbol{y}^{(i)}, \theta), \tag{3.15}$$

where θ is the set of all parameters of NN, i.e, weights and biases values. Please note that the input to the NN $\boldsymbol{y}^{(i)}$ captures the effect of the ISI in $\boldsymbol{H}$ based on 3.9. Since we train the NN for each value of τ, for which the ISI matrix $\boldsymbol{H}$ will be the same for all training data, we decided to not feed the NN with $\boldsymbol{H}$ directly.

The first and last layers are the input and output layers, respectively; while the three middle layers are the hidden layers. The first two hidden layers are recurrent neural network

(RNN)[4] layers with 128 neurons and a simple fully connected layer with 64 neurons is used as the third hidden layer. Please note that the number of hidden layers and the number of neurons in each layer has been chosen experimentally. We use the activation function Relu for all hidden layers and it is defined as $\mathrm{Relu}(u) = \max(0, u)$. We use the mean square error (MSE) to evaluate the prediction error of the initial radius, and it is defined as:

$$L(\theta) = \frac{1}{|S|} \sum_{i=1}^{S} \left(R^{(i)} - f\left(\boldsymbol{y}^{(i)}, \theta\right)\right)^2, \tag{3.16}$$

where the desired radius $R^{(i)}$ is output when $\boldsymbol{y^{(i)}}$ is used as an input. An approximation of (3.16) in each iteration t over one training epoch can obtain as follow:

$$\tilde{L}_t(\theta) = \frac{1}{|S_t|} \sum_{i \in S_t} \left(R^{(i)} - f\left(\boldsymbol{y}^{(i)}, \theta\right)\right)^2, \tag{3.17}$$

where we divide our data set S to B mini-batches, each mini-batch S_t has a size of $|S_t| = |S|/B$. The complexity of the gradient computation is remarkably reduced when we increase the number of mini-batches B, while the variance of updating the NN parameter, i.e., θ, still decreases. Finally, Adam [47] is used as an optimization method for updating θ.

3.3.3 The Testing Phase of the Proposed DL-LSD Algorithm

In the testing phase, the received FTN signaling $\boldsymbol{y}$ is fed to the trained NN, and the NN gives the estimation of initial radius $\hat{R}$ that approximately guarantees to have N_{L} lattice points inside the hypersphere centered at $\boldsymbol{y}$. Then the LSD algorithm is executed with an initial radius equal to the obtained initial radius from the NN, i.e., $d = \hat{R}$. However, there is a possibility that the d is large enough to include more than N_{L} points at the testing phase. In this case, we keep only the N_{L} points with the smallest radii from $\boldsymbol{y}$ and discard the extra points with the largest radii. In case d is not large enough to have at least N_{L} points at the testing phase, we propose to increase the radius d by a value δ_d, and then, execute the LSD algorithm with new radius $d + \delta_d$. The selection of δ_d can be explained with the help of Fig.

[4] The motivation behind using RNN at the first two layer is the ISI structure of FTN signaling, because the input data of the NN model is similar to sequential data where RNN is much powerful in exploiting sequential data.

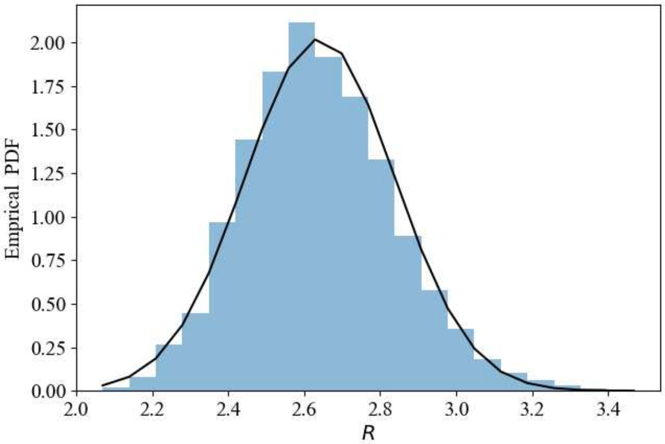

Figure 3.3: The histogram of the obtained radii of the training phase for $\tau = 0.6$, $\beta_h = 0.35$, $\beta_v = 0.12$, and $E_b/N_0 = 8$ dB.

3.3 as follows. In Fig. 3.3, we sketch the empirical probability density function (PDF) of all the obtained radii from the training phase at $\tau = 0.6$, $\beta_h = 0.35$, $\beta_v = 0.12$, and $E_b/N_o = 8$. We observe that the empirical PDF can be approximated as a Gaussian distribution with a standard deviation δ_d. That said, in case the initial radius d has less than N_{L}, we increase the radius by δ_d. The proposed DL-LSD algorithm is summarized at the top of this column. Finally, the approximate LLR values are calculated according to (3.14), and then passed to the channel decoder as soft inputs to estimate the transmit data symbols $\hat{\boldsymbol{a}}$.

3.4 Simulation Results

In this section, we investigate the performance of the proposed DL-LSD to detect coded BPSK FTN signaling. We consider a standard convolutional code (7, [171 133]) to encode the information bits at the transmitter and a Viterbi decoder to decode the approximate soft outputs of the proposed DL-LSD at the receiver. The roll-off factors β_h and β_v are set to 0.35 and 0.12, respectively. We consider $N = 25$ data symbols per block transmission and an acceleration factor of $\tau = 0.6$ and 0.74. Please note that the choice of these values of τ meets the condition in (3.5).

The implementation steps of the proposed algorithm are as follows: we first create uniform

The Proposed DL-LSD Algorithm

```
Input: $\boldsymbol{H}$, $\boldsymbol{y}$, $\delta_d$, $f(.\,,\boldsymbol{\theta})$
Output: Calculated LLR values
$d \leftarrow f(\boldsymbol{y},\boldsymbol{\theta})$                    ▷ Estimating radius with NN
while True do
    $\boldsymbol{\mathcal{L}} \leftarrow$ LSD($\boldsymbol{H}$, $\boldsymbol{y}$, $d$)          ▷ LSD algorithm returns a list
    if $|\boldsymbol{\mathcal{L}}| < N_{\mathrm{L}}$ then
        $d = d + \delta_d$                              ▷ Increasing radius
    else
        $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}(1 : N_{\mathrm{L}})$        ▷ Picking first $N_{\mathrm{L}}$ closet point to $\boldsymbol{y}$
        break                                   ▷ breaking the while loop
    end if
end while
LLR($\boldsymbol{\mathcal{L}}$)                      ▷ Calculation of LLR based on (3.14)
```

random data and then we implemented the LSD algorithm by Hassibi et al. [48] to label our dataset. Then, we used the Keras library to design the structure of DL-LSD as discussed in Section 3.3. After saving the trained model, we have implemented the proposed algorithm based on the provided pseudo code in Section 3.3. This work is totally implemented in the Python3 environment.

The training of the proposed DL-LSD can be summarized as follows. For $E_b/N_0 = 4$ and 6 dB, we use 800 blocks of random data symbols; while for $E_b/N_0 = 8$ and 10 dB, we use 8000 blocks. For each of the training blocks, the number of random data symbols per block is set to $N = 25$. Please note that the low number of blocks used to train the NN at $E_b/N_0 =$ 4 and 6 dB is due to the huge computational complexity required to obtain the training data symbols from the original LSD that selects the initial radius based on the noise variance [48]. Also, since the operating region for our system model is $E_b/N_0 > 6$ dB, where the bit error rate is less than 10^{-3}, we tried to have a well-trained model with more number of data points. We experimentally set the learning rate of the Adam optimizer to 0.0001, and the mini-batch size S_t is set to 20. We labeled the dataset by implementing the LSD algorithm from Hassibi et al. [48], then since the data for the training process are generated uniformly random, we set the initial state of the learning parameters of the neural network randomly. The early stopping has been implemented to ensure the model does not fall into over-fitting scenarios. That being said, the number of neurons in each layer has been chosen experimentally; other types of neural network structure such as Long short-term memory (LSTM) and simple fully connected layers without RNN has been tried and compared to the proposed structure, where all had greater MSE in both testing and validation sets. For hyper-parameter tuning, it is suggested to try the different numbers of layers and the different numbers of neurons on each layer. The model is very sensitive to the learning rate of the Adam optimizer we suggest not changing the learning rate, or alternatively trying different learning decay methods. Also, it is suggested to change different weight initialization such as He weight initialization we used the default weight initialization in Keras, i.e., Glorot Uniform; because the model is not sensitive to the weight initialization.

As discussed earlier, the aim of the proposed DL-LSD algorithm is to select a number of lattice points N_{L} to approximate the calculations of the LLR value of each bit in (3.14) without deteriorating the error rate performance. Comparisons of the performance of the sphere decoding with other low complexity FTN signaling detection techniques can be found in [13], [27]. To strike a balance between the computational complexity and the BER performance, we plot in Fig. 3.4 the BER performance for different values of N_{L} at $\tau = 0.6$. As can be seen, the value of $N_L = 32$ shows negligible BER loss when compared to $N_L = 128$, while significantly reduces the complexity of calculating (3.14). Hence, we adopt the value of N_{L} to be 32 in the rest of the simulation results.

Fig. 3.5 depicts the average number of lattice points inside the hypersphere of both the original-LSD and proposed DL-LSD versus E_b/N_0 for $\tau = 0.6$. Please note that the average number of lattice points is calculated based on averaging the results of 10 transmit blocks. As can be seen, the average number of lattice points obtained by the proposed DL-LSD algorithm is close to the target value of $N_L = 32$, and more importantly, is insensitive to the noise power. This is in contrast to the original-LSD where the initial radius is set based on the noise variance [48], and hence, can have a large number of lattice points inside hypersphere

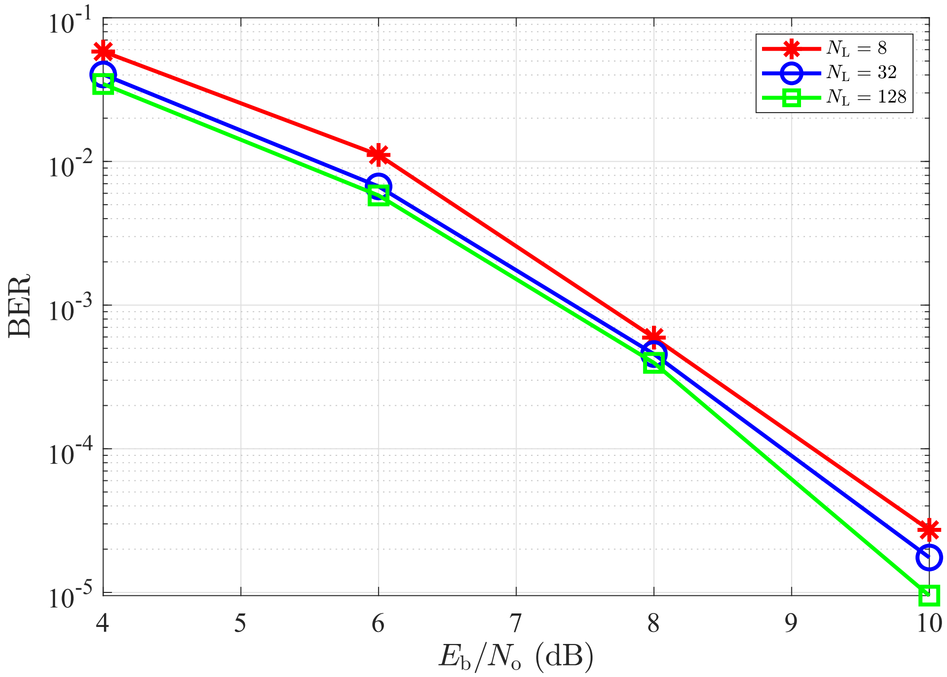

Figure 3.4: BER as a function of E_b/N_0 at $\tau = 0.6$ for different values of N_L.

at low E_b/N_0. Since the complexity of the tree search exponentially increases with increasing the number of lattice points inside the hypersphere, the proposed DL-LSD is expected to have a reduced complexity when compared to the original-LSD.

To quantify the reduction of the computational complexity of the proposed DL-LSD algorithm with respect to the original-LSD, in Fig. 3.6 we plot the ratio of the number of floating point operations (flops) of the proposed DL-LSD to the number of flops in original-LSD as a function of E_b/N_0 for $\tau = 0.6$. A flop serves as a basic unit of computation, and it denotes one addition, subtraction, multiplication, or division of floating point numbers. To have a fair complexity comparison, both DL-LSD and the original-LSD use the same implementation of the LSD algorithm but they are different only in the selection of the initial radius (the proposed DL-LSD algorithm estimates the initial radius from the trained NN, while the original-LSD estimates the initial radius based on the noise variance as introduced by Hassibi et al. [48]). As one can see, the proposed DL-LSD algorithm has more than three orders of magnitude lower number of flops when compared to the original-LSD algorithm for low values of E_b/N_0. For high values of E_b/N_0, the proposed algorithm achieves an order of

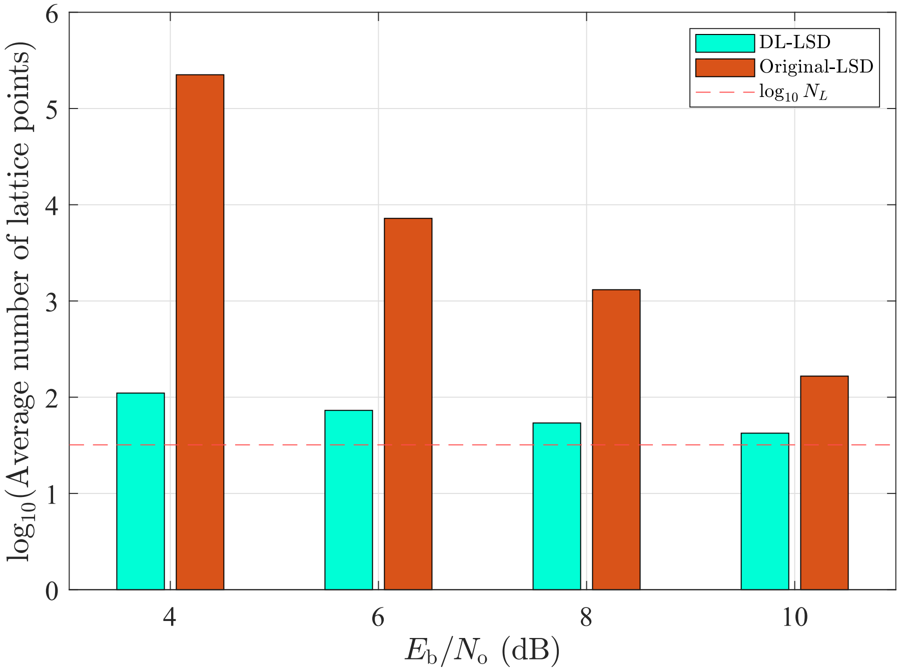

Figure 3.5: Comparison of the average number of lattice points inside the hypersphere of the proposed DL-LSD algorithm and the original-LSD at $\tau = 0.6$.

magnitude lower number of flops.

In Fig. 3.7, we depict the BER of the uncoded and coded FTN signaling as a function of E_b/N_0 for different values of τ. As one can see, for the uncoded transmission at $\tau = 0.74$, the BER approaches its counterpart of Nyquist signaling which represents 35% in the SE at no increase in E_b/N_o. Decreasing the value of τ will results in an improvement in the SE but at the cost of increasing E_b/N_o. For the coded transmission, the proposed DL-LSD showed approximately savings of 1.5 dB in E_b/N_o when compared to the uncoded results at both $\tau = 0.74$ and 0.6 at BER of 10^{-4}.

3.5 Conclusion

FTN signaling can improve the SE without increasing the transmission bandwidth, and hence, it is a promising technology for future communication systems. In this paper, we presented an equivalent transmission model for FTN signaling that uses a set of orthonormal basis

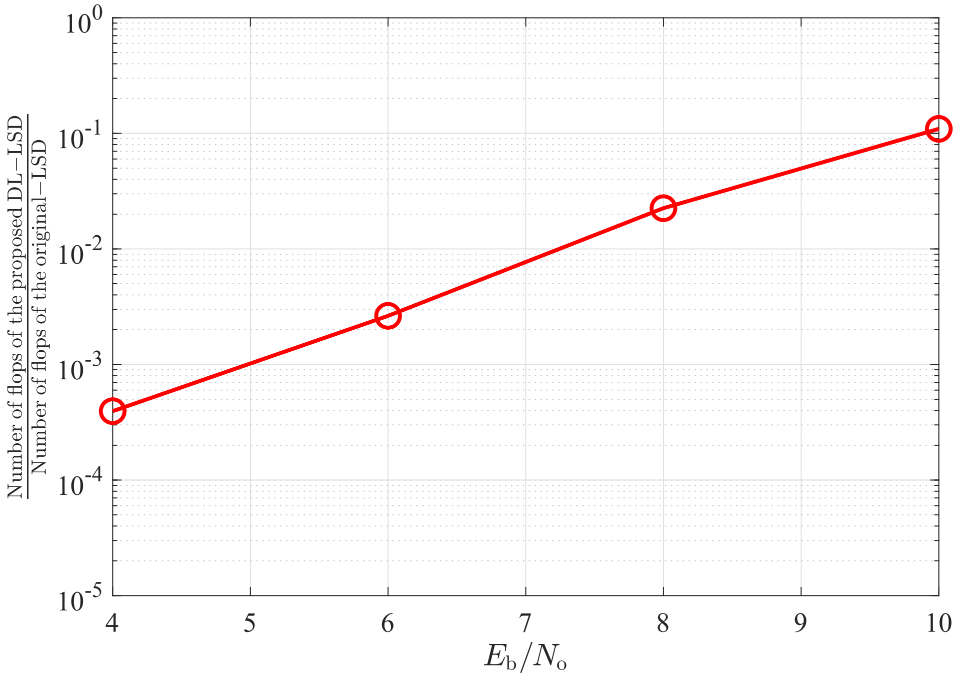

Figure 3.6: Comparison of the average number of flops of the proposed DL-LSD algorithm and the original-LSD at $\tau = 0.6$.

functions to eliminate the need to design a noise whitening filter at the receiver. We then proposed a DL-LSD algorithm to learn and update an approximate initial radius to include a certain number of points N_{L} inside the hypersphere. In case the initial radius has less than N_{L} points, we increase the approximate initial radius by a value that depends on the standard deviation of the distribution of the output radii from the training phase. Simulation results showed that the average number of flops of the proposed DL-LSD algorithm is three order and one order of magnitude lower than its counterpart of the original LSD, with a selection of the initial radius based on the noise variance, at low and high E_b/N_0 values, respectively.

Appendix

Proof of Lemma 1

The proof was introduced by Anderson et al. [43], and it is included here for completeness of the presentation. We define the discrete-time Fourier transform $H_s(f_d) = \sum_n h_n e^{-j2\pi f_d n}$,

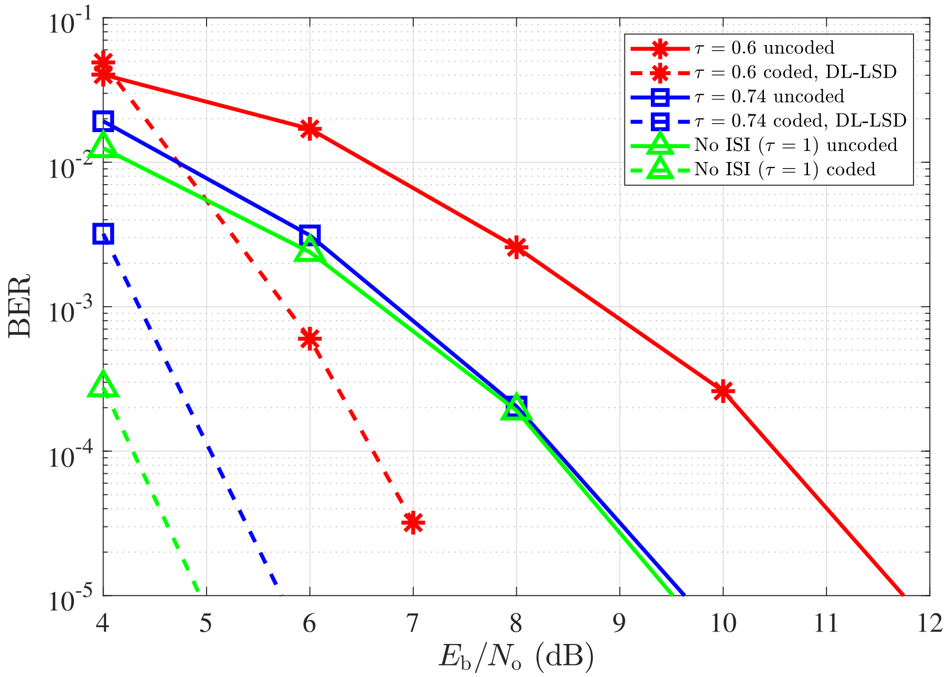

Figure 3.7: Coded and uncoded BER performance for different values of τ.

where $\{h_n\}$ is the sampled sequence of $h(t)$ every τT. From the properties of the Fourier transform we know that $H_s(f) = \frac{1}{\tau T}\sum_n H(f - \frac{n}{\tau T})$, where $H(f)$ is the continuous-time Fourier transform of $h(t)$, hence:

$$\begin{aligned} H(f) &= \tau T H_s(f) \\ &= \tau T \sum_n h(n\tau T) e^{-j2\pi f n \tau T}, \quad |f| \le \frac{1}{\tau T} - W. \end{aligned} \tag{3.18}$$

At the same time, from the definition of inverse continues-time Fourier transform we have $h(t) \triangleq \int H(f) e^{j2\pi f t} df$. Considering the fact that $V(f)$ is constant over the support of $H(f)$, i.e., $|f| < W$. Then, we can multiply $V(f)$ inside the integral and divide by the constant C_0 outside of the integral to have:

$$h(t) = \frac{1}{C_0} \int H(f) V(f) e^{j2\pi f t} df. \tag{3.19}$$

Since $V(f) = 0$ for $|f| \geq \frac{1}{\tau T} - W$, substituting (3.18) into (3.19) results in:

$$\begin{aligned} h(t) &= \frac{1}{C_0} \int \left[\tau T \sum_n h(n\tau T) e^{-j2\pi f n\tau T} \right] V(f) e^{j2\pi f t} df \\ &= \sum_n \left[\frac{\tau T h(n\tau T)}{C_0} \right] \int V(f) e^{j2\pi f (t - n\tau T)} df \\ &= \sum_n h_n v(t - n\tau T), \end{aligned} \tag{3.20}$$

where $h_n = \frac{\tau T}{C_0} h(n\tau T)$ which concludes the proof. ■

3.6 References

[13] E. Bedeer, H. Yanikomeroglu, and M. H. Ahmed, “Reduced complexity optimal detection of binary faster-than-Nyquist signaling,” in *Proceeding of the IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6 (cit. on pp. 12, 18, 26, 37, 47).

[18] J. E. Mazo, “Faster-than-Nyquist signaling,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1451–1462, Oct. 1975 (cit. on pp. 17, 26, 47).

[23] S. Li, B. Bai, J. Zhou, P. Chen, and Z. Yu, “Reduced-complexity equalization for faster-than-Nyquist signaling: New methods based on Ungerboeck observation model,” *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1190–1204, Nov. 2017 (cit. on pp. 18, 28).

[24] E. Bedeer, M. H. Ahmed, and H. Yanikomeroglu, “A very low complexity successive symbol-by-symbol sequence estimator for faster-than-Nyquist signaling,” *IEEE Access*, vol. 5, pp. 7414–7422, Mar. 2017 (cit. on pp. 18, 26, 47).

[25] M. Kulhandjian, E. Bedeer, H. Kulhandjian, C. D’Amours, and H. Yanikomeroglu, “Low-complexity detection for faster-than-Nyquist signaling based on probabilistic data association,” *IEEE Communications Letters*, vol. 24, no. 4, pp. 762–766, Dec. 2019 (cit. on pp. 18, 26).

[27] A. Ibrahim, E. Bedeer, and H. Yanikomeroglu, “A novel low complexity faster-than-Nyquist (FTN) signaling detector for ultra high-order QAM,” *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2566–2580, Nov. 2021 (cit. on pp. 18, 37).

[34] T. Ishihara, S. Sugiura, and L. Hanzo, “The evolution of faster-than-Nyquist signaling,” *IEEE Access*, vol. 9, pp. 86 535–86 564, Jun. 2021 (cit. on pp. 19, 26).

[36] B. Liu, S. Li, Y. Xie, and J. Yuan, “A novel sum-product detection algorithm for faster-than-Nyquist signaling: A deep learning approach,” *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5975–5987, Jun. 2021 (cit. on pp. 20, 26).

[37] J. B. Anderson, F. Rusek, and V. Öwall, “Faster-than-Nyquist signaling,” *Proceedings of the IEEE*, vol. 101, no. 8, pp. 1817–1830, Mar. 2013 (cit. on p. 26).

[38] F. Rusek and J. B. Anderson, “Non binary and precoded faster than Nyquist signaling,” *IEEE Transaction on Communications*, vol. 56, no. 5, pp. 808–817, May 2008 (cit. on p. 26).

[39] O. Simeone, “A very brief introduction to machine learning with applications to communication systems,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, Dec. 2018 (cit. on p. 26).

[40] P. Song, F. Gong, Q. Li, G. Li, and H. Ding, “Receiver design for faster-than-Nyquist signaling: Deep-learning-based architectures,” *IEEE Access*, vol. 8, pp. 68 866–68 873, Apr. 2020 (cit. on pp. 26, 47).

[41] A. Prlja, J. B. Anderson, and F. Rusek, “Receivers for faster-than-Nyquist signaling with and without turbo equalization,” in *Proceedings of the IEEE International Symposium on Information Theory*, Jul. 2008, pp. 464–468 (cit. on p. 28).

[42] A. Prlja and J. B. Anderson, “Reduced-complexity receivers for strongly narrowband intersymbol interference introduced by faster-than-Nyquist signaling,” *IEEE Transactions on Communications*, vol. 60, no. 9, pp. 2591–2601, Sep. 2012 (cit. on p. 28).

[43] J. B. Anderson, *Coded Modulation Systems.* Kluwer Academic Publishers, 2002 (cit. on pp. 28, 40, 48, 49).

[44] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, Apr. 2003 (cit. on pp. 31, 32).

[45] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4368–4378, Sep. 2019 (cit. on p. 33).

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, Nov. 2016 (cit. on p. 33).

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, Dec. 2014 (cit. on p. 34).

[48] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Jul. 2005 (cit. on pp. 36–38).

4 Low Complexity Classification Approach for Faster-than-Nyquist (FTN) Signaling Detection

Sina Abbasi and Ebrahim Bedeer[1]

Abstract

Faster-than-Nyquist (FTN) signaling can improve the spectral efficiency (SE); however, at the expense of high computational complexity to remove the introduced intersymbol interference (ISI). Motivated by the recent success of machine learning (ML) in physical layer (PHY) problems, in this chapter we investigate the use of ML in reducing the detection complexity of FTN signaling. In particular, we view the FTN signaling detection problem as a classification task, where the received signal is considered as an unlabeled class sample that belongs to a set of all possible classes samples. If we use an off-shelf classifier, then the set of all possible classes samples belongs to an N-dimensional space, where N is the transmission block length, which has a huge computational complexity. We propose a low-complexity classifier (LCC) that exploits the ISI structure of FTN signaling to perform the classification task in $N_p \ll N$-dimension space. The proposed LCC consists of two stages: 1) offline pre-classification that constructs the labeled classes samples in the N_p-dimensional space and 2) online classification where the detection of the received samples occurs. The proposed LCC is extended to produce soft-outputs as well. Simulation results show the effectiveness of the proposed LCC in balancing performance and complexity.

4.1 Introduction

Improving the spectral efficiency (SE) is one of the main goals of next generation communication systems. Faster-than-Nyquist (FTN) signaling is one of the promising solutions to improve the SE, and this is achieved by increasing the data rate beyond the rate of conventional Nyquist communication systems while using the same transmission bandwidth. Essentially in FTN signaling, the transmit data symbols are sent at a rate of $1/(\tau T)$, $\tau \leq 1$, which is faster than the Nyquist rate of $1/T$. Such improvements in the SE come at the

[1] The work in this Chapter is under review in IEEE Communication Letters journal.

expense of inter-symbol interference (ISI) between the transmit symbols that requires extra processing at the transmitter and/or the receiver to achieve acceptable performance.

One of the early studies on FTN signaling was in 1975 [18] when Mazo in his experimental work proved that if we set the acceleration parameter τ between $0.802 \leq \tau \leq 1$, we maintain the same asymptotic error rate as the Nyquist signaling using the same bandwidth. However, this is at the cost of considerable computational complexity to compensate for the introduced ISI. Several works have been done, especially in the past decade, to reduce the detection complexity of FTN signaling. For instance, the works [13], [22], [24], [49] are focused on utilizing conventional estimation theory and signal processing methods for detecting FTN signaling.

Machine learning (ML) techniques have shown tremendous improvements in various domains, such as computer vision and natural language processing. Recently, there has been increasing interest in applying ML techniques in signal processing, and physical layer (PHY) problems [50]. In the context of FTN signaling, the work of Song et al. [40], and Abbasi and Bedeers [51], successfully reduced the detection complexity of FTN signaling receivers. Song et al. proposed two different deep learning (DL)-based architectures for FTN signaling receivers [40]. Abbasi and Bedeer proposed a DL-based algorithm [51], to approximate the initial radius of the list sphere decoding algorithm to detect FTN signaling. The proposed DL-based list sphere decoding (DL-LSD) considerably reduces the detection complexity when compared to the list sphere decoding.

Motivated by the recent success of ML in PHY problems, in this chapter we investigate the use of ML in reducing the detection complexity of FTN signaling. In particular, we view the FTN signaling detection problem as a classification task, where the received signal is considered as an unlabeled class sample that belongs to a set of all possible classes samples. If we use an off-shelf classifier, then the set of all possible classes samples belongs to an N-dimensional space, where N is the transmission block length, which has a huge computational complexity. We propose a low-complexity classifier (LCC) that exploits the ISI structure of FTN signaling to perform the classification task in $N_p \ll N$-dimension space. The proposed LCC consists of two stages: 1) offline pre-classification that constructs the labeled classes samples in the N_p-dimensional space and 2) online classification where the detection of the received samples occurs. The proposed LCC is extended to produce soft-outputs as well. Simulation results show the effectiveness of the proposed LCC in balancing the performance and complexity.

The rest of the chapter is organized as follows. In Section 4.2, we present the system model of FTN signaling. In Section 4.3, we discuss the proposed LCC, and its computational complexity analysis is introduced in Section 4.4. Simulation results are presented in Section 4.5, and in Section 4.6 we conclude the chapter.

We use calligraphic bold uppercase letters, e.g. $\boldsymbol{\mathcal{A}}$, for sets, bold uppercase letters, e.g. $\boldsymbol{A}$, for matrices, bold lowercase letters, e.g. $\boldsymbol{a}$, for vectors and a_i for pointing the ith element of vector $\boldsymbol{a}$. In addition, we use $\boldsymbol{a}^{(\boldsymbol{i})}$ to specify the elements of the vector $\boldsymbol{a}$ that are centered at the ith element.

4.2 System Model and Problem Formulation

4.2.1 FTN Signaling Model

We consider the transmission of a block of size N data symbol, $a_n, n = 1, ..., N$, that are carried by a unit-energy pulse $h(t)$. The conventional FTN signaling model formulates the transmit signal $s(t)$ as:

$$s(t) = \sum_n a_n h(t - n\tau T), \tag{4.1}$$

where $0 < \tau \leq 1$ is the time acceleration parameter, and τT is the symbol duration. In this chapter, we adopt an equivalent FTN signaling model based on the orthonormal basis functions [43], [51]. In the equivalent FTN signaling model, the T-orthogonal pulse $h(t)$ is approximated by the sum of multiple τT-orthonormal pulses as:

$$h(t) \approx \sum_n h_n v(t - n\tau T), \tag{4.2}$$

where h_n is a scaled sample of $h(t)$ at τT that is given as [43], [51]:

$$h_n = \sqrt{\tau T} h(n \tau T). \tag{4.3}$$

Substituting (4.2) and (4.3) into (4.1) gives us following equivalent expression for the transmit FTN signal:

$$s(t) = \sum_n b_n v(t - n\tau T), \tag{4.4}$$

where $b_n = \sum_l a_{n-l} h_l$. For a root-raised cosine pulse $h(t)$, its roll-off factor β_h must satisfy $\tau < 1/(1+\beta_h)$ for the equivalent model to hold [51]. The received signal after passing through a filter matched to $v(t)$ and sampling at every τT is written as:

$$y_n = b_n + w_n, \tag{4.5}$$

where w_n is the sampled white Gaussian noise with zero mean and σ^2 variance. The matrix expression of (4.5) is written as:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{a} + \boldsymbol{w}, \tag{4.6}$$

where the vector $[h_{-j}, ..., h_{-2}, h_{-1}, h_0, h_1, h_2, ..., h_{N-j-1}]$ is the jth row, $0 \leq j \leq N-1$, of the matrix $\boldsymbol{H}$.

4.2.2 FTN detection as a Classification Problem

One of ML's main types of tasks is the supervised learning, which includes two main categories: regression and classification problems. In regression problems, the task is to predict a continuous value; while in classification problems, the goal is to assign a new unlabeled data sample to one of the existing classes. For example, a binary classification problem is illustrated in Fig. 4.1 where each data sample is expressed by two features, i.e., u_1 and u_2. The known labeled samples belong to two classes circle and cross, while the square represents a new unlabeled data sample. The goal of any classification algorithm is to assign the proper label for the unlabeled data.

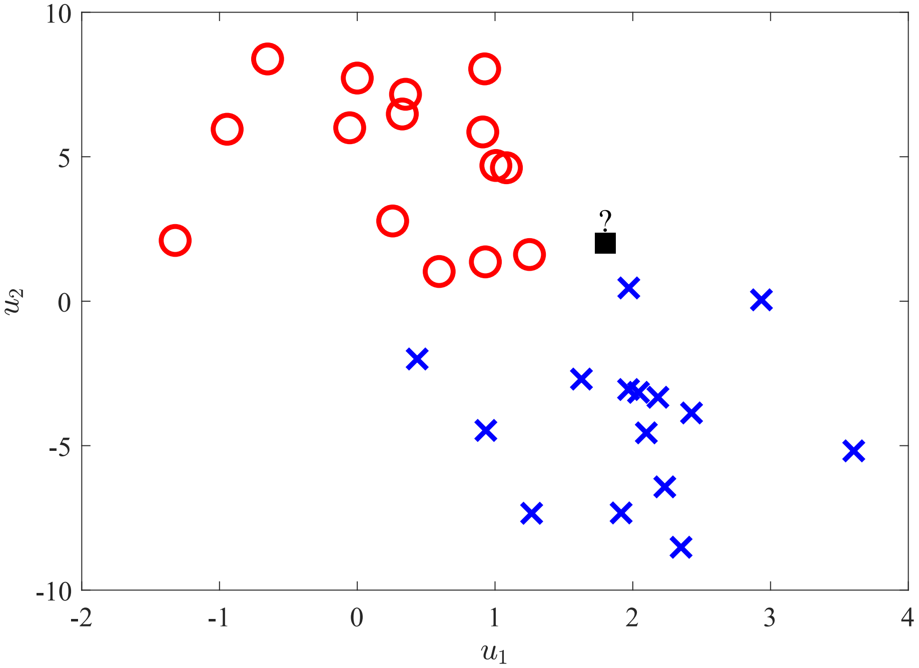

Figure 4.1: An example of binary classification problem.

For FTN signaling system model, the set of all possible data symbol blocks is defined as $\boldsymbol{\mathcal{M}} = \{\boldsymbol{m_1}, \boldsymbol{m_2}, ..., \boldsymbol{m_{2^N}}\}$, where $\boldsymbol{m_i}$, $i = 1, ..., N$, is a $N \times 1$ vector representing one of the possible odds for the transmit vector $\boldsymbol{a}$. We define $\boldsymbol{\mathcal{S}} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, ..., \boldsymbol{s_{2^N}}\}$ to be the set of all possible points in a skew lattice, where $\boldsymbol{s_i} = g(\boldsymbol{m_i}) = \boldsymbol{H}\boldsymbol{m_i}$ and $g(.)$ is an injective function.

In the context of classification, each $\boldsymbol{s_i}$ is a different class and we have 2^N different classes in total. Given the received vector $\boldsymbol{y}$, a classifier is defined as a function $f(.)$ such that:

$$\boldsymbol{y} \xrightarrow{f(.)} \boldsymbol{s}_k \xrightarrow{g(.)} \boldsymbol{m}_k,\ k \in \{1, 2, ..., 2^N\}. \tag{4.7}$$

In other words, the classifier $f(.)$ partitions $\boldsymbol{\mathcal{S}}$ into the 2^N different classes such that $\boldsymbol{s_1} \cap \boldsymbol{s_2} \cap ... \cap \boldsymbol{s_{2^N}} = \phi$ and $\boldsymbol{s_1} \cup \boldsymbol{s_2} \cup ... \cup \boldsymbol{s_{2^N}} = \boldsymbol{\mathcal{S}}$. Consequently, since $g(.)$ is an injective function, $f(g(.))$ also partitions $\boldsymbol{\mathcal{M}}$ into the 2^N possible transmit block of symbols of size N. Therefore, the received vector $\boldsymbol{y}$ can be detected and assigned to one of the elements in $\boldsymbol{\mathcal{M}}$.

4.2.3 Soft Output

Soft output from the FTN detection process is needed to be used by the channel decoder. The maximizing a posteriori probability (APP) for a given bit can be applied to achieve the soft-outputs, and generally is expressed as a log-likelihood ratio (LLR) value. Given the received vector $\boldsymbol{y}$, the LLR value for a bit x_k is obtained by:

$$L_D\left(x_k \mid \boldsymbol{y}\right)=\log \frac{P\left(x_k=1 \mid \boldsymbol{y}\right)}{P\left(x_k=0 \mid \boldsymbol{y}\right)}, \tag{4.8}$$

where x_k is the kth element of vector $\boldsymbol{x}=\operatorname{map}(\boldsymbol{a})$, which the map is the modulation mapping function symbols to bits. Assuming that $x_k, k=0,\ldots,N-1$, are statistically independent, we use the Bayes theorem to re-write (4.8) as [51]:

$$\begin{aligned} & L_D\left(x_k \mid \boldsymbol{y}\right)= \\ & L_A\left(x_k\right)+\ln \frac{\sum_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}_{k,1}} p(\boldsymbol{y} \mid \boldsymbol{x}) \cdot \exp \sum_{j\in\mathcal{J}_{k,\boldsymbol{x}}} L_A\left(x_j\right)}{\sum_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}_{k,0}} p(\boldsymbol{y} \mid \boldsymbol{x}) \cdot \exp \sum_{j\in\mathcal{J}_{k,\boldsymbol{x}}} L_A\left(x_j\right)}, \end{aligned} \tag{4.9}$$

where $\boldsymbol{\mathcal{X}}=\operatorname{map}(\boldsymbol{\mathcal{M}})$ is the set of all 2^N possible bits $\boldsymbol{x}$ which map is the mapping function from symbols to bits, $\boldsymbol{\mathcal{X}}_{k,1}=\{\boldsymbol{x} \mid x_k=1\}$, $\boldsymbol{\mathcal{X}}_{k,0}=\{\boldsymbol{x} \mid x_k=0\}$, $\mathcal{J}_{k,\boldsymbol{x}}=\{j|j=0,\ldots,N-1, j\neq k\}$, and

$$L_A(x_j)=\ln \frac{P(x_j=1)}{P(x_j=0)}, \tag{4.10}$$

and the likelihood function $p(\boldsymbol{y} \mid \boldsymbol{x})$ is given as follow:

$$p(\boldsymbol{y} \mid \boldsymbol{x})=\frac{\exp\left(-\frac{1}{2\sigma^2}\cdot\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{a}\|^2\right)}{\left(2\pi\sigma^2\right)^N}. \tag{4.11}$$

4.3 Proposed Low Complexity Classification of FTN Signaling

As discussed earlier, the number of classes in the conventional classification problem is 2^N, and hence, one of the biggest hindrances for such a conventional classification problem is the huge computational complexity, especially for long transmit block of data symbols, i.e., $N = 1000$. In the following, we propose a LCC that exploits the inherent structure of FTN signaling to reduce the classification computational complexity. To show the intuition behind the proposed LCC, we provide the following examples.

— *Example 1*: Let us assume a noise free transmission of N transmit symbols, where each symbol is affected only by ISI from one past and one upcoming symbol, i.e., $y_i = \sum_{\ell=-1}^{1} a_{i-\ell} h_\ell$ and $h_{-1} = 0.3, h_0 = 0.8, h_1 = 0.3$. At the receiver, let us intentionally ignore the ISI and detect each symbol independent from the adjacent symbols. One can show that for all the possible values of the transmit data symbols, the possible values of $y_i \in \{-1.4, -0.8, -0.2, 0.2, 0.8, 1.4\}$. These values of y_i are plotted on the horizontal axis in Fig. 4.2, where the cross and circle points represent the values of y_i corresponding to $a_i = 1$ and $a_i = -1$, respectively. If we consider the classification objective to be the nearest distance, then the dashed line in Fig. 4.2 shows the boundary between the two different classes, where the first class has the samples 0.2, 0.8, and 1.4 while the second class has the samples -0.2, -0.8, and -1.4. Then, the closest distance d between the two different classes samples is $0.2 + 0.2 = 0.4$. In the presence of the noise, the received sample y_i will deviate from these classes samples depending on the noise power, and the classifier detects the transmit symbol based on the closest distance to the two different classes' samples.

— *Example 2*: The detection of a transmit symbol by observing just one sample of the received vector $\boldsymbol{y}$, as discussed in Example 1, comes with significant performance degradation, and this is as each transmit symbol experiences ISI from other adjacent symbols. In Example 2, we re-consider the transmission scheme of Example 1, but the detection is done differently. In particular, we detect one symbol by jointly considering an upcoming sample in addition to the current sample, i.e., y_i, y_{i+1}. Let us consider Fig. 4.3, where the horizontal axis represents y_i and the vertical axis represents y_{i+1}. Similar to Example 1, cross and circle points correspond to $a_i = 1$ and $a_i = -1$; respectively, the dashed line shows the classification boundary. The closest distance between the two classes samples is $d = \sqrt{(0.2+0.2)^2 + (0.2+0.2)^2} = 0.57$ which is greater than its counterpart in Example 1.

Therefore, a distance-based classifier benefits from observing more samples which leads

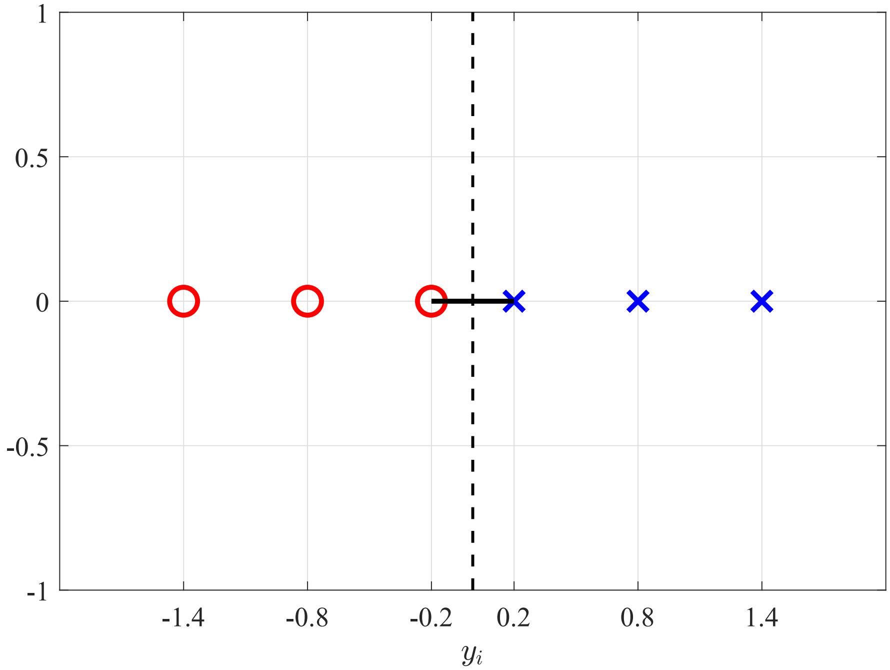

Figure 4.2: The class samples of Example 1.

to distance expansion between different classes [52]. Similarly, if we increase the number of observations to 3, i.e., considering y_{i-1}, y_i, and y_{i+1}, for the detection process of a_i, the distance d becomes $d = \sqrt{(0.2+0.2)^2 + (0.2+0.2)^2 + (0.2+0.2)^2} = 0.69$, which is larger than the distances in Examples 1 and 2. That said, we extend this idea by observing N_p samples centered at y_i during detection process of the transmit symbol a_i for FTN signaling detection. This will increase the distance between the classes samples and eventually will improve the detection performance.

4.3.1 Offline Pre-Classification Process

As mentioned earlier, we define N_p as the number of samples we observe for detecting one transmit symbol. This is equivalent to the classification process happening in N_p-dimensional space. Recall that $y_i = \sum_\ell h_\ell a_{\ell-i}$, and hence, to generate the exact values of all the classes samples y_i we need to consider an infinite length of ISI due to FTN signaling. However, this will significantly increase the number of classes samples in a way that some classes

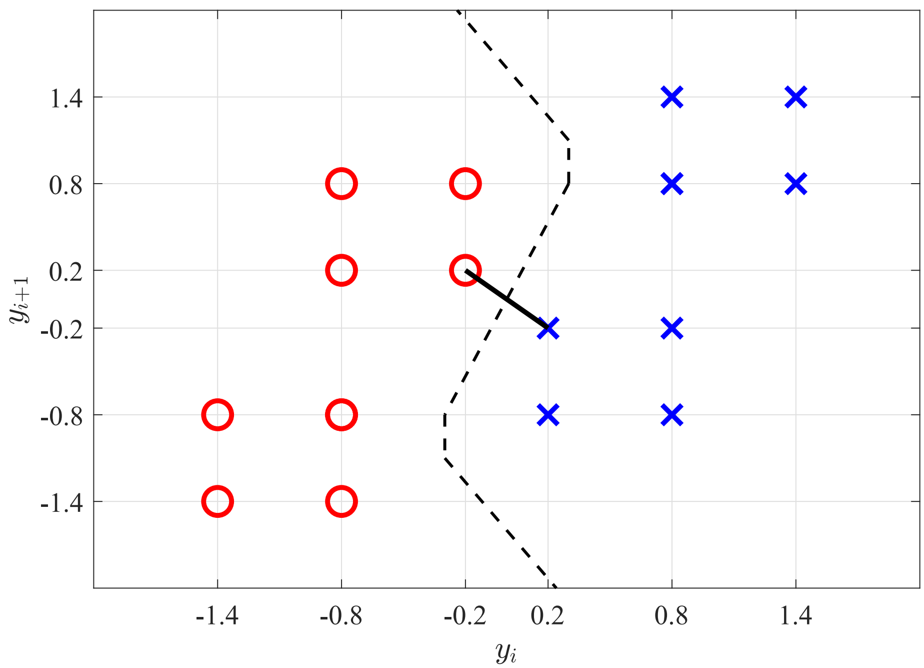

Figure 4.3: The class samples of Example 2.

samples are very close to each other due to the very small values of the tails of the ISI. That said, to reduce the offline pre-classification complexity, we select the dominant coefficients of the ISI N_t. Please note that this does not mean that the actual transmission of the FTN signaling is generated with only N_t ISI coefficients; however, it is generated with the full ISI coefficients. Therefore, to calculate all possible choices of the observation vector with a size of N_p, we have to have all possible $N_p + N_t - 1$ consecutive transmit symbols, i.e. $\boldsymbol{a'_i} = [a_{i-(N_p+N_t-1)/2}, ..., a_{i-1}, a_i, a_{i+1}, ..., a_{i+(N_p+N_t-1)/2}]^{\mathrm{T}}$. Then, we define the set of all possible choices of data symbols $\boldsymbol{a'}$ as $\boldsymbol{\mathcal{M}'} = \{\boldsymbol{m'_1}, \boldsymbol{m'_2}, ..., \boldsymbol{m'_{2^{N_p+N_t-1}}}\}$ where each $\boldsymbol{m'_k}$ is a $(N_p + N_t - 1) \times 1$ vector from one possible choice of $\boldsymbol{a'_i}$. Subsequently, the set of classes samples in this N_p dimensional space is $\boldsymbol{\mathcal{S}'} = \{\boldsymbol{s'_1}, \boldsymbol{s'_2}, ..., \boldsymbol{s'_{2^{N_p+N_t-1}}}\}$, where half of them belong to one class and the half belong to the other class, i.e, $a_i = 1$ or $a_i = -1$.

4.3.2 Online Classification Process

After generating the labeled classes samples in the pre-classification process and given the received vector $\boldsymbol{y}$, we pick the N_p samples centered around the ith sample, i.e., the unlabeled observation class sample $\boldsymbol{o^{(i)}} = [y_{i-N_p/2}, ..., y_{i-1}, y_i, y_{i+1}, ..., y_{i+N_p/2}]^{\mathrm{T}}$, to detect the ith transmit symbol a_i. In the presence of noise, the unlabeled observation class sample $\boldsymbol{o^{(i)}}$ is nothing but an element in the set $\boldsymbol{\mathcal{S}'}$ that is perturbed by noise. Hence, the LCC is defined as the function $f(.)$ such that:

$$y_i \rightarrow \boldsymbol{o^{(i)}} \xrightarrow{f(.)} \mathcal{C}_j, \quad j \in \{-1, 1\}, \tag{4.12}$$

where $\mathcal{C}_{-1} = \{\boldsymbol{m'_k} | a_i = -1\}$ and $\mathcal{C}_1 = \{\boldsymbol{m'_k} | a_i = 1\}$ are the two partitions representing the classes that the ith transmit symbol is -1 or 1, respectively.

4.3.3 Modified Soft Output

In (4.9), we calculate the soft output for each bit x_i based on likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$ because the detection process happened once based on receiving the vector $\boldsymbol{y}$. However, in the proposed LCC, the detection process happens separately for each transmit symbol. Therefore, the likelihood probability changes from $p(\boldsymbol{y}|\boldsymbol{x})$ to $p(\boldsymbol{o^{(i)}}|\boldsymbol{x})$ for the ith symbol. Please note that $p(\boldsymbol{y}|\boldsymbol{x})$ in (4.11) is based on the Euclidean distance and the $p(\boldsymbol{o^{(i)}}|\boldsymbol{x})$ is nothing but projecting the N-dimensional $\boldsymbol{y}$ into the N_p-dimensional vector $\boldsymbol{o^{(i)}}$. Since $N \gg N_p$ this replacement comes with an error when compared to the exact value of the LLR based on (4.9). To quantify this error, we re-write (4.5) as:

$$\begin{aligned} y_i = \sum_{l<i-N_p/2} a_{i-l} h_l + \sum_{i-N_p/2 \leq l \leq i+N_p/2} a_{i-l} h_l \\ + \sum_{l>i+N_p/2} a_{i-l} h_l + w_i, \end{aligned} \tag{4.13}$$

where the second term of the right hand side is exactly the ith element of vector $\boldsymbol{o^{(i)}}$, i.e. $o_i^{(i)} = \sum_{i-N_p/2 \leq l \leq i+N_p/2} a_{i-l} h_l$. Then, the error ϵ is defined as:

$$\epsilon = \sum_{l<i-N_p/2} a_{i-l} h_l + \sum_{l>i+N_p/2} a_{i-l} h_l. \tag{4.14}$$

Please note that, since the tails of $h(t)$ has very small values, ϵ is also small. Similarly, the other elements of $\boldsymbol{o^{(i)}}$ is calculated, and we can approximate (4.9) by replacing $p(\boldsymbol{o^{(i)}}|\boldsymbol{x})$ to

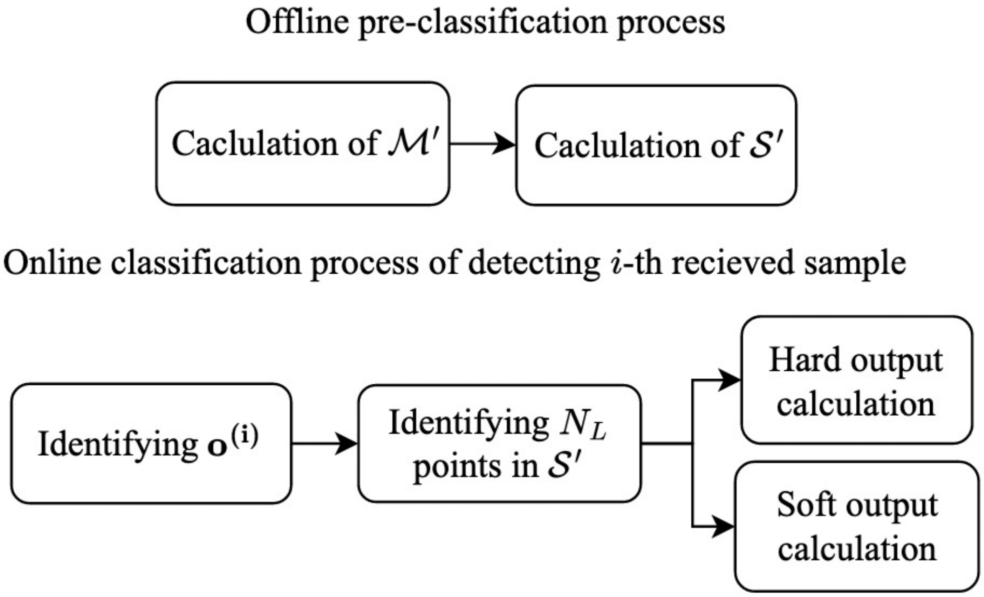
Offline pre-classification process
Caclulation of $\mathcal{M}'$
Caclulation of $\mathcal{S}'$
Online classification process of detecting i-th recieved sample
Identifying $\mathbf{o}^{(\mathbf{i})}$
Identifying N_L points in $\mathcal{S}'$
Hard output calculation
Soft output calculation

Figure 4.4: Illustration of the proposed LCC.

$p(\boldsymbol{y}|\boldsymbol{x})$. Therefor, the approximate LLR value for i-th symbol is as:

$$\begin{split}&\tilde{L}_D\left(x_i \mid \boldsymbol{o}^{(i)}\right) = \\ &L_A\left(x_i\right) + \ln \frac{\sum_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}'}_{i,1}} p(\boldsymbol{o}^{(i)} \mid \boldsymbol{x}) \cdot \exp \sum_{j\in\mathcal{J}_{i,\boldsymbol{x}}} L_A\left(x_j\right)}{\sum_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}'}_{i,0}} p(\boldsymbol{o}^{(i)} \mid \boldsymbol{x}) \cdot \exp \sum_{j\in\mathcal{J}_{i,\boldsymbol{x}}} L_A\left(x_j\right)},\end{split} \tag{4.15}$$

where $\boldsymbol{\mathcal{X}'} = \text{map}(\boldsymbol{\mathcal{M}'})$. Further reduction in the computational complexity comes from reducing the search space in the lattice $\boldsymbol{\mathcal{X}'}$ where we only consider a pre-defined N_L number of closest lattice points, $\boldsymbol{\mathcal{L}}$, to the vector $\boldsymbol{o}^{(i)}$ and exclude the rest from $\boldsymbol{\mathcal{X}'}$. The results of the Abbasi et al. [51] showed that such approximated LLR values is very close to the exact values of LLR. The proposed LCC is depicted visually in Fig. 4.4.

4.4 Computational Complexity Analysis

As can be seen in Fig. 4.4, the first stage of the online process of the proposed LCC to detect the ith sample is to identify $\boldsymbol{o}^{(i)}$ samples which have constant computational complexity.

The second step requires identifying a radius that includes N_L closest points to $\boldsymbol{o}^{(i)}$ from a pre-trained offline DL-LSD which has a negligible online complexity. To calculate the hard outputs in the third step, one requires a complexity of $O(N_L N_p)$; this is as we calculate the minimum distance from $\boldsymbol{o}^{(i)}$ to N_L points in N_p-dimension space. To calculate the soft outputs in the third step, one requires a complexity of $O(N_L^2 N_p)$. That said, the total online computational complexity of the proposed LCC to detect N samples is $O(N N_L N_p)$ and $O(N N_L^2 N_p)$ for the hard and soft outputs, respectively.

4.5 Simulation Results

In this section, we evaluate the performance of the proposed LCC algorithm in detecting BPSK FTN signaling. We consider the pulse shapes $h(t)$ and $v(t)$ to be root raised cosine with roll-off factors 0.35 and 0.12, respectively. We set $N = 1000$ data symbols per transmission block, and we employ a standard convolutional code (7, [171 133]) at the transmitter side and a Viterbi decoder to decode the approximated soft outputs of the proposed LCC at the receiver. Following [51], we set $N_L = 8$ as there is negligible, i.e. 0.2 dB, performance degradation when compared to the case of $N_L = 128$. For the classification task, we use the distance-based K-nearest neighbor (KNN) classifier, with $K = 1$.

The implementation steps of the proposed algorithm are as follows: As a first step, using the selected N_p we generate $\boldsymbol{\mathcal{S}'}$ based on Subsection 4.3.1. Then, we fit the KNN by $\boldsymbol{\mathcal{S}'}$ as a training set. After saving the KNN model, from the testing phase, we use $\boldsymbol{o^{(i)}}$ for each transmit block as discussed in detail in Subsection 4.3.2 for the KNN model to classify. After classification, we implemented the (4.15) to calculate the soft output. All the implementation has been done in a Python3 environment, and we used the KNN classifier from the sci-kit-learn library [53].

It was demonstrated earlier through Examples 1 and 2 that increasing the number of observations N_p eventually increases the distance between the classes samples. To select a proper value of N_p that strikes a balance between performance and complexity, Fig. 4.5 plots

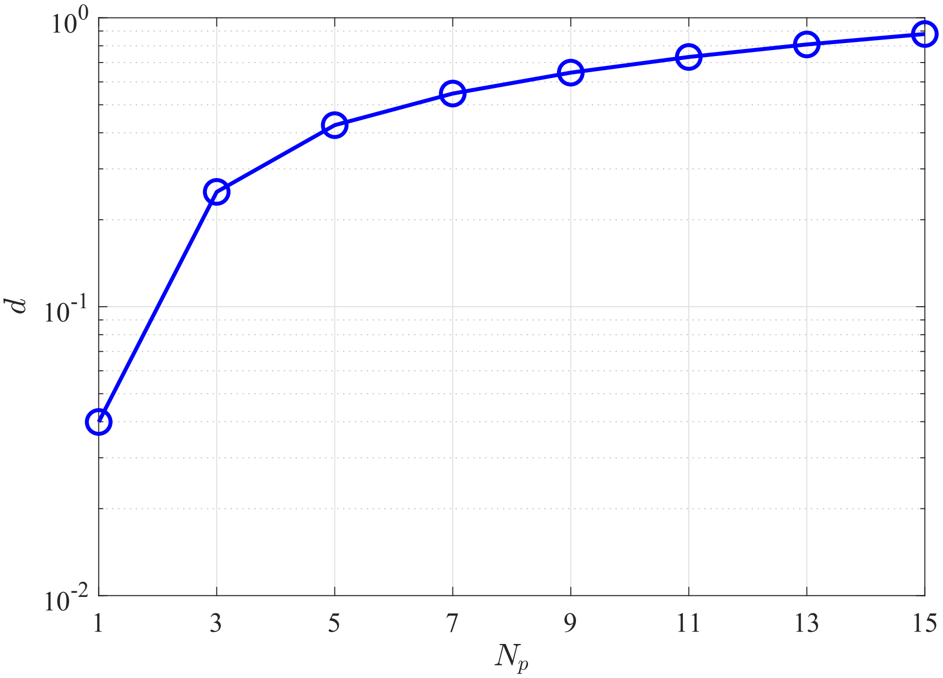

Figure 4.5: The distance d between the classes samples as a function of N_p.

the closest distance d between the two different classes samples as a function of N_p. As can be seen, increasing the value of N_p initially increases the distance between the classes samples, and hence, improves the detection performance; however, such improvement is reduced at high values of N_p. Please note that at high values of N_p, the proposed LCC will suffer from the curse of dimensionality. That said, we choose the value of N_p to be 13 or 15 through the rest of the simulations.

To study the effect of N_t on the BER performance, in Fig. 4.6 we plot the BER of the proposed LCC at $N_t = 3$ and 5 and the DL-LSD [51] at $\tau = 0.6$ and $N_p = 13$. As one can see, selecting $N_t = 3$ will significantly deteriorate the BER performance. However, increasing the value of N_t to 5 results in a BER performance that is close to the optimal BER obtained from the DL-LSD.

In Fig. 4.7, we plot the BER performance at $\tau = 0.5$ (and $N_t = 7$) and 0.6 (and $N_t = 5$) for the proposed LCC at $N_p = 13$ and 15 and the DL-LSD [51]. At $\tau = 0.5$ and BER of 3×10^{-4}, the difference in E_b/N_o when $N_p = 15$ and $N_p = 13$ and the optimal performance of the DL-LSD is around 0.8 dB and 1 dB, respectively. At $\tau = 0.6$ and BER of 10^{-4}, the

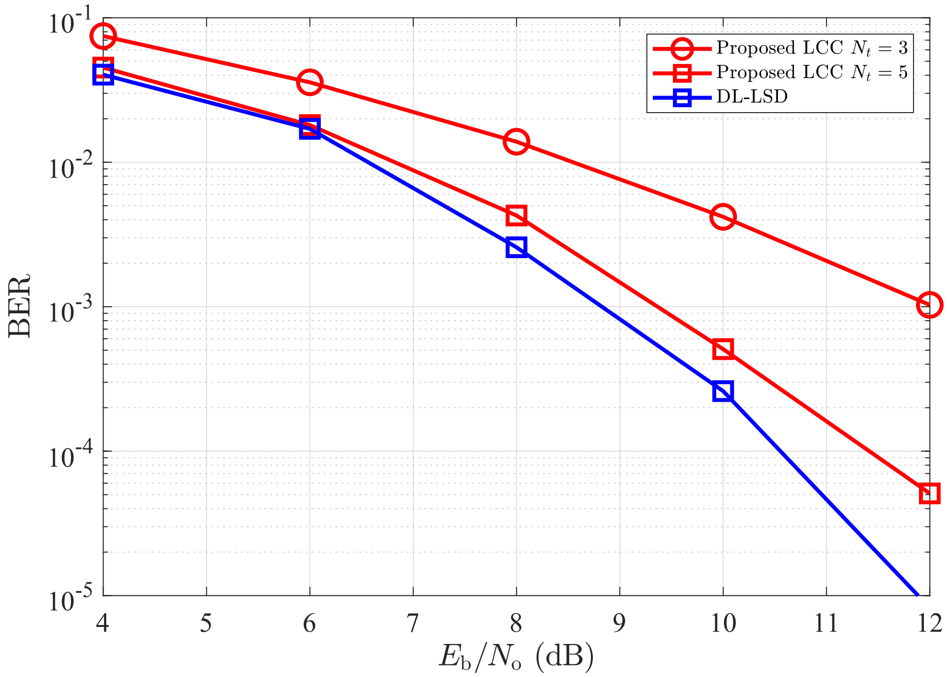

Figure 4.6: BER performance of the proposed LCC algorithm at $N_t = 3$ and 5, and the DL-LSD algorithm.

penalty in E_b/N_p reduces to 0.4 dB and 0.8 dB, respectively.

Fig. 4.8 depicts the coded BER performance of the proposed LCC at $N_p = 13$ and the DL-LSD [51] at $\tau = 0.6$. As can be seen, at BER of 10^{-4} there is about 0.5 dB between the proposed LCC algorithm with $N_p = 13$ and the DL-LSD with $N_p = N$; however, at the cost of huge reduction in the computational complexity.

4.6 Conclusion

FTN signaling is a promising technique in future communication systems since it improves the SE without changing the transmission bandwidth. In this chapter, we proposed the LCC algorithm that exploits the ISI structure of FTN signaling to perform the classification task in $N_p \ll N$-dimension space. The proposed LCC algorithm reduced the computational complexity in both coded and uncoded scenarios by removing the exponential part related to N and replacing it with a small number, i.e., $N_p + N_t - 1$. However, such an improvement

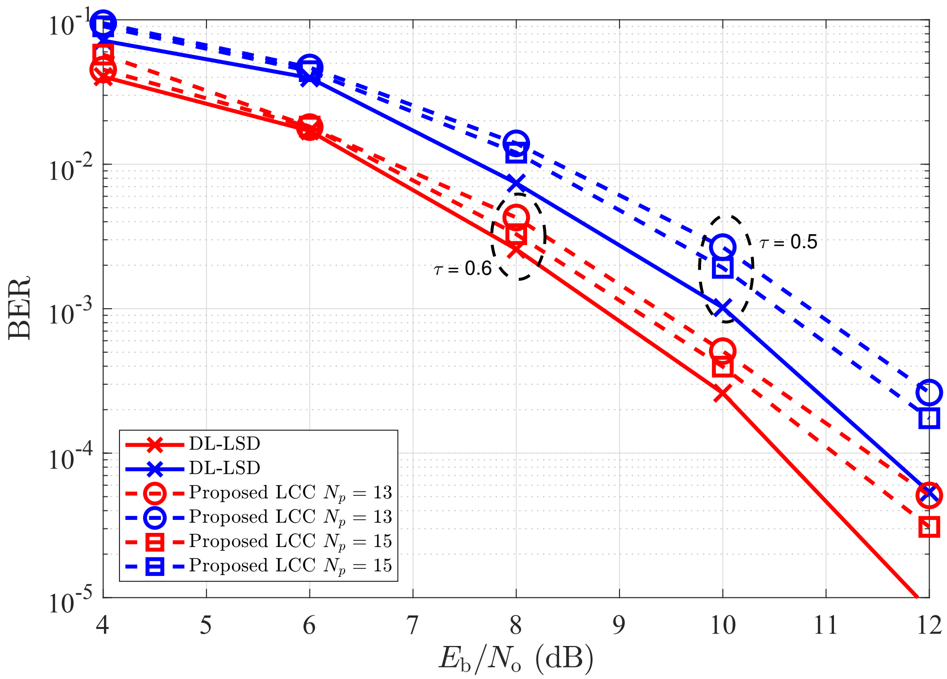

Figure 4.7: BER performance of the proposed LCC algorithm (at $N_p = 13$ and 15) and the DL-LSD (at $N_p = N$) [51] at $\tau = 0.6$ and 0.5.

comes with degradation in BER performance where for example simulation results showed that at $\tau = 0.6$ and BER of 10^{-4} there is 0.4 dB penalty in comparison to the optimal solution.

4.7 References

[13] E. Bedeer, H. Yanikomeroglu, and M. H. Ahmed, “Reduced complexity optimal detection of binary faster-than-Nyquist signaling,” in *Proceeding of the IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6 (cit. on pp. 12, 18, 26, 37, 47).

[18] J. E. Mazo, “Faster-than-Nyquist signaling,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1451–1462, Oct. 1975 (cit. on pp. 17, 26, 47).

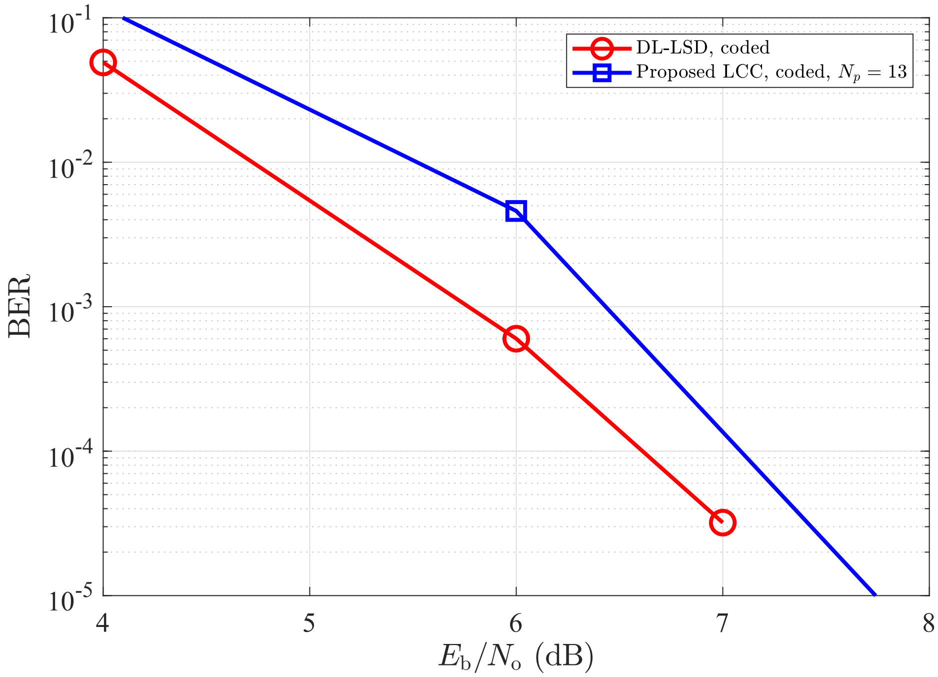

Figure 4.8: BER performance of the proposed LCC at $N_p = 13$ and the DL-LSD [51] at $\tau = 0.6$.

[22] J. B. Anderson, A. Prlja, and F. Rusek, "New reduced state space BCJR algorithms for the ISI channel," in *Proceeding of the IEEE International Symposium on Information Theory*, Jun. 2009, pp. 889–893 (cit. on pp. 18, 47).

[24] E. Bedeer, M. H. Ahmed, and H. Yanikomeroglu, "A very low complexity successive symbol-by-symbol sequence estimator for faster-than-Nyquist signaling," *IEEE Access*, vol. 5, pp. 7414–7422, Mar. 2017 (cit. on pp. 18, 26, 47).

[40] P. Song, F. Gong, Q. Li, G. Li, and H. Ding, "Receiver design for faster-than-Nyquist signaling: Deep-learning-based architectures," *IEEE Access*, vol. 8, pp. 68 866–68 873, Apr. 2020 (cit. on pp. 26, 47).

[43] J. B. Anderson, *Coded Modulation Systems*. Kluwer Academic Publishers, 2002 (cit. on pp. 28, 40, 48, 49).

[49] A. D. Liveris and C. N. Georghiades, "Exploiting faster-than-Nyquist signaling," *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1502–1511, Sep. 2003 (cit. on p. 47).

[50] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Oct. 2017 (cit. on p. 47).

[51] S. Abbasi and E. Bedeer, "Deep learning-based list sphere decoding for Faster-than-Nyquist (FTN) signaling detection," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)-Spring*, Jun. 2022 (cit. on pp. 47–49, 51, 56–61).

[52] S. Theodoridis, C. Cowan, C. Callender, and C. See, "Schemes for equalisation of communication channels with nonlinear impairments," *IEE Proceedings Communications*, vol. 142, no. 3, pp. 165–171, 1995 (cit. on p. 53).

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011 (cit. on p. 57).

5 Summary and Future Work

5.1 Summary

Improving the SE emerge as an important factor for future wireless communication systems. FTN signaling is a promising nominee to address this goal. However, FTN signaling consumes a lot of computational resources; hence, we aimed to come up with new algorithms to address this issue. In this thesis we proposed two different DL/ML based algorithms to decrease the high computational complexity regarding the detection process of FTN signaling.

In the first contribution, we looked at the well-known LSD algorithm for the detection of the received samples in FTN signaling. In fact, the computational complexity of LSD is directly correlated with its initial radius. To address the computational complexity issue of using LSD algorithms, we proposed a DL-LSD algorithm that learns and update an approximate initial radius to include a pre-defined number of points N_L inside of hypersphere. Essentially, the DL-LSD predicts a initial radius and searches the hypersphere to find points within this radius. If total founded points are less than N_L simply we increase the initial radius by the standard deviation of the distribution of predicted radius in training phase, and we proceed the search over hypersphere aging with new initial radius. Simulation results showed that number of flops of the proposed DL-LSD algorithm is three order and one order of magnitude lower than original LSD at low and high values of E_b/N_o, respectively.

In the second contribution, we investigated the complexity issue when we have long transmission blocks, i.e. $N \gg 1000$. Basically, even the proposed DL-LSD in first contribution can be computationally complex when we operate the search over N-dimensional hypersphere. The proposed LCC algorithm exploits the ISI structure of FTN signaling to perform the detection task in $N_p \ll N$-dimension space. The proposed LCC algorithm reduced the computational complexity in both coded and uncoded scenarios by removing the exponential part related to N and replacing it with a small number, i.e., $N_p + N_t - 1$. Beside, by using the DL-LSD in coded scenario we reduced the computational complexity further to $O(N_p^2 N_L)$ to even remove the exponential part related to N_p. However, such improvements come with a slight degradation in the BER performance where for example simulation results showed that at $\tau = 0.6$ and BER of 10^{-4} there is 0.4 dB penalty in comparison to the optimal solution.

5.2 Suggestion for Future Studies

- All studies presented in this thesis are conducted in AWGN channel. As a further development of the first contribution, one can extend it into the time-variant channel situation as well. In this regard, the basic idea will be feeding not only the received vector but the channel matrix information as input to the neural network. Moreover, since the training process will be huge in this case, and the time-variant channels can vary quickly, one can employ the incremental learning techniques, i.e., online learning, instead of traditional learning techniques. In incremental learning, the model effectively can adopt in new changes in the recent stream of data, which is more likely to occur in practical situations.

- Another aspect of extending the first contribution is considering the situation where the value of τ is unknown. One can use the mean and standard deviation calculated for radii during training phase of different τ, e.g., $\tau = 0.3,\ 0.5,\ 0.7$. Then, in testing mode, received signal will be given to each trained model of τ. Simply by comparing the calculated radii of each model of τ with calculated mean value of radii from training process we can estimate the range of operating τ.

- As a further development of the second contribution, the calculated schema in training process can be shrunk by a small trade-off with bit error performance. In this scenario, it is valuable to find this trade-off threshold because we may be able decrease the complexity in half but losing performance in small amount. As an example to make this point clear, one can look at Fig. 4.3 on Chapter 4, where we can remove four each red and blue points that are far from boundary line while maintaining same boundary line and not losing performance and decreasing half of the computational complexity. In general, in higher dimensional space this might come with slight loss in performance, which can also be studied as a new factor for balancing performance and complexity.

6 List of Publications

- **Accepted:**

  S. Abbasi and E. Bedeer, “Deep learning-based list sphere decoding for Faster-than-Nyquist (FTN) signaling detection,” in Proceeding of the IEEE Vehicular Technology Conference (VTC)-Spring, Jun. 2022.

- **Under review:**

  S. Abbasi and E. Bedeer, “Low Complexity Classification Approach for Faster-than-Nyquist (FTN) Signalling Detection”, Under review, IEEE Communications Letters.

7 Bibliography

[1] "Propagation modeling for wireless communications in the terahertz band," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 96–101, Jun. 2018 (cit. on p. 2).

[2] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Apr. 2020 (cit. on p. 3).

[3] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020 (cit. on p. 3).

[4] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," in *Proceedings of the IEEE Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct. 2017, pp. 76–80 (cit. on p. 3).

[5] Z. Zhao, M. C. Vuran, F. Guo, and S. D. Scott, "Deep-waveform: A learned OFDM receiver based on deep complex-valued convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2407–2420, Jun. 2021 (cit. on p. 3).

[6] M. Honkala, D. Korpi, and J. M. Huttunen, "Deeprx: Fully convolutional deep learning receiver," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3925–3940, Feb. 2021 (cit. on p. 3).

[7] Y. Wang, M. Martonosi, and L.-S. Peh, "A supervised learning approach for routing optimizations in wireless sensor networks," in *Proceedings of the IEEE*, May 2006, pp. 79–86 (cit. on p. 4).

[8] H. Agirman-Tosun, Y. Liu, A. M. Haimovich, *et al.*, “Modulation classification of MIMO-OFDM signals by independent component analysis and support vector machines,” in *Proceedings of the IEEE*, Nov. 2011, pp. 1903–1907 (cit. on p. 4).

[9] G. De Veciana and A. Zakhor, “Neural net-based continuous phase modulation receivers,” *IEEE Transactions of Communication*, vol. 40, no. 8, pp. 1396–1408, Aug. 1992 (cit. on p. 4).

[10] E. A. Lee and D. G. Messerschmitt, “Digital communication,” *Springer Science Business Media*, 2012 (cit. on p. 10).

[11] R. Mancini, “Op amps for everyone: Design reference,” *Newnes*, 2003 (cit. on p. 10).

[12] A. Goldsmith, “Wireless communications,” *Cambridge University Press*, 2005 (cit. on p. 10).

[13] E. Bedeer, H. Yanikomeroglu, and M. H. Ahmed, “Reduced complexity optimal detection of binary faster-than-Nyquist signaling,” in *Proceeding of the IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6 (cit. on pp. 12, 18, 26, 37, 47).

[14] E. Bedeer, M. H. Ahmed, and H. Yanikomeroglu, “Low-complexity detection of high-order QAM faster-than-Nyquist signaling,” *IEEE Access*, vol. 5, pp. 14 579–14 588, Jul. 2017 (cit. on pp. 13, 18).

[15] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of computation*, vol. 44, no. 170, pp. 463–471, 1985 (cit. on p. 13).

[16] B. Hassibi and H. Vikalo, “On the sphere-decoding algorithm I. Expected complexity,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Jul. 2005 (cit. on p. 14).

[17] J. B. Anderson, F. Rusek, and V. Öwall, “Faster-than-Nyquist signaling,” *Proceedings of the IEEE*, vol. 101, no. 8, pp. 1817–1830, Mar. 2013 (cit. on pp. 16, 19).

[18] J. E. Mazo, “Faster-than-Nyquist signaling,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1451–1462, Oct. 1975 (cit. on pp. 17, 26, 47).

[19] C.-K. Wang and L.-S. Lee, "Practically realizable digital transmission significantly below the Nyquist bandwidth," *IEEE Transactions on Communications*, vol. 43, no. 2, pp. 166–169, Dec. 1995 (cit. on p. 17).

[20] N. Seshadri, "Error performance of trellis modulation codes on channels with severe intersymbol interference," Ph.D. dissertation, Rensselaer Polytechnic Institute, Sep. 1986 (cit. on p. 17).

[21] A. D. Liveris and C. N. Georghiades, "Exploiting faster-than-nyquist signaling," *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1502–1511, Sep. 2003 (cit. on p. 17).

[22] J. B. Anderson, A. Prlja, and F. Rusek, "New reduced state space BCJR algorithms for the ISI channel," in *Proceeding of the IEEE International Symposium on Information Theory*, Jun. 2009, pp. 889–893 (cit. on pp. 18, 47).

[23] S. Li, B. Bai, J. Zhou, P. Chen, and Z. Yu, "Reduced-complexity equalization for faster-than-Nyquist signaling: New methods based on Ungerboeck observation model," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1190–1204, Nov. 2017 (cit. on pp. 18, 28).

[24] E. Bedeer, M. H. Ahmed, and H. Yanikomeroglu, "A very low complexity successive symbol-by-symbol sequence estimator for faster-than-Nyquist signaling," *IEEE Access*, vol. 5, pp. 7414–7422, Mar. 2017 (cit. on pp. 18, 26, 47).

[25] M. Kulhandjian, E. Bedeer, H. Kulhandjian, C. D'Amours, and H. Yanikomeroglu, "Low-complexity detection for faster-than-Nyquist signaling based on probabilistic data association," *IEEE Communications Letters*, vol. 24, no. 4, pp. 762–766, Dec. 2019 (cit. on pp. 18, 26).

[26] E. Bedeer, H. Yanikomeroglu, and M. H. Ahmed, "Low-complexity detection of M-ary PSK faster-than-Nyquist signaling," in *Proceeding of the IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, Apr. 2019, pp. 1–5 (cit. on p. 18).

[27] A. Ibrahim, E. Bedeer, and H. Yanikomeroglu, “A novel low complexity faster-than-Nyquist (FTN) signaling detector for ultra high-order QAM,” *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2566–2580, Nov. 2021 (cit. on pp. 18, 37).

[28] J. Fan, Y. Ren, Y. Zhang, and X. Luo, “MLSE equalizer with channel shortening for faster-than-Nyquist signaling,” *IEEE Photonics Technology Letters*, vol. 30, no. 9, pp. 793–796, Mar. 2018 (cit. on p. 19).

[29] S. Li, J. Yuan, B. Bai, and N. Benvenuto, “Code-based channel shortening for faster-than-Nyquist signaling: Reduced-complexity detection and code design,” *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 3996–4011, Apr. 2020 (cit. on p. 19).

[30] X. Wen, W. Yuan, D. Yang, N. Wu, and J. Kuang, “Low complexity message passing receiver for faster-than-Nyquist signaling in nonlinear channels,” *IEEE Access*, vol. 6, pp. 68 233–68 241, Oct. 2018 (cit. on p. 19).

[31] S. Li, N. Wu, Q. Shi, and Q. Guo, “FTN signaling-aided space-time multi-mode index modulation systems with a GMP-based receiver,” *IEEE Access*, vol. 7, pp. 162 898–162 912, Nov. 2019 (cit. on p. 19).

[32] W. Yuan, N. Wu, A. Zhang, X. Huang, Y. Li, and L. Hanzo, “Iterative receiver design for FTN signaling aided sparse code multiple access,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 915–928, Nov. 2019 (cit. on p. 19).

[33] K. Wang, A. Liu, X. Liang, S. Peng, and Q. Zhang, “A faster-than-Nyquist (FTN)-based multicarrier system,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 947–951, Aug. 2018 (cit. on p. 19).

[34] T. Ishihara, S. Sugiura, and L. Hanzo, “The evolution of faster-than-Nyquist signaling,” *IEEE Access*, vol. 9, pp. 86 535–86 564, Jun. 2021 (cit. on pp. 19, 26).

[35] P. Song, F. Gong, Q. Li, G. Li, and H. Ding, “Receiver design for faster-than-Nyquist signaling: Deep-learning-based architectures,” *IEEE Access*, vol. 8, pp. 68 866–68 873, Apr. 2020 (cit. on p. 20).

[36] B. Liu, S. Li, Y. Xie, and J. Yuan, "A novel sum-product detection algorithm for faster-than-Nyquist signaling: A deep learning approach," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5975–5987, Jun. 2021 (cit. on pp. 20, 26).

[37] J. B. Anderson, F. Rusek, and V. Öwall, "Faster-than-Nyquist signaling," *Proceedings of the IEEE*, vol. 101, no. 8, pp. 1817–1830, Mar. 2013 (cit. on p. 26).

[38] F. Rusek and J. B. Anderson, "Non binary and precoded faster than Nyquist signaling," *IEEE Transaction on Communications*, vol. 56, no. 5, pp. 808–817, May 2008 (cit. on p. 26).

[39] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, Dec. 2018 (cit. on p. 26).

[40] P. Song, F. Gong, Q. Li, G. Li, and H. Ding, "Receiver design for faster-than-Nyquist signaling: Deep-learning-based architectures," *IEEE Access*, vol. 8, pp. 68 866–68 873, Apr. 2020 (cit. on pp. 26, 47).

[41] A. Prlja, J. B. Anderson, and F. Rusek, "Receivers for faster-than-Nyquist signaling with and without turbo equalization," in *Proceedings of the IEEE International Symposium on Information Theory*, Jul. 2008, pp. 464–468 (cit. on p. 28).

[42] A. Prlja and J. B. Anderson, "Reduced-complexity receivers for strongly narrowband intersymbol interference introduced by faster-than-Nyquist signaling," *IEEE Transactions on Communications*, vol. 60, no. 9, pp. 2591–2601, Sep. 2012 (cit. on p. 28).

[43] J. B. Anderson, *Coded Modulation Systems*. Kluwer Academic Publishers, 2002 (cit. on pp. 28, 40, 48, 49).

[44] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, Apr. 2003 (cit. on pp. 31, 32).

[45] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4368–4378, Sep. 2019 (cit. on p. 33).

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, Nov. 2016 (cit. on p. 33).

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, Dec. 2014 (cit. on p. 34).

[48] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Jul. 2005 (cit. on pp. 36–38).

[49] A. D. Liveris and C. N. Georghiades, "Exploiting faster-than-Nyquist signaling," *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1502–1511, Sep. 2003 (cit. on p. 47).

[50] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Oct. 2017 (cit. on p. 47).

[51] S. Abbasi and E. Bedeer, "Deep learning-based list sphere decoding for Faster-than-Nyquist (FTN) signaling detection," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)-Spring*, Jun. 2022 (cit. on pp. 47–49, 51, 56–61).

[52] S. Theodoridis, C. Cowan, C. Callender, and C. See, "Schemes for equalisation of communication channels with nonlinear impairments," *IEE Proceedings Communications*, vol. 142, no. 3, pp. 165–171, 1995 (cit. on p. 53).

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011 (cit. on p. 57).