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

Sina Abbasi, Member, IEEE, and Ebrahim Bedeer, Member, IEEE

Abstract—In this letter, we investigate the use of machine learning (ML) to reduce the detection complexity of faster-than-Nyquist (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 the set of all possible classes samples. We observe that by jointly considering \( N_p \) samples, where \( N_p \ll N \) and \( N \) is the transmission block length, for the FTN signaling detection, the distance between the classes samples of any distance-based classifier increases, and hence, the detection performance improves. That said, we propose a low-complexity classifier (LCC) that exploits the ISI structure of FTN signaling to perform the classification task in \( N_p \)-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.

Index Terms—Classification, faster-than-Nyquist signaling, intersymbol interference, machine learning.

I. INTRODUCTION

Improving the spectral efficiency (SE) is one of the main goals of next generation communication systems. Faster-than-Nyquist (FTN) signaling [1] 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 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. Several algorithms, based on concepts from traditional detection theory, signal processing, or optimization theory have been proposed recently to reduce the detection complexity of FTN signaling, see, e.g., [2], [3], [4], [5] and references therein.

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. This is as ML algorithms can potentially reduce the computational complexity in comparison to traditional algorithms by shifting a major part of the complexity to the offline phase while providing a robust low-complexity online phase. The authors in [6] proposed two different deep learning (DL)-based architectures for FTN signaling receivers (one for detection while the other for both the detection and decoding) that can achieve near-optimal performance in non-severe ISI regions. In [7], the authors proposed a DL-based algorithm to approximate the initial radius of the list sphere decoding (LSD) algorithm to detect FTN signaling. The proposed DL-LSD considerably reduces the detection complexity of its online phase when compared to the LSD algorithm; however, the data generation for the training process, i.e., the offline phase, is of high computational complexity for large transmission block lengths. The authors in [8] proposed a DL-based sum-product algorithm for FTN signaling detection 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.

Against the aforementioned literature, 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 the 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 observe that jointly considering \( N_p \ll N \) samples for FTN signaling detection increases the distance between the classes samples of any distance-based classifier, and eventually improves the detection performance. A further increase in the considered number of samples beyond \( N_p \) does not lead to tangible improvements of the distance between the classes samples. That said, 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.

The rest of the letter is organized as follows. In Section II, we present the system model of FTN signaling. In Section III, we discuss the proposed LCC, and its computational
complexity analysis is introduced in Section IV. Simulation results are presented in Section V, and in Section VI we conclude the letter.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. FTN Signaling Model

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

$$s(t) = \sum a_i h(t - i\tau T),$$

where $0 < \tau \leq 1$ is the time acceleration parameter, and $\tau T$ is the symbol duration. In this letter, we adopt an equivalent FTN signaling model based on the orthonormal basis functions [9]. In the equivalent FTN signaling model, the $T$-orthogonal pulse $h(t)$ is approximated by the sum of multiple $\tau T$-orthonormal pulses $h(t) \approx \sum h_i v(t - i\tau T)$, where $h_i$ is a sampled block of $h(t)$ at $\tau T$ and it is given as $h_i = \sqrt{\tau T}h(i\tau T)$ [9]. That said, an equivalent expression for the transmit FTN signal can be written as:

$$s(t) = \sum b_i v(t - i\tau T),$$

where $b_i = \sum_i a_i h_i$. For a root-raised cosine pulse $h(t)$, its roll-off factor $\beta_h$ must satisfy $\tau < 1/(1 + \beta_h)$ for the equivalent model in (2) to hold [9]. The received signal after passing through a filter matched to $v(t)$ and sampling at every $\tau T$ is given as $y_i = b_i + w_i$, where $w_i$ is the sampled white Gaussian noise with zero mean and $\sigma^2$ variance. Please note that the noise samples after the matched filter are uncorrelated as we have an orthogonal transmission that carries a larger alphabet of real-valued symbols, i.e., $b_i = \sum_i a_i h_i$, instead of conventional $M$-ary $a_i$ symbols. The received signal can be written in a matrix form as:

$$y = H a + w,$$

where the $j$th row, $0 \leq j \leq N - 1$, of the matrix $H$ is a vector given as $[h_{-j}, \ldots, h_{-2}, h_{-1}, h_0, h_1, h_2, \ldots, h_{N-j-1}]$. Please note that the elements of $H$ are scaled versions of the samples of $h(t)$ as discussed earlier, i.e., $h_i = \sqrt{\tau T}h(i\tau T)$.

B. FTN Detection as a Classification Problem

One of ML's main types of tasks is 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. The goal of any classification algorithm is to assign the proper label for the unlabeled data.

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

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

$$y \xrightarrow{f(.)} s_k = g(m_k), \quad k \in \{1, 2, \ldots, 2^N\}. \quad (4)$$

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

III. PROPOSED LOW COMPLEXITY CLASSIFICATION OF FTN SIGNALING

As discussed earlier, the number of classes in the conventional classification problem to detect FTN signaling is $2^N$ and, hence, one of the biggest hindrances for such a conventional classification approach is the huge computational complexity, especially for long transmit block of data symbols. In the following, we propose a LCC that exploits the inherent structure of ISI to reduce the classification computational complexity of binary phase shift keying (BPSK) FTN signaling. Please note that the proposed LCC can be extended straightforwardly to high-order modulation; however, we opted to use BPSK modulation to facilitate the explanation of the main idea of the proposed LCC. To show the intuition behind the proposed LCC, we provide the following examples.

Example 1: Let us assume a noise free transmission, i.e., $y_i = b_i$, of $N$ transmit symbols, where each symbol is affected only by ISI from one past and one upcoming symbol, i.e., $y_i = b_i = \sum_{\ell=-1}^1 a_{\ell}h_\ell$ 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. 1, where the cross and circle points represent the values of $y_i$ corresponding to $a_1 = 1$ and $a_i = -1$, respectively. If we consider the classification objective to be the nearest distance, then the dashed line in Fig. 1 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 $y$, 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. 2, 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 to distance expansion between different classes. 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. This will increase the distance between the classes samples and eventually will improve the detection performance (as will be seen later in the simulation results); however, at the cost of an increase of the computational complexity (as will be discussed later in the following subsections).

Please note that jointly considering more samples to improve the classification distance, and hence, the BER performance is similar to the idea of a reduced-search trellis-based detection where the states are defined by consecutive symbols with non-negligible interference [3]. It is also worthy to emphasize that jointly considering $N_p$ samples is different than channel shortening techniques where one aims to find a target impulse response to minimize a certain design criterion.

**A. 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 $y_i = \sum_{\ell} h_{\ell} a_{\ell-i} + w_i$, and to generate the exact values of all the classes samples $y_i$, we assume noise-free transmission and 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 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. $a_i^T = [a_i-(N_t+N_t-1)/2, \ldots, a_i-1, a_i, a_i+1, \ldots, a_i+(N_p+N_t-1)/2]^T$.

Then, we define the set of all possible choices of data symbols $a'$ as $\mathcal{M}' = \{m'_1, m'_2, \ldots, m'_{2N_p+N_t-1}\}$ where each $m'_k$ is a $(N_p + N_t - 1) \times 1$ vector from one possible choice of $a_i^T$. Subsequently, the set of classes samples in this $N_p$-dimensional space is $\mathcal{S}' = \{s'_1, s'_2, \ldots, s'_{{2N_p+N_t-1}}\}$, where half of them belongs to one class and the other half belongs to the other class, i.e., $a_i = 1$ or $a_i = -1$.

**B. Online Classification Process**

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

$$y_i \rightarrow o^{(i)} \xrightarrow{f(.)} C_j, \quad j \in \{-1, 1\},$$  \hspace{1cm} (5)

where $C_{-1} = \{m'_k | a_i = -1\}$ and $C_1 = \{m'_k | a_i = 1\}$ are the two partitions representing the classes that the $i$th transmit symbol is $-1$ or $1$, respectively.

**C. Modified Soft Output**

We can calculate the soft output for each bit $x_j$ based on likelihood function $p(y|x)$ because the detection process happens based on receiving $y$. However, in the proposed LCC, the detection process is based on receiving $o^{(i)}$. Therefore, the likelihood probability changes from $p(y|x)$
to \( p(o^{(i)}_l | x) \) for the \( i \)-th symbol. Please note that \( p(y|x) \) is based on the Euclidean distance \( \|y - Ha\|^2 \) and the \( p(o^{(i)}_l | x) \) is nothing but projecting the \( N \)-dimensional \( y \) into the \( N_p \)-dimensional vector \( o^{(i)} \). Since \( N \gg N_p \) this replacement comes with an error when compared to the exact value of the LLR. To quantify this error, we re-write \( y_i = b_i + w_i \) as:

\[
y_i = \sum_{l<i-N_p/2} a_{i-l} h_l + \sum_{i-N_p/2 \leq l \leq i+N_p/2} a_{i-l} h_l + \sum_{l>i+N_p/2} a_{i-l} h_l + w_i, \tag{6}
\]

where the second term of the right hand side is exactly the \( i \)-th element of vector \( o^{(i)} \), 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 \( \epsilon \) is defined as:

\[
\epsilon = \sum_{l<i-N_p/2} a_{i-l} h_l + \sum_{l>i+N_p/2} a_{i-l} h_l. \tag{7}
\]

Since the tails of \( h(t) \) have very small values, \( \epsilon \) is also small. Please note that similar error terms are analyzed in [3] and [5]. The approximate LLR value for \( i \)-th symbol can be written as [7]:

\[
\hat{L}_D \left( x_i | o^{(i)} \right) = L_A \left( x_i \right) + \ln \left( \sum_{x \in \mathcal{X}'} p(o^{(i)} | x) \sum_{x \in \mathcal{X}'} p(o^{(i)} | x) \exp \sum_{j \in \mathcal{J}_x} L_A (x_j) \right),
\]

where \( \mathcal{X}' = \text{map}(\mathcal{M}') \). Further reduction in the computational complexity comes from reducing the search space in the lattice \( \mathcal{X}' \) where we only consider a pre-defined \( N_L \) number of closest lattice points, \( L \), to the vector \( o^{(i)} \) and exclude the rest from \( \mathcal{X}' \). The results in [7] showed that such approximated LLR values is very close to the exact values of LLR.

The proposed LCC is depicted visually in Fig. 3.

**IV. Computational Complexity Analysis**

As can be seen in Fig. 3, the first stage of the online process of the proposed LCC to detect the \( i \)-th sample is to identify \( o^{(i)} \) samples which have constant computational complexity. The second step requires identifying a radius that includes \( N_L \) closest points to \( o^{(i)} \) 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 \( o^{(i)} \) 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.

**V. 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 [7], 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 \( K = 1 \), from scikit-learn python library as an example of distance-based classifiers.

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 plots 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 \( N_p \). This is as at high values of \( N_p \), i.e., 13 and 15, the increase of the distance appears to be saturating. This idea is also supported by Fig. 6 where the performance difference between \( N_p = 13 \) and \( N_p = 15 \) is less than 0.5 dB. 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. Please note that the choice of the value of \( N_p \) here in the reduced search over lattice is somewhat similar to the concept in coding theory where 5 to 7 times the constraint length is sufficient for determining the current trellis state in reduced-search based trellis detectors [3], [10].

To study the effect of \( N_t \) on the BER performance, in Fig. 5 we plot the BER of the proposed LCC at \( N_t = 3 \) and 5 and
Similarly, at BER of $10^{-4}$, application for higher values of $N_p$ in the DL-LSD has an exponential complexity in $N$, which limits its application for higher values of $N$ unlike the proposed LLC. Similarly, at BER of $10^{-4}$, there is about 1.5 dB difference between the proposed LLC and the M-BCJR algorithm. It is worth to emphasize that in the M-BCJR the computational complexity is exponential in the ISI length; however, in the proposed LLC, the complexity is linear in $N$ and $N_p$.

VI. CONCLUSION

FTN signaling is a promising technique in future communication systems since it improves the SE without changing the transmission bandwidth. In this letter, we proposed a low-complexity LLC algorithm that exploits the ISI structure of FTN signaling to perform the classification task in $N_p \ll N$-dimension space. Simulation results showed the trade-off between computational complexity and BER performance for both the uncoded and coded cases. For example, at $\tau = 0.6$, there are 0.4 dB and 1.5 dB to the optimal performance of the uncoded and coded transmission, respectively, with lower computational complexity.

REFERENCES

[1] J. B. Anderson, F. Rusek, and V. Öwall, “Faster-than-Nyquist signaling,” Proc. IEEE, vol. 101, no. 8, pp. 1817–1830, Aug. 2013.
[2] 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 Trans. Commun., vol. 68, no. 7, pp. 4011–4021, Jul. 2020.
[3] S. Li, B. Bai, J. Zhou, P. Chen, and Z. Yu, “Reduced-complexity equalization for faster-than-Nyquist signaling: New methods based on an Ungerboeck observation model,” IEEE Trans. Commun., vol. 66, no. 3, pp. 1190–1204, Mar. 2018.
[4] A. Ibrahim, E. Bedeir, and H. Yanikomeroglu, “A novel low complexity faster-than-Nyquist (FTN) signaling detector for ultra high-order QAM,” IEEE Open J. Commun. Soc., vol. 2, pp. 2566–2580, 2021.
[5] S. Li, W. Yuan, J. Yuan, B. Bai, D. Wing Kwan Ng, and L. Hanzo, “Time-domain vs. frequency-domain equalization for faster-than-Nyquist signaling,” IEEE Trans. Veh. Technol., vol. 69, no. 8, pp. 9233–9247, Aug. 2020.
[6] 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. 68008–68019, 2020.
[7] S. Abbasi and E. Bedeir, “Deep learning-based list sphere decoding for faster-than-Nyquist (FTN) signaling detection,” Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring), Jun. 2022. [Online]. Available: https://doi.org/10.1109/VTC2022-Spring.54318.2022.9860470
[8] 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 Trans. Commun., vol. 69, no. 9, pp. 5975–5987, Sep. 2021.
[9] J. B. Anderson, Coded Modulation Systems. Norwell, MA, USA: Kluwer, 2002.
[10] M. Loncar and F. Rusek, “On reduced-complexity equalization based on Ungerboeck and Forney observation models,” IEEE Trans. Signal Process., vol. 56, no. 8, pp. 3784–3789, Aug. 2008.