

FTN-GFDM Detection Based on Reduced-Complexity Soft Sphere Decoding Integrated with Polar Codes

Mariana Baracat de Mello, Luciano Leonel Mendes, Daniely Gomes Silva, Paulo Ricardo Branco da Silva, and Tiago Cardoso Barbosa

Abstract—In remote rural areas, it is not possible to employ massive multiple-input multiple-output (MIMO), small cells, and ultra-dense networks (UDNs) with the aim of increasing throughput. A solution is to improve the waveform spectral efficiency, integrating faster than Nyquist (FTN) signaling with generalized frequency division multiplexing (GFDM). However, this presents high self-interference in the time and frequency domains, requiring dedicated detectors for performance loss mitigation. Hard decision detection schemes primarily designed for MIMO have been adapted to detect FTN-GFDM signals without degradation of the uncoded bit error rate (BER), but these schemes are suboptimal in terms of capacity as they do not provide all the information contained in log-likelihood ratios (LLRs). We design and evaluate in this paper a soft sphere detector (SD) algorithm for FTN-GFDM that can be integrated with state-of-the-art forward error control (FEC) decoders for good BER performance over mobile channels. The SD detector is combined with polar codes, and the BER and complexity are evaluated for different channel models. The results show that FTN-GFDM can provide high spectrum efficiency gains without significant coded BER losses and with affordable complexity on the receiver side, which makes this waveform an interesting candidate for mobile networks in remote areas.

Index Terms—Faster than Nyquist, generalized frequency division multiplexing, polar coding, single tree search algorithm, soft sphere decoder.

I. INTRODUCTION

THE spectral efficiency of beyond fifth generation (5G) and sixth generation (6G) networks needs to increase

Manuscript received May 15, 2023; approved for publication November 29, 2023; approved for publication by Yousefi'zadeh, Homayoun Division 2 Editor, April 17, 2024.

This work has received partial funding from the project XGM-AFCCT-2024-2-15-1, supported by xGMobile – EMBRAPII-Inatel Competence Center on 5G and 6G Networks, with financial resources from the PPI IoT/Manufacturing 4.0 program of MCTI grant number 052/2023, signed with EMBRAPII. Additionally, this work was partially supported by the project APQ-00810-21 funded by FAPEMIG, the MAI/DAI project (403612/2020-9 funded by CNPq and CPQD), the SEMEAR project (22/09319-9 funded by FAPESP), and CNPq-Brasil.

M. B. de Mello, L. L. Mendes, D. G. Silva, and T. C. Barbosa are with Instituto Nacional de Telecomunicações (Inatel), Santa Rita do Sapucaí, MG, 37540-000, Brazil, email: {mariana.baracat, luciano1, daniely, tiago.cardoso}@inatel.br.

P. R. B. da Silva is in Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD), Campinas, SP, 13086-902, Brazil, email: prsilva@cpqd.com.br.

M. B. de Mello is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2024.000018

significantly to enable the high volume of data traffic predicted for future applications. Several new applications scenarios are being proposed by researchers around the world to provide global connectivity, invisible security, digital twins and other high demanding use cases [1]. Massive multiple-input multiple-output (MIMO) and ultra-dense network (UDN) improve capacity in densely populated areas, but these technologies are not readily in remote and rural areas [2]. Since enhanced remote area communications (eRAC) applications are more likely to operate in bands below 1 GHz, it is desirable to increase the spectral efficiency of the waveform to achieve higher data rates. By using non-orthogonal waveforms compressed in time and frequency domains beyond the Nyquist limit for interference-free communication, it is possible to increase the number of transmitted bits per sample of the waveform, achieving higher data rates and overall better quality of experience (QoE). This requires a dedicated detector because linear receivers designed for orthogonal waveforms cannot mitigate the interference, leading to a poor bit error rate (BER) performance.

In principle, faster than Nyquist (FTN) can be applied to any waveform, but generalized frequency division multiplexing (GFDM) [3] stands out among possible waveforms for 6G networks because of its flexibility to address different requirements, such as low adjacent channel leakage ratio (ACLR), low peak-to-average power ratio (PAPR) or high cyclic prefix (CP) efficiency [4]. Also, GFDM has Orthogonal frequency division multiplexing (OFDM), discrete Fourier transform spread orthogonal frequency division multiplexing (DFT-s-OFDM), orthogonal time frequency space (OTFS) and other waveforms as corner cases [5], [6], making it compatible with legacy networks or other wireless standards.

Due to its flexibility, FTN can be easily applied to GFDM simultaneously in time and frequency domains, increasing the overall spectrum efficiency gains if the Mazo limit [7], [8] in each dimension is respected. This approach leads to a new waveform, defined as FTN-GFDM, which has been shown to provide higher spectrum efficiency without uncoded BER performance loss [9].

However, in order to achieve the desirable performance, the receiver must resolve the interference introduced by the FTN signaling. It has been shown that non-linear MIMO detectors can be redesigned to address the self-interference in FTN-GFDM waveform [10]. From the available detectors sphere decoder (SD) has shown to achieve the best uncoded

Creative Commons Attribution-NonCommercial (CC BY-NC).

This is an Open Access article distributed under the terms of Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided that the original work is properly cited.

BER performance with an affordable complexity. Although, the SD detector available in the literature delivers hard decisions for the forward error correction (FEC) decoder, which means that the error correction algorithm cannot provide its best performance. In order to achieve the optimal performance, the FEC decoder needs the log-likelihood ratios (LLRs) from the waveform detector, which means that the SD must provide a soft output.

The main aim of this paper is to design a soft SD detector for FTN-GFDM system and evaluate its performance in terms of complexity and coded BER when integrated with a state-of-the-art FEC scheme. Polar codes have been selected to be integrated with the FTN-GFDM because of its high error correction capability and flexibility in terms of code rates and codeword length [11], [12]. Polar codes have also been shown to achieve the channel capacity for both discrete and continuous memoryless channels [11]. Polar codes also can be designed to be systematic, reducing the encoder and decoder complexities. These characteristics are beneficial for eRAC applications in remote and rural areas, since the trade-off between complexity, robustness and data rate can be tailored for different situations. Furthermore, polar codes have attracted interest from the academic community, having been chosen by the 3rd Generation Partnership Project (3GPP) as the FEC scheme for the control channel in 5G networks [13]. The key contributions of this paper, when compared to the existing literature, are as follows:

- Design of a soft-output detector based on SD for the FTN-GFDM waveform;
- Integration with a coding scheme to evaluate the performance gain achieved from using a soft-output detector for FTN-GFDM;
- Performance analysis under different channel models; and
- Complexity analysis and comparison between soft and hard decision detectors.

It is important to highlight that the aim of this paper is not an exhaustive exploration of all state-of-the-art FEC schemes in order to define the scheme that achieves the best performance. Instead, the contributions presented are closely related to the design, implementation, and evaluation of a soft-output detection scheme for the FTN-GFDM waveform, which will benefit the performance of any FEC scheme that can take advantage of the extra information provided by such detector. This paper shows that simple FEC schemes can also benefit from a soft-output detector, achieving better BER performance even when low complexity FEC decoders are employed.

The remainder of this paper has the following structure. Section II presents the proposed system model. Section III describes the design of the soft SD detector, in particular the use of tree search algorithms to find symbols that minimize the distance to the discrete filtered received signal. The detector's complexity analysis is also presented in this section. Section IV presents the coded BER performance of the proposed detector in comparison with hard decision SD assuming different channel models. Also, the complexity of the proposed detector is compared with the hard SD in terms of floating-point operations (FLOPs). Finally, Section V

concludes this paper and presents future research directions.

II. SYSTEM MODEL

The block diagram of the proposed FTN-GFDM transceiver is shown in Fig. 1. Each subsystem is described in the following subsections.

A. Polar Encoding and Decoding

Polar coding is a channel coding technique that uses polarized bits to improve the reliability of transmission over noisy channels [11]. Polar codes were first introduced in [11] and gained considerable attention due to their excellent error correction performance, scalability, and low decoding complexity. The idea of polar codes is to take a channel with a high error probability and transform it into two channels, one with a low error probability and one with a high error probability. The information is transmitted on the low error probability channel, while the high error probability channel is used for error correction [12]. Therefore, polar coding is integrated before FTN-GFDM modulation and after FTN-GFDM detection to ensure the reliability of the communication system.

1) *Encoding*: The polar encoder block, shown in Fig. 1, converts a block of information bits $\mathbf{b}_i \in \mathbb{R}^{N_i \times 1}$ into a sequence of coded bits $\mathbf{b}_c \in \mathbb{R}^{N_c \times 1}$. The systematic encoding is performed by two non-systematic encoders interleaved with the insertion of frozen bits between them. The non-systematic encoding process can be defined as

$$\mathbf{b}_c = \mathbf{b}_u \mathbf{B}_n \mathbf{F}^{(\oplus n)}, \quad (1)$$

where $\mathbf{b}_u \in \mathbb{R}^{N_c \times 1}$ is formed by concatenating the information bits \mathbf{b}_i with $N_c - N_i$ frozen bits, and $\mathbf{F}^{(\oplus n)} \in \mathbb{R}^{N_c \times N_c}$ is the n th power Kronecker matrix, which its first power is given by

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (2)$$

The permutation matrix $\mathbf{B}_n \in \mathbb{R}^{N_c \times N_c}$ is called bit-reversal matrix and it reverses the row positions of $\mathbf{F}^{(\oplus n)}$. The coding rate can be expressed as $R_c = N_i/N_c$, where N_i is the number of information bits and N_c is the number of coded bits, with N_c being a power of 2.

The code was constructed as proposed in [14], where an exhaustive search for the set of frozen bits was performed. Firstly, a range of signal-to-noise ratio (SNR) of interest was considered. Then, \mathcal{X} polar codes were designed using the Bhattacharyya parameter [11], where \mathcal{X} represents the SNR set of interest. The BER and frame error rate (FER) were evaluated and, based on the results, the code with the best performance was selected. As demonstrated in [14], the performance is independent of the method when the code is optimized for the best performance considering BER and FER parameters.

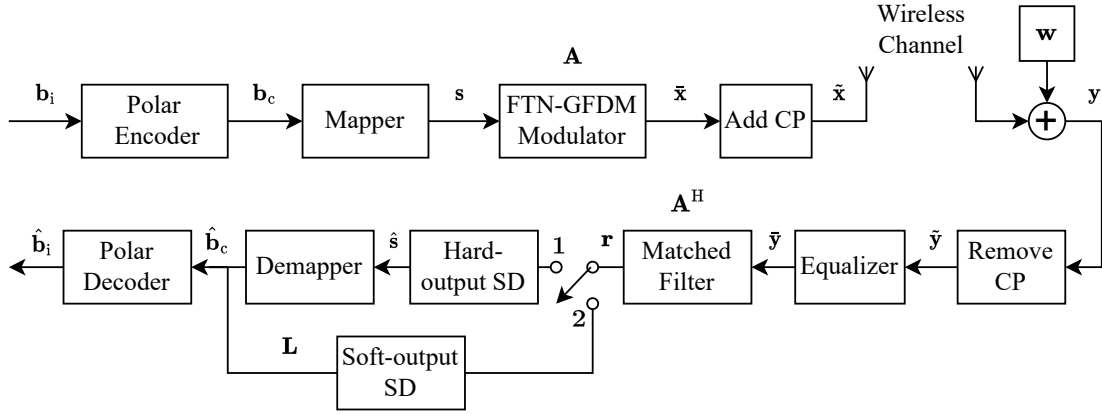


Fig. 1. FTN-GFDM transceiver block diagram.

2) *Puncturing*: To integrate the polar code into the FTN-GFDM system, the number of coded bits N_c must be divisible by $N = KM$, where K and M are the number of FTN-GFDM subcarriers and subsymbols, respectively. Otherwise, some bits must be punctured. The number of punctured bits is equal to the remainder of the division. As shown in [11], the code performance increases with N_c and, hence, this parameter shall be chosen to result in the lowest number of punctured bits possible.

The punctured bits are initially inserted in the last positions of \mathbf{b}_u . The initial puncturing pattern is derived from the column weight of $\mathbf{F}^{(\oplus n)}$ based on the non-systematic polar encoding structure. A bit reversal permutation \mathbf{B}_n is applied to generate the puncturing pattern \mathcal{I}_p . Once this is done, the punctured bits are considered part of the information bits, and only the frozen bits are re-inserted after the first encoding process. Finally, using the knowledge of \mathcal{I}_p , the punctured bits are removed from \mathbf{b}_c and they are not transmitted.

At the receiver side, before decoding, the punctured bits must be re-inserted. Assuming that the punctured bit positions at the receiver can be filled with “0”, then the LLRs at these positions can be set to infinity [15].

3) *Decoding*: The polar decoder uses a decoding algorithm called successive cancellation (SC), which operates by iteratively canceling the effect of unreliable bits in the code [11].

In the SC algorithm, $\hat{\mathbf{b}}_u$ is estimated by a recursive computation of likelihood ratios (LRs) of $\hat{\mathbf{b}}_c$. The ι th decoded bit, \hat{b}_u^ι , is defined by

$$\hat{b}_u^\iota = \mathcal{H}(\text{LR}[\hat{\mathbf{b}}_c, \hat{\mathbf{b}}_u^{(0:\iota-1)}]). \quad (3)$$

Here, $\hat{\mathbf{b}}_u^{(0:\iota-1)} = \{\hat{b}_u^0, \hat{b}_u^1, \dots, \hat{b}_u^{(\iota-1)}\} \in \mathbb{R}^{\iota \times 1}$ and $\mathcal{H}(\cdot)$ is decision function given by

$$\mathcal{H}(\text{LR}[\hat{\mathbf{b}}_c, \hat{\mathbf{b}}_u^{(0:\iota-1)}]) = \begin{cases} 0 & \text{if } \text{LR}[\hat{\mathbf{b}}_c, \hat{\mathbf{b}}_u^{(0:\iota-1)}] \leq 1 \text{ or } \iota \in \mathcal{I}_f, \\ 1 & \text{otherwise,} \end{cases} \quad (4)$$

where \mathcal{I}_f is the subset of frozen bits index and

$$\text{LR}[\hat{\mathbf{b}}_c, \hat{\mathbf{b}}_u^{(0:\iota-1)}] = \frac{\Pr[(\hat{\mathbf{b}}_c, \hat{\mathbf{b}}_u^{(0:\iota-1)}) | \hat{b}_u^\iota = 0]}{\Pr[(\hat{\mathbf{b}}_c, \hat{\mathbf{b}}_u^{(0:\iota-1)}) | \hat{b}_u^\iota = 1]}. \quad (5)$$

Here, $\Pr[(\hat{\mathbf{b}}_c, \hat{\mathbf{b}}_u^{(0:\iota-1)}) | \hat{b}_u^\iota = s]$ represents the conditional probability of the received codeword being $\hat{\mathbf{b}}_c$ and the previously decoded bits being $\hat{\mathbf{b}}_u^{(0:\iota-1)}$, given that $\hat{b}_u^\iota = s \in \{0, 1\}$. LR in (5) can be computed recursively using a graph defined by the functions f and q , each given by

$$f(a, b) = \frac{1 + ab}{a + b}, \quad (6)$$

and

$$q(a, b, \hat{b}_u^{(\text{sum})}) = a^{1-2\hat{b}_u^{(\text{sum})}} b, \quad (7)$$

where $\hat{b}_u^{(\text{sum})}$ is the partial sum of the elements of $\hat{\mathbf{b}}_u^{(0:\iota-1)}$. Thus, the LR can be computed as

$$L_{\ell, \iota} = \begin{cases} f(L_{\ell+1, \iota}, L_{\ell+1, \iota+2^\ell}) & \text{if } \langle \frac{\iota}{2^\ell} \rangle_2 = 0, \\ q(\hat{b}_u^{(\text{sum})}_{\ell, \iota-2^\ell}, L_{\ell+1, \iota-2^\ell}, L_{\ell+1, \iota}) & \text{otherwise,} \end{cases} \quad (8)$$

where $0 \leq \ell < \log_2 N_c$ is the graph stage, ι is the index of the current decoded bit.

In the FTN-GFDM transceiver of Fig. 1, estimates of the coded bits, $\hat{\mathbf{b}}_c \in \mathbb{R}^{N_s \times 1}$, or the LLR values, $\mathbf{L} \in \mathbb{R}^{N_c \times 1}$, can be supplied to the polar decoder when the switch is in position 1 or 2, respectively. The vector containing the estimated information bits, $\hat{\mathbf{b}}_i \in \mathbb{R}^{N_s \times 1}$, is delivered at the output of the polar decoder.

The SC polar decoder with LLR has been shown to provide a good trade-off between decoding performance and computational complexity. It can achieve near-optimal performance with moderate computational complexity, making it suitable for practical applications [16]. Despite its effectiveness, when compared to more sophisticated decoding techniques, such as SC-list [17] and SC-flip [18], the SC algorithm is considered a baseline with inferior performance. Nonetheless, the purpose of this article is to demonstrate that even the simplest decoder can benefit from LLRs and improve BER compared to the hard output detectors.

B. Symbol Mapping and Demapping

In the block diagram of Fig. 1, the mapper converts the coded bits \mathbf{b}_c into J -quadrature amplitude modulation (QAM) symbols and allocates the symbols into the FTN-GFDM data

matrix, $\mathbf{S} \in \mathbb{C}^{K \times M}$. As a result, \mathbf{S} contains the N QAM symbols to be transmitted in an FTN-GFDM block, such that

$$\mathbf{S} = \begin{bmatrix} s_{0,0} & s_{0,1} & \cdots & s_{0,M-1} \\ s_{1,0} & s_{1,1} & \cdots & s_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{K-1,0} & s_{K-1,1} & \cdots & s_{K-1,M-1} \end{bmatrix}. \quad (9)$$

On the receiver side, the demapper is responsible for converting each QAM symbol provided by the hard output detector into its sequence of $\log_2 J$ bits. Thus, the demapper outputs $N \log_2 J$ bits for each received FTN-GFDM block.

C. FTN-GFDM Modulation

Slight modifications in the definition of the time-frequency grid enable GFDM to include FTN signaling [9], [19]. Let the prototype filter have \mathcal{N} samples divided into \mathcal{P} periods with \mathcal{S} samples each. Let also the subsymbols be defined at time intervals of \mathcal{K} samples and the subcarriers at frequency intervals of \mathcal{M} samples. Using these definitions, the GFDM signal is given by

$$x[n] = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} s_{k,m} g[\langle n - m\mathcal{K} \rangle_{\mathcal{N}}] \exp\left(j2\pi \frac{k\mathcal{M}}{\mathcal{N}} n\right), \quad (10)$$

where $n = 0, \dots, \mathcal{N} - 1$, $s_{k,m}$ is the QAM symbol for the k th subcarrier and m th subsymbol, and $g[n]$ is the prototype filter impulse response.

There are three approaches to apply FTN signaling in GFDM. The first one consists of decreasing the time delay between the prototype filters for each subsymbol ($\mathcal{K} < \mathcal{S}$), increasing the time overlap among the subsymbols. The second approach consists of reducing the frequency spacing among the subcarriers ($\mathcal{M} < \mathcal{P}$), increasing the overlapping in the frequency domain. The third approach is the combination of the first two.

The spectral efficiency gain can be computed by the squeezing factors in time and frequency, namely $v_t = \mathcal{K}/\mathcal{S}$ and $v_f = \mathcal{M}/\mathcal{P}$, respectively.

It is important to highlight that

$$K = \frac{\mathcal{P}\mathcal{S}}{\mathcal{M}} = \frac{\mathcal{S}}{v_f} = \left\lfloor \frac{\mathcal{N}}{\mathcal{M}} \right\rfloor, \quad (11)$$

$$M = \frac{\mathcal{P}\mathcal{S}}{\mathcal{K}} = \frac{\mathcal{P}}{v_t} = \left\lfloor \frac{\mathcal{N}}{\mathcal{K}} \right\rfloor, \quad (12)$$

and when \mathcal{N}/\mathcal{M} and \mathcal{N}/\mathcal{K} are not integers, the squeezing factors must be adjusted by $\bar{v}_t = \mathcal{P}/M$ and $\bar{v}_f = \mathcal{S}/K$, respectively, and the GFDM signal must be adjusted to

$$\bar{x}[n] = \sqrt{\bar{v}_t \bar{v}_f} \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} s_{k,m} g[\langle n - mv_t \mathcal{S} \rangle_{\mathcal{N}}] \exp\left(j2\pi \frac{kv_f}{\mathcal{S}} n\right), \quad (13)$$

where $\sqrt{\bar{v}_t \bar{v}_f}$ normalizes the transmit power. Fig. 2 shows the block diagram of the FTN-GFDM transmitter, where $\mathbf{s} = \text{vec}(\mathbf{S}) \in \mathbb{C}^{N \times 1}$ is the symbol vector obtained by stacking the columns of \mathbf{S} , i.e., $\mathbf{s} = (s_{0,0}, s_{1,0}, \dots, s_{K-1,0}, s_{0,1}, \dots, s_{K-1,M-1})^T$. This vector is parallelized and the element $s_{m,k}$ is carried

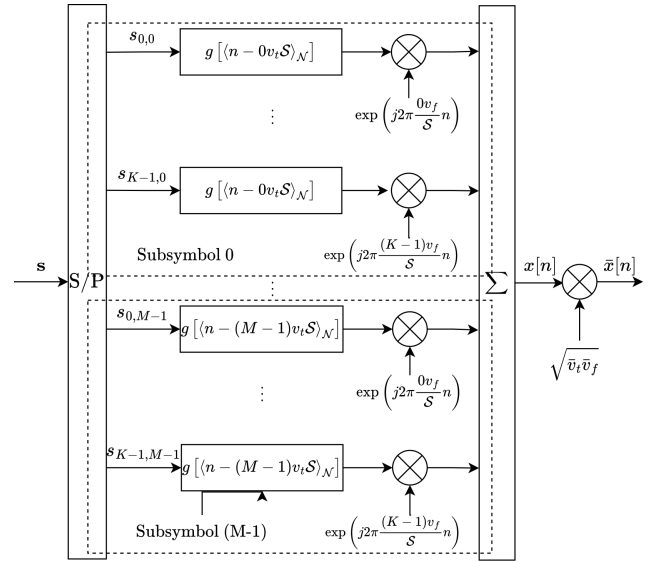


Fig. 2. FTN-GFDM modulator block diagram.

by the m th subsymbol, denoted by $g[\langle n - mv_t \mathcal{S} \rangle_{\mathcal{N}}]$, of the k th subcarrier, denoted by $\exp\left(j2\pi \frac{kv_f}{\mathcal{S}} n\right)$. The FTN-GFDM signal, $x[n]$, is obtained by adding the subsymbols from all subcarriers. As described in (13), power normalization is used to generate $\bar{x}[n]$.

In matrix notation, vector $\bar{\mathbf{x}} = (\bar{x}[0], \bar{x}[1], \dots, \bar{x}[\mathcal{N} - 1])^T \in \mathbb{C}^{\mathcal{N} \times 1}$ can be computed as

$$\bar{\mathbf{x}} = \mathbf{A}\mathbf{s}, \quad (14)$$

where $\mathbf{A} \in \mathbb{C}^{\mathcal{N} \times N}$ is the transmission matrix containing $N = KM$ versions of the prototype filter $\mathbf{g} \in \mathbb{C}^{\mathcal{N} \times 1}$ circularly shifted in time and frequency, i.e.,

$$\mathbf{A} = [\mathbf{g}_{0,0} \ \mathbf{g}_{1,0} \ \cdots \ \mathbf{g}_{K-1,0} \ \mathbf{g}_{0,1} \ \cdots \ \mathbf{g}_{K-1,M-1}]. \quad (15)$$

The column vector

$$\mathbf{g}_{k,m} = \sqrt{\bar{v}_t \bar{v}_f} g[\langle n - mv_t \mathcal{S} \rangle_{\mathcal{N}}] \exp\left(j2\pi \frac{kv_f}{\mathcal{S}} n\right), \quad (16)$$

contains the prototype filter samples shifted to the k th subcarrier and the m th subsymbol. Before transmitting the signal $\bar{\mathbf{x}} \in \mathbb{C}^{\mathcal{N} \times 1}$, a CP can be added to the FTN-GFDM vector by copying the last N_{CP} samples from $\bar{\mathbf{x}}$ to the beginning, thus creating the transmission vector $\tilde{\mathbf{x}} \in \mathbb{C}^{(\mathcal{N} + N_{\text{CP}}) \times 1}$.

D. Channel and Equalization

The received vector $\tilde{\mathbf{y}} \in \mathbb{C}^{\mathcal{N} \times 1}$ after removing CP is given by

$$\tilde{\mathbf{y}} = \mathbf{H}\tilde{\mathbf{x}} + \mathbf{w}, \quad (17)$$

where $\mathbf{w} \in \mathbb{C}^{\mathcal{N} \times 1}$ is the additive white Gaussian noise (AWGN) vector and $\mathbf{H} \in \mathbb{C}^{\mathcal{N} \times \mathcal{N}}$ contains circularly

shifted versions of the channel impulse response $\mathbf{h} \in \mathbb{C}^{\mathcal{L} \times 1}$, such that

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \cdots & h_{\mathcal{L}-1} & \cdots & h_1 \\ \vdots & h_0 & \ddots & 0 & \ddots & \vdots \\ h_{\mathcal{L}-1} & \vdots & \ddots & \vdots & \ddots & h_{\mathcal{L}-1} \\ 0 & h_{\mathcal{L}-1} & \ddots & h_0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{\mathcal{L}-2} & \cdots & h_0 \end{bmatrix}. \quad (18)$$

In this paper, three channel models are considered: the additive white Gaussian noise (AWGN) channel, the time-invariant frequency-selective (TIFS) channel, and the block time-variant flat (TVF) channel. In particular, a TVF channel with a Rayleigh tap changes with each FTN-GFDM block, and a flat frequency response is assumed, i.e., the channel response varies with each transmitted FTN-GFDM block, but it is constant over the block interval.

The received signal can be represented in the frequency domain as

$$\tilde{\mathbf{Y}} = \mathbf{F}_{\mathcal{N}} \mathbf{H} \bar{\mathbf{x}} + \mathbf{F}_{\mathcal{N}} \mathbf{w} = \mathbf{F}_{\mathcal{N}} \mathbf{H} \mathbf{F}_{\mathcal{N}}^H \bar{\mathbf{X}} + \mathbf{W}, \quad (19)$$

where $\mathbf{F}_{\mathcal{N}} \in \mathbb{C}^{\mathcal{N} \times \mathcal{N}}$ is the Fourier matrix, $\bar{\mathbf{x}} \in \mathbb{C}^{\mathcal{N} \times 1}$ is the transmit vector in the frequency domain, $\mathbf{W} \in \mathbb{C}^{\mathcal{N} \times 1}$ is the AWGN vector in the frequency domain, and $(\cdot)^H$ represents the Hermitian operator. Additionally, $\mathbf{F}_{\mathcal{N}} \mathbf{H} \mathbf{F}_{\mathcal{N}}^H$ is a diagonal matrix containing the channel frequency response. Assuming knowledge of the channel impulse response, the equalization in the frequency domain is given by

$$\bar{\mathbf{Y}} = (\mathbf{F}_{\mathcal{N}} \mathbf{H} \mathbf{F}_{\mathcal{N}}^H)^{-1} \tilde{\mathbf{Y}} = \bar{\mathbf{X}} + \mathbf{F}_{\mathcal{N}} \mathbf{H}^{-1} \mathbf{F}_{\mathcal{N}}^H \mathbf{W}, \quad (20)$$

where $\bar{\mathbf{Y}} \in \mathbb{C}^{\mathcal{N} \times 1}$ is the equalized received vector in the frequency domain. Alternatively, $\bar{\mathbf{Y}}$ can be represented in the time domain by

$$\bar{\mathbf{y}} = \mathbf{F}_{\mathcal{N}}^H \bar{\mathbf{Y}} = \mathbf{F}_{\mathcal{N}}^H \bar{\mathbf{X}} + \mathbf{F}_{\mathcal{N}}^H \mathbf{F}_{\mathcal{N}} \mathbf{H}^{-1} \mathbf{F}_{\mathcal{N}}^H \mathbf{W} = \bar{\mathbf{x}} + \mathbf{H}^{-1} \mathbf{w}, \quad (21)$$

where $\bar{\mathbf{y}} \in \mathbb{C}^{\mathcal{N} \times 1}$ is the equalized received vector in the time domain.

E. FTN-GFDM Demodulation

The matched filter (MF) is used to decouple the FTN-GFDM subcarriers and subsymbols in $\bar{\mathbf{y}}$ to allow proper detection of the symbols vector. Furthermore, MF maximizes the signal-to-interference-plus-noise ratio (SINR) of $\bar{\mathbf{y}}$. The received discrete signal at the MF output $\mathbf{r} \in \mathbb{C}^{\mathcal{N} \times 1}$ is given by

$$\mathbf{r} = \mathbf{A}^H \bar{\mathbf{y}} = \mathbf{A}^H \bar{\mathbf{x}} + \mathbf{A}^H \mathbf{H}^{-1} \mathbf{w} = \mathbf{G} \mathbf{s} + \bar{\mathbf{w}}, \quad (22)$$

where $\mathbf{G} = \mathbf{A}^H \mathbf{A} \in \mathbb{C}^{\mathcal{N} \times \mathcal{N}}$ is the correlation coefficient matrix, $\bar{\mathbf{w}} = \mathbf{A}^H \mathbf{H}^{-1} \mathbf{w} \in \mathbb{C}^{\mathcal{N} \times 1}$. Since the linear operation $\mathbf{A}^H \mathbf{H}^{-1}$ correlates the samples of \mathbf{w} , the filtered samples in $\bar{\mathbf{w}}$ are colored.

The MF block diagram is shown in Fig. 3, where $\bar{y}[n]$ denotes the n th sample in the equalizer output and the index $n = 0, \dots, \mathcal{N} - 1$ is reset with each new FTN-GFDM block received.

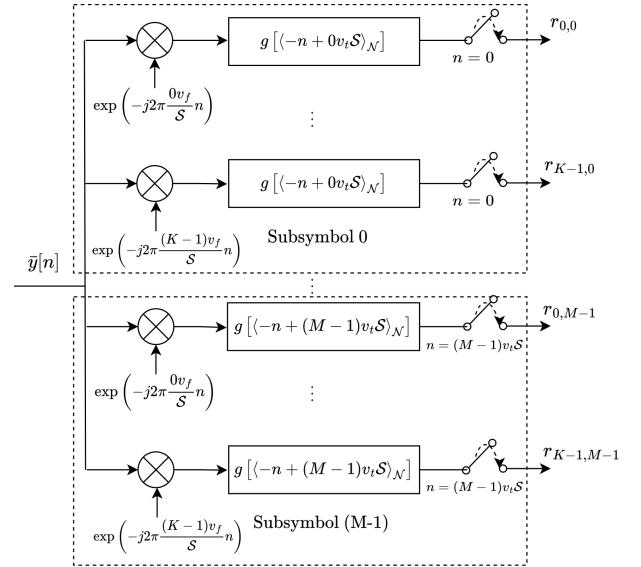


Fig. 3. Matched filter block diagram.

F. Soft FTN-GFDM Detection

Given that the ν th J -QAM symbol, with $\nu = 1, \dots, N$, represents a set of $\log_2 J$ bits, i.e., $\{b_{\nu,1}, \dots, b_{\nu, \log_2 J}\}$, the soft-output FTN-GFDM detection is calculated as follows

$$L_{\nu,\mu} = \log \left(\frac{\Pr[\hat{b}_{\nu,\mu} = 1 | \mathbf{r}]}{\Pr[\hat{b}_{\nu,\mu} = 0 | \mathbf{r}]} \right), \quad (23)$$

where $L_{\nu,\mu}$ is the LLR of the μ th bit in the ν th symbol of the estimated QAM symbol sequence vector $\hat{\mathbf{s}} \in \mathbb{C}^{\mathcal{N} \times 1}$, whose binary representation $\hat{\mathbf{b}} \in \mathbb{R}^{(N \log_2 J) \times 1}$ is given by

$$\hat{\mathbf{b}} = (\hat{b}_{1,1}, \dots, \hat{b}_{1, \log_2 J}, \hat{b}_{2,1}, \dots, \hat{b}_{\nu,\mu}, \dots, \hat{b}_{N, \log_2 J}). \quad (24)$$

The direct computation of the LLRs in (23) results in prohibitive computational complexity. The max-log approximation can be employed to reduce the complexity. Hence, equation (23) can be approximated as

$$L_{\nu,\mu} \approx \min_{\hat{\mathbf{s}} \in \mathbb{M}_{\nu,\mu}^{(0)}} \|\mathbf{r} - \mathbf{G}\hat{\mathbf{s}}\|^2 - \min_{\hat{\mathbf{s}} \in \mathbb{M}_{\nu,\mu}^{(1)}} \|\mathbf{r} - \mathbf{G}\hat{\mathbf{s}}\|^2, \quad (25)$$

where $\mathbb{M}_{\nu,\mu}^{(0)}$ and $\mathbb{M}_{\nu,\mu}^{(1)}$ represent the FTN-GFDM sequences that have $b_{\nu,\mu} = 0$ and $b_{\nu,\mu} = 1$, respectively. For each bit $\hat{b}_{\nu,\mu}$, one of the minima in (25) will be the Euclidean distance of the maximum likelihood (ML) solution, while the other will be the minimum Euclidean distance of the ML antipodal solution [20].

The ML solution $\hat{\mathbf{s}}^{\text{ML}} \in \mathbb{C}^{\mathcal{N} \times 1}$ is given by

$$\hat{\mathbf{s}}^{\text{ML}} = \arg \min_{\hat{\mathbf{s}} \in \mathbb{M}} \|\mathbf{r} - \mathbf{G}\hat{\mathbf{s}}\|^2, \quad (26)$$

where \mathbb{M} is the set of all possible J -QAM symbol sequences. The distance d^{ML} is the Euclidean distance of the ML solution $\hat{\mathbf{s}}^{\text{ML}}$, given by

$$d^{\text{ML}} = \|\mathbf{r} - \mathbf{G}\hat{\mathbf{s}}^{\text{ML}}\|^2, \quad (27)$$

and $\overline{d}_{\nu,\mu}^{\text{ML}}$ is the minimum Euclidean distance of the ML antipodal solution, given by

$$\overline{d}_{\nu,\mu}^{\text{ML}} = \min_{\hat{\mathbf{s}} \in \mathbb{M}^{\hat{b}_{\nu,\mu}^{\text{ML}}}} \|\mathbf{r} - \mathbf{G}\hat{\mathbf{s}}\|^2, \quad (28)$$

where $\hat{b}_{\nu,\mu}^{\text{ML}}$ is the μ th bit of the ν th symbol of the ML solution $\hat{\mathbf{s}}^{\text{ML}}$ and $\hat{b}_{\nu,\mu}^{\overline{\text{ML}}}$ is its binary complement. Then, $\mathbb{M}^{\hat{b}_{\nu,\mu}^{\overline{\text{ML}}}}$ are the FTN-GFDM sequences that have $\hat{b}_{\nu,\mu}^{\overline{\text{ML}}}$ for the (ν, μ) element.

With the aid of (26), (27), and (28), equation (25) can be rewritten as

$$L_{\nu,\mu} = \begin{cases} d_{\nu,\mu}^{\text{ML}} - d_{\nu,\mu}^{\overline{\text{ML}}}, & \text{if } \hat{b}_{\nu,\mu}^{\text{ML}} = 0, \\ d_{\nu,\mu}^{\overline{\text{ML}}} - d_{\nu,\mu}^{\text{ML}}, & \text{if } \hat{b}_{\nu,\mu}^{\text{ML}} = 1. \end{cases} \quad (29)$$

To calculate the LLRs in (29), it is necessary to find the Euclidean distance of the ML antipodal solution, $d_{\nu,\mu}^{\overline{\text{ML}}}$, for each bit of each symbol of the FTN-GFDM signal [20]. Since soft-output detection uses the minimum Euclidean distances of all antipodal ML solutions, soft-output detection results in higher computational complexity than hard-output detection [20], [21].

The maximum likelihood sequence estimation (MLSE) detector can find d^{ML} and $\mathbf{d}^{\overline{\text{ML}}} \in \mathbb{R}^{(N \log_2 J) \times 1}$ in (29). Since the Mazo limit is respected, the MLSE detector can effectively handle the interference caused by FTN-GFDM and reduce the BER. The search for the ML solution is performed as in (27). As a result, all possible combinations of transmitted QAM symbols are exhaustively scanned to find the one that gives the minimum Euclidean distance for \mathbf{r} . Then, the MLSE detector must find the ML antipodal solution for each bit of $\hat{\mathbf{b}}^{\text{ML}} \in \mathbb{R}^{(N \log_2 J) \times 1}$ as in (28). The set of possible ML antipodal solutions for each bit $\mathbb{M}^{\hat{b}_{\nu,\mu}^{\overline{\text{ML}}}}$ consists of $J^N/2$ combinations of QAM symbols.

A FLOP [22] can be defined as an addition, subtraction, division, or multiplication operation in the real domain. Thus, the computational complexity of MLSE detection as a function of FLOPs can be expressed in two steps. The first step is to find $\hat{\mathbf{s}}^{\text{ML}}$. This step requires $(16N^2 + 8N - 2)J^N$ FLOPs [10]. The second step is to search for $N \log_2 J$ ML antipodal solutions. Since the number of possible ML antipodal solutions for each bit is $J^N/2$, the computational complexity to find all $d_{\nu,\mu}^{\overline{\text{ML}}}$ is $(16N^2 + 8N - 2) \frac{J^N}{2} N \log_2 J$ FLOPs. Therefore, the total MLSE complexity is given by

$$\mathcal{O}_{\text{MLSE}} = J^N \left(\frac{N \log_2 J}{2} + 1 \right) (16N^2 + 8N - 2). \quad (30)$$

MLSE detection complexity is constant among all SNR values and grows exponentially with transmission matrix size and modulation order. Although the nonlinear MLSE detector can find the LLRs and minimize the error probability, its computational complexity is prohibitive, making this detector unfeasible. Therefore, lower computational complexity detectors must be used to find d^{ML} and $\mathbf{d}^{\overline{\text{ML}}}$.

III. SOFT SPHERE DECODING

The design of a soft SD for recovering the LLRs from the received FTN-GFDM signal after the MF is the main contribution of this paper. This detector improves the overall BER performance of the system, as it provides more information for the FEC decoder. The following subsections describe the sphere decoding principles, the proposed algorithm, and the complexity analysis of our solution as compared with other detectors.

A. Sphere Decoding Principles

Traditional ML detectors search for the best candidate sequence from the entire sample space of the modulation scheme, which means comparing the Euclidean distance between the received sequence and all possible transmit sequences. This search mechanism has a prohibitive high complexity when the modulation order, number of subsymbols and number of subcarriers take on standard operation values. One approach to deal with this problem is to exploit sub-optimal detectors. Sphere decoding consists of limiting the search space to a multidimensional sphere with a radius of ρ and searching only for lattice points that are inside the sphere [23]. In order to find the ML and its antipodal solution with affordable complexity, the SD can be employed to solve (29) efficiently. The closest lattice point to the vector received inside the sphere is also the closest lattice point of the entire search space. Thus, the SD can achieve ML optimal¹ performance with lower computational cost than exhaustive search algorithms [24].

The search radius must be large enough to contain the ML solution and to have few lattice points inside it [25]. The search within the sphere can be transformed into a tree search, where a node and its sub-tree are pruned when they can no longer lead to an ML solution update. Every time a leaf node is found, i.e., the first level of the tree search is reached, a lattice point is found inside the sphere, and the ML solution is updated.

The sphere radius in SD is defined according to the AWGN variance. Since the problem in (29) has noise samples colored by MF, a whitening filter is used, resulting in

$$\arg \min_{\hat{\mathbf{s}} \in \mathbb{M}} \|(\mathbf{A}^{\text{H}})^{-1}(\mathbf{r} - \mathbf{G}\hat{\mathbf{s}})\|^2 \leq \rho^2. \quad (31)$$

Considering $\zeta = (\mathbf{A}^{\text{H}}\mathbf{A})^{-1}\mathbf{A}^{\text{H}}\mathbf{r}$, the SD radius can be expressed by

$$(\zeta - \hat{\mathbf{s}})^{\text{H}}\mathbf{G}^{\text{H}}\mathbf{A}^{-1}(\mathbf{A}^{-1})^{\text{H}}\mathbf{A}^{\text{H}}\mathbf{A}(\zeta - \hat{\mathbf{s}}) = (\zeta - \hat{\mathbf{s}})^{\text{H}}\mathbf{G}^{\text{H}}(\zeta - \hat{\mathbf{s}}) = \rho. \quad (32)$$

To simplify the evaluation of the Euclidean norm in (31), the interference matrix \mathbf{G} must be decomposed by the Cholesky factorization [26] given by

$$\text{chol}\{\mathbf{G}^{\text{H}}\} = \mathbf{R}^{\text{H}}\mathbf{R}, \quad (33)$$

where $\mathbf{R} \in \mathbb{C}^{N \times N}$ is an upper triangular matrix. Therefore, the condition in (31) can be expressed as

$$\arg \min_{\hat{\mathbf{s}} \in \mathbb{M}} \|\mathbf{R}(\zeta - \hat{\mathbf{s}})\|^2 \leq \rho^2. \quad (34)$$

As a result of the structure of \mathbf{R} , the inequality (34) can be rewritten as [27]

$$\rho^2 \geq \sum_{\ell=1}^N \left| \sum_{i=\ell}^N R_{\ell,i} (\zeta_i - \hat{s}_i) \right|^2. \quad (35)$$

Note that, because of the triangular structure of \mathbf{R} , the N th element of (35) depends only on \hat{s}_N , the $(N-1)$ th element depends only on $\{\hat{s}_N, \hat{s}_{N-1}\}$, and so on [23], [27]. Thus,

¹In fact, the performance is max-log optimal, since the smallest-vector algorithm operates on the approximated problem of equation (29).

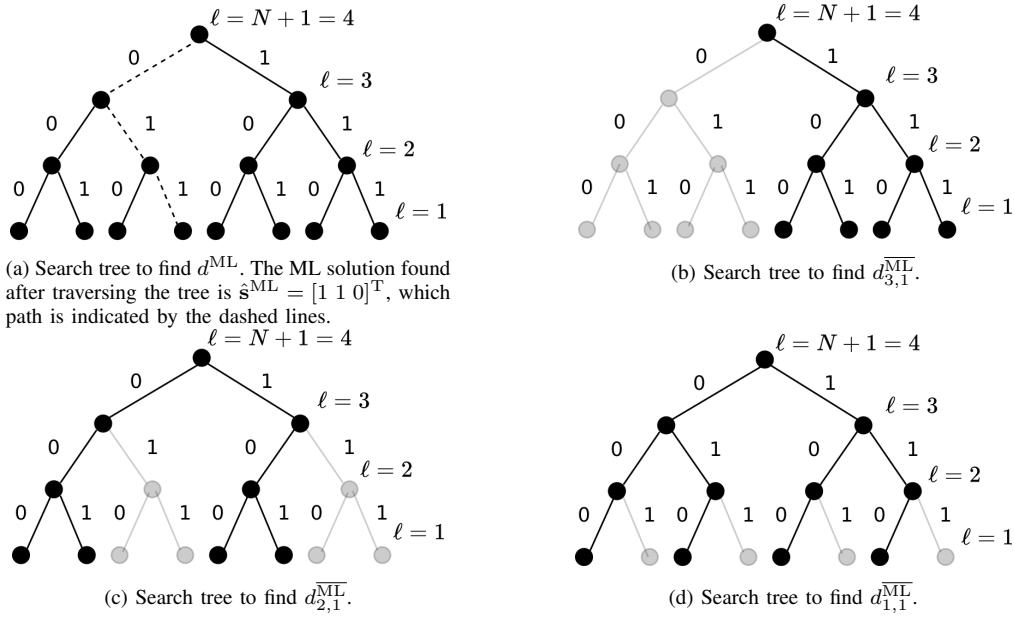


Fig. 4. Search trees considering $N = 3$, BPSK and $\hat{\mathbf{s}}^{\text{ML}} = [1 \ 1 \ 0]^T$.

the search for lattice points within the sphere becomes a tree search.

Since the ML solution can be approximated by (34) in SD, d^{ML} and $d_{\nu,\mu}^{\text{ML}}$ can be rewritten as

$$d^{\text{ML}} = \|\mathbf{R}(\zeta - \hat{\mathbf{s}}^{\text{ML}})\|^2, \quad (36)$$

and

$$d_{\nu,\mu}^{\text{ML}} = \min_{\hat{\mathbf{s}} \in \mathcal{M}_{\nu,\mu}^{\text{ML}}} \|\mathbf{R}(\zeta - \hat{\mathbf{s}})\|^2, \quad (37)$$

respectively.

As with the MLSE detector, to compute the LLRs in (29), the SD approach must first find the ML solution and, then, the minimum Euclidean distances of the ML antipodal solutions. To find d^{ML} , the SD algorithm must explore a search tree with a depth of $N + 1$ levels and J^N nodes at level $\ell = 1$. To find all ML antipodal solutions, $N \log_2 J$ search trees with $N + 1$ levels and $J^N/2$ nodes on level $\ell = 1$ must be examined. Nevertheless, the number of nodes in each tree depends on the searched bit position in $\hat{b}_{\nu,\mu}^{\text{ML}}$. Fig. 4 shows the search trees for $N = 3$ and binary phase-shift keying (BPSK) symbols. The search tree used to determine the ML solution, $\hat{\mathbf{s}}^{\text{ML}}$, is shown in Fig. 4a. Since $\hat{\mathbf{b}}^{\text{ML}} = [1 \ 1 \ 0]^T$ in this example, Figs. 4b, 4c, and 4d show the search tree for each bit of the ML antipodal solution. However, the SD complexity depends on the number of nodes scanned. In the worst case, the SD algorithm must search the entire search tree, and its complexity becomes exponential, approaching that of MLSE. Although the SD algorithm is less complex than the MLSE detector, its computational complexity can be high for some practical applications where the devices are limited in computing power.

B. The Single Tree Search Algorithm

The soft-output single tree search (STS)-SD algorithm proposed in [20] ensures complexity minimization by jointly

searching for the ML solution and the Euclidean distances of the ML antipodal solution. Partial distances are the intermediate distances when traversing levels in the search tree. In this strategy, the underlying branch of a node is only traversed if the partial Euclidean distance $d(\hat{\mathbf{s}})$ of this node leads to: i) An update of the ML solution, i.e., if $d(\hat{\mathbf{s}}) < d^{\text{ML}}$ or; ii) An update of the Euclidean distance of at least one ML antipodal solution, i.e., if $d(\hat{\mathbf{s}}) < d_{\nu,\mu}^{\text{ML}}$. Thus, STS-SD can be employed to reduce complexity, ensuring that the search tree nodes are visited at most once.

The STS-SD algorithm can be divided into two steps: i) List administration and; ii) Tree pruning [20]. The list administration step happens whenever the first level is reached, i.e., a leaf node of the search tree is reached (Algorithm 1 lines 26 to 40). If the Euclidean distance of \mathbf{s} at a leaf node is smaller than the Euclidean distance of the current ML solution, i.e., $d(\hat{\mathbf{s}}) < d^{\text{ML}}$, a new ML solution has been found (Algorithm 1 line 28). Before updating the ML solution with the newly found \mathbf{s} , all $d_{\nu,\mu}^{\text{ML}}$ for which $\hat{b}_{\nu,\mu} = \hat{b}_{\nu,\mu}^{\text{ML}}$ are updated with the previous d^{ML} (Algorithm 1 line 29). This guarantees that for each bit in the ML solution that is changed in the update process, the Euclidean distance of the previous ML solution becomes the Euclidean distance of the new ML antipodal solution. Then, the symbol vector $\hat{\mathbf{s}}^{\text{ML}}$ is assigned the new distance-minimizing vector $\hat{\mathbf{s}}$. Correspondingly, the Euclidean distance d^{ML} is updated to $d(\hat{\mathbf{s}})$ (Algorithm 1 line 30).

However, if the Euclidean distance found in the leaf node is greater than the Euclidean distance of the ML solution, i.e., $d(\hat{\mathbf{s}}) > d^{\text{ML}}$, only the ML antipodal solution is checked. For all ν and μ simultaneously satisfying the bit sequence $\hat{b}_{\nu,\mu}^{\text{ML}}$ and $d(\hat{\mathbf{s}}) < d_{\nu,\mu}^{\text{ML}}$, the Euclidean distances $d_{\nu,\mu}^{\text{ML}}$ must be updated to $d(\hat{\mathbf{s}})$ (Algorithm 1 line 32).

Tree pruning can happen while scanning a node at level ℓ , for $\ell > 1$ (Algorithm 1 lines 8 to 25). The partial vector $\hat{\mathbf{s}}^\ell \in \mathbb{C}^{(N-\ell+1) \times 1}$ at level ℓ (corresponding to the sequence of

symbols $(\hat{s}_1, \dots, \hat{s}_{\ell-1})$ of a sub-tree at level ℓ is defined by the sequence of symbols $(\hat{s}_\ell, \dots, \hat{s}_N)$.

The pruning criterion for \hat{s}^ℓ takes into account all Euclidean distances of the ML antipodal solution that can be updated. First, the corresponding bits of the partial vector, $\mathbf{b}^\ell \in \mathbb{R}^{((N-\ell+1)\log_2 J) \times 1}$, are compared to the corresponding bits of the current ML solution, i.e., $\hat{\mathbf{b}}^{\text{ML}}$.

All Euclidean distances $d_{\nu,\mu}^{\overline{\text{ML}}}$ that correspond to $\hat{b}_{\nu,\mu} = \hat{b}_{\nu,\mu}^{\overline{\text{ML}}}$, $\nu = \ell, \dots, N$, can be updated by searching the sub-tree of \mathbf{s}^ℓ . Similarly, the Euclidean distances $d_{\nu,\mu}^{\overline{\text{ML}}}$, $\nu = 1, \dots, \ell - 1$, corresponding to the bits of node \hat{s}^ℓ can also be updated. Therefore, the set of Euclidean distances that may be updated during the search in the sub-tree underlying the node \hat{s}^ℓ is given by

$$\mathcal{A}(\hat{\mathbf{s}}^\ell) = \left\{ d_{\nu,\mu}^{\overline{\text{ML}}} | (\nu \geq \ell) \wedge (b_{\nu,\mu} = b_{\nu,\mu}^{\overline{\text{ML}}}) \right\} \cup \left\{ d_{\nu,\mu}^{\overline{\text{ML}}} | \nu < \ell \right\}. \quad (38)$$

The node $\hat{\mathbf{s}}^\ell$ and its sub-tree is pruned if the partial Euclidean distance, $d(\hat{\mathbf{s}}^\ell)$, satisfies

$$d(\hat{\mathbf{s}}^\ell) > \max_{a \in \mathcal{A}(\hat{\mathbf{s}}^\ell)} a. \quad (39)$$

If the condition in (39) is not met, the search must continue at level $(\ell - 1)$.

The steps to implement the soft-output STS-SD detector can be seen in Algorithm 1. In order to avoid choosing an initial radius and still lead to an efficient tree pruning, the STS algorithm employs the radius reduction strategy, the main idea of which is to initiate the algorithm with $d^{\text{ML}} \leftarrow \infty$ and update the radius according to $d^{\text{ML}} \leftarrow d(\hat{\mathbf{s}})$ whenever a leaf $\hat{\mathbf{s}}$ has been reached.

The STS-SD pruning criterion ensures that a node and its sub-tree are only scanned if this can result in: i) an update of the ML solution or; ii) at least an update of one of $d_{\nu,\mu}^{\overline{\text{ML}}}$. This approach reduces the computational complexity when compared to other soft-output detection algorithms based on tree search [20], [21].

C. Computational Complexity

The complexity of tree search algorithms is dictated mainly by the number of nodes visited and the operational cost at each node. In SD, the number of nodes visited depends on search radius reduction and, consequently, on the SNR and the conditioning of \mathbf{G} . The operational costs of each node depend on the algorithm. Hence, finding an exact expression for the SD complexity is not a trivial task. For \mathbf{G} with poorly conditioning or low SNR, the complexity of the SD can be exponential and, in the worst-case scenario, can be as high as the complexity of the ML detector.

The complexity of the proposed STS-SD is obtained by evaluating the partial Euclidean distance d_ℓ in line 7 of Algorithm 1. Because \mathbf{R} is an upper triangular matrix, the number of operations required to compute d_ℓ increases as ℓ decreases. Therefore, it is necessary to count the number of visited nodes at the ℓ th level and weight them by the number of operations required to compute d_ℓ . Assuming the worst-

Algorithm 1 Soft-output STS-SD detector

```

1: Input: Upper triangular matrix  $\mathbf{R}$ , unconstrained ML
   estimate  $\zeta$ , search tree depth  $N$ , and number of QAM
   symbols  $J$ 
2:  $d^{\text{ML}} \leftarrow \infty$ ;  $\hat{\mathbf{s}}_{1:N}^{\text{ML}} \leftarrow \mathbf{0}$ ;  $b_{1:NJ}^{\text{ML}} \leftarrow \mathbf{0}$  and  $d_{1:N}^{\overline{\text{ML}}} \leftarrow \infty$ ;
3:  $\varsigma_N \leftarrow 1$ ; (number of nodes visited at  $N$ th level)
4:  $\ell \leftarrow N$ ;
5:  $d_{N+1} \leftarrow 0$ ;
6: while  $\varsigma_N \leq J$  (top level not exhausted) do
7:    $d_\ell \leftarrow d_{\ell+1} + \left| \sum_{i=\ell}^N R_{\ell,i} (\zeta_i - \hat{s}_i) \right|^2$  (compute partial
   Euclidean distance);
8:   if  $\ell > 1$  (non-leaf node) then
9:     if  $d_\ell < d^{\text{ML}}$  then
10:       $\ell \leftarrow \ell - 1$  (go down a level)
11:       $\varsigma_\ell \leftarrow \varsigma_\ell + 1$  (choose a symbol)
12:     else
13:      (Check if there is a smaller Euclidean distance
   for the ML antipodal solution in the sub-tree; find
    $\mathcal{A}(\hat{\mathbf{s}}_\ell)$ )
14:      if  $d_k > \max_{a \in \mathcal{A}(\hat{\mathbf{s}}_\ell)} a$  then
15:         $\ell \leftarrow \ell - 1$  (go down a level)
16:         $\varsigma_\ell \leftarrow \varsigma_\ell + 1$  (choose a symbol)
17:      else
18:        (go right or up to next node)
19:        if  $\varsigma_\ell > J$  (level already exhausted) then
20:           $\varsigma_{1:\ell} \leftarrow \mathbf{0}$ ;  $\ell \leftarrow \ell + 1$ ;  $\varsigma_\ell \leftarrow \varsigma_\ell + 1$ ; (go back to
   upper level and go right to next node)
21:        else
22:           $\varsigma_\ell \leftarrow \varsigma_\ell + 1$ ; (go right to next node)
23:        end if
24:      end if
25:    end if
26:    else
27:      (A leaf node is reached, i.e.,  $\ell == 1$ )
28:      if  $d_\ell < d^{\text{ML}}$  (smaller distance is found; a new ML
   solution is found) then
29:         $\mathbf{d}^{\text{ML}}(\mathbf{b} \neq \mathbf{b}^{\text{ML}}) \leftarrow d^{\text{ML}}$ ; (update the ML antipo-
   dal solution)
30:         $d^{\text{ML}} \leftarrow d_\ell$ ;  $\hat{\mathbf{s}}^{\text{ML}} \leftarrow \hat{\mathbf{s}}$ ;  $\mathbf{b}^{\text{ML}} \leftarrow \mathbf{b}$ ; (update the ML
   solution)
31:      else
32:         $\mathbf{d}^{\text{ML}}(\mathbf{b} \neq \mathbf{b}^{\text{ML}} \text{ and } \mathbf{d}^{\overline{\text{ML}}} > d_\ell) \leftarrow d_\ell$ ; (update the
   ML antipodal solution)
33:      end if
34:      (move right or up to next node)
35:      if  $\varsigma_\ell > J$  (level already exhausted) then
36:         $\varsigma_{1:\ell} \leftarrow \mathbf{0}$ ;  $\ell \leftarrow \ell + 1$ ;  $\varsigma_\ell \leftarrow \varsigma_\ell + 1$ ; (go back to
   upper level and go right to next node)
37:      else
38:         $\varsigma_\ell \leftarrow \varsigma_\ell + 1$ ; (go right to next node)
39:      end if
40:    end if
41:  end while
42: Output: ML solution  $\hat{\mathbf{s}}^{\text{ML}}$  and  $d^{\text{ML}}$ , and the Euclidean
   distances vector of the ML antipodal solution  $\mathbf{d}^{\overline{\text{ML}}}$ 

```

TABLE I
POLAR CODE PARAMETERS.

Parameters	Config. A	Config. B
Coding rate (R_c)	1/2	1/2
Block size (N_c)	1024	2048
Punctured bits	24	8
Decoding algorithm	SC	SC

TABLE II
FTN-GFDM SYSTEM PARAMETERS. BPSK, $P=4$ AND $S=5$.

FTN-GFDM Parameters	Time	Frequency
Prototype filter	Dirichlet	rect
Subsymbol distance factor (v_t)	0.8	1
Subcarrier distance factor (v_f)	1	0.8
Number of subsymbols (M)	5	4
Number of subcarriers (K)	5	6

case scenario, where all nodes in the search tree are visited, the complexity SD is given by

$$\mathcal{O}_{SD} = \sum_{\ell=1}^N J^{(N+1-\ell)} \mathcal{O}_{\ell}, \quad (40)$$

where \mathcal{O}_{ℓ} is the number of operations required to compute d_{ℓ} on a node at the ℓ th level and $M^{(N+1-\ell)}$ is the number of nodes at the same level. Thus, \mathcal{O}_{ℓ} can be expressed in terms of the number of FLOPs:

$$\mathcal{O}_{\ell} = 10(N - \ell) + 12. \quad (41)$$

Substituting (41) into (40) and applying some sum identities [28], the upper bound for the SD complexity is given by

$$\mathcal{O}_{SD} = \frac{2J((5N+1)J^{N+1} - (5N+6)J^N - J + 6)}{(J-1)^2}. \quad (42)$$

The difference in complexity between the the soft-output and the hard-output SD relates to the number of nodes visited. As the soft-output SD requires finding the Euclidean distances of antipodal solutions, the number of nodes visited is higher than in hard-output SD, in which it is only necessary to find the minimum Euclidean distance of the ML solution.

IV. SIMULATION RESULTS

In this section, the simulation results are discussed. Subsection IV-A evaluates the performance in terms of BER, while subsection IV-B analyzes the computational complexity of the implemented algorithms.

The parameters of each block in the communication chain used for performance evaluation are presented in Tables I, II, and III. Table I presents polar code parameters. Two possible configurations (A and B) have been defined. These configurations are used in accordance with the data payload of the FTN-GFDM block, which may change according to the time or frequency compression.

Table II presents the FTN-GFDM system parameters used for performance evaluation. Two different configurations are defined, one being applied for time and the other for frequency compression. The parameters have been chosen to make the spectrum efficiency practically equal for both configurations.

TABLE III
CHANNEL MODELS.

Channel	Impulse response
AWGN	$\mathbf{h}_{AWGN} = \mathbf{1}$
Time-invariant frequency-selective	$\mathbf{h}_{TIFS} = [1 \ 0.4 \ 0.2 \ 0.08]^T$
Time-variant flat	$\mathbf{h}_{TVF} = h \sim \mathcal{CN}(0, 1)$

TABLE IV
APPROXIMATE CODING GAINS FOR SOFT AND HARD OUTPUTS OBTAINED IN FIG. 5.

Channel	Soft $E_b/N_0 @ 10^{-3}$	Hard $E_b/N_0 @ 10^{-3}$	Gain (dB)
AWGN	2.82	4.59	1.77
TIFS	3.47	5.29	1.82
TVF	14.35	17.58	3.23

TABLE V
APPROXIMATE CODING GAINS FOR SOFT AND HARD OUTPUTS OBTAINED IN FIG. 6.

Channel	Soft $E_b/N_0 @ 10^{-3}$	Hard $E_b/N_0 @ 10^{-3}$	Gain (dB)
AWGN	2.47	4.61	2.12
TIFS	3.18	5.18	2
TVF	14.59	17.53	2.94

Table III shows parameters of the simulated channels. In this paper, three channel models are considered. The first is an AWGN channel, with a flat and time-invariant frequency response. The second is a frequency-selective channel with a time-invariant impulse response, while the third is a doubly-dispersive channel, whose impulse response is randomly chosen for each FTN-GFDM block, remaining time-invariant during each block. Perfect channel state information (CSI) is assumed on the receiver side for all channel models, since our goal is to evaluate the performance of the proposed FTN-GFDM detectors without taking into account the losses resulting from channel estimation.

A. BER Performance Analysis

The BER performance of the soft-output SD detector is compared with the hard-output SD considering: i) time-compressed FTN-GFDM and; ii) frequency-compressed FTN-GFDM.

Fig. 5 shows BER versus E_b/N_0 of hard and soft-output SD detectors for time-compressed FTN-GFDM. The waveform parameters are presented in Table II for time-compression and the three channels described in Table III have been considered. The time compression $v_t = 0.8$ introduces intersymbol interference (ISI) and improves the spectrum efficiency by 25%. The uncoded BER performance of the proposed soft-output SD is equal to the theoretical optimum detector. Fig. 5 also displays the polar coded BER performance for soft and hard-output SD. The parametrization of the polar code assumes the values described for Config. A in Table I, because this configuration allows for the codeword to be split in an integer number of FTN-GFDM blocks. Fig. 5 shows that the polar decoder benefits from the soft information since the proposed detector provides significant performance gains when compared with the hard-output SD. Table IV presents the E_b/N_0 required by each detector to achieve the target BER

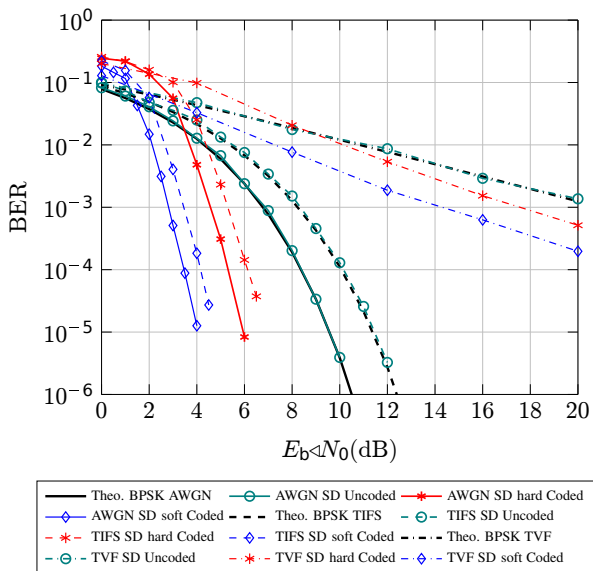


Fig. 5. BER performance for time-compressed FTN-GFDM.

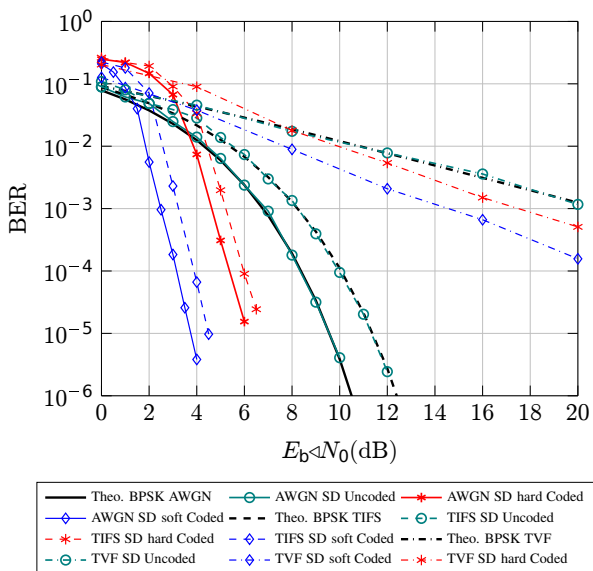


Fig. 6. BER performance for frequency-compressed FTN-GFDM.

of 10^{-3} . It also lists the performance gains of soft over hard outputs.

Fig. 6 presents the BER performance for the frequency-compressed FTN-GFDM. The waveform parameters are those in Table II for frequency-compression. Once again, the channels proposed in Table III have been considered. The polar code with Config. B (see Table I) is employed in this scenario. Fig. 6 shows that the 20% improvement in spectrum efficiency does not imply in any BER performance degradation when the proposed soft-output SD algorithm is used. Also, the proposed detector combined with the polar code is compared with the the polar-coded hard-output SD detector. Again, the polar decoder benefits from the soft output, providing significant BER performance gains, as highlighted in Table V.

 TABLE VI
 COMPUTATIONAL COMPLEXITY COMPARISON BETWEEN HARD- AND SOFT-OUTPUT SD.

Channel	Output	SNR			
		0 dB	5 dB	10 dB	
AWGN	Hard	Nodes	1.65×10^4	2.36×10^3	1.04×10^3
		FLOPs	1.38×10^6	2.16×10^5	8.90×10^4
	Soft	Nodes	1.76×10^5	3.66×10^4	1.57×10^4
		FLOPs	1.03×10^7	2.37×10^6	1.01×10^6
TIFS	Hard	Nodes	2.02×10^4	2.71×10^3	1.08×10^3
		FLOPs	1.67×10^6	2.50×10^5	9.35×10^4
	Soft	Nodes	1.91×10^5	4.11×10^4	1.65×10^4
		FLOPs	1.16×10^7	2.72×10^6	1.07×10^6
TVF	Hard	Nodes	3.01×10^5	8.75×10^4	4.21×10^4
		FLOPs	1.22×10^7	3.81×10^6	1.64×10^6
	Soft	Nodes	5.97×10^5	2.18×10^5	8.24×10^4
		FLOPs	2.52×10^7	9.67×10^6	3.81×10^6

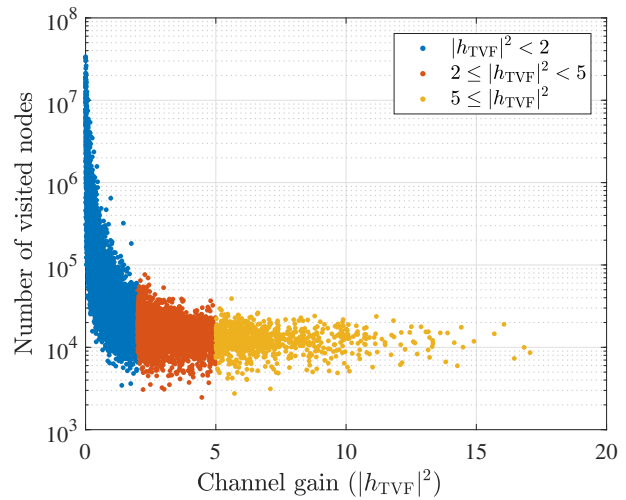


Fig. 7. Number of nodes visited by the STS-SD algorithm versus TVF channel gain, under an average SNR of 5 dB.

B. Computational Complexity Analysis

Table VI presents the mean computational complexity in terms of FLOPs and the mean number of visited nodes for the hard and soft-output SD algorithms for SNR values of 0, 5, and 10 dB. These results are obtained using the frequency-compression system parameters shown in Table II. The complexity is analyzed for the three channels shown in Table III. Feeding the system parameters into (42), the upper limit for the SD complexity is 7,784,628,240 FLOPs, and the total nodes in the tree search is 33,554,430. Assuming an SNR of 10 dB, the hard-output SD complexity represents 8.85% of the soft-output SD complexity for the AWGN channel, 8.75% for the TIFS channel, and 43.15% for the TVF channel. This shows that the soft-output SD detector presents higher computational complexity than the hard-output detector. In the worst case, both can reach the upper bound. However, since the soft-output SD visits more nodes, the probability of it reaching the upper bound is higher.

Table VI shows that the number of visited nodes and the computational complexity decrease with increasing SNR. A 90.19% mean complexity reduction can be observed when comparing the SD soft-output detector for 10 and 0 dB in the AWGN channel. For the TIFS channel, the reduction was 90.78%, and for the TVF channel it was 84.88%. These results

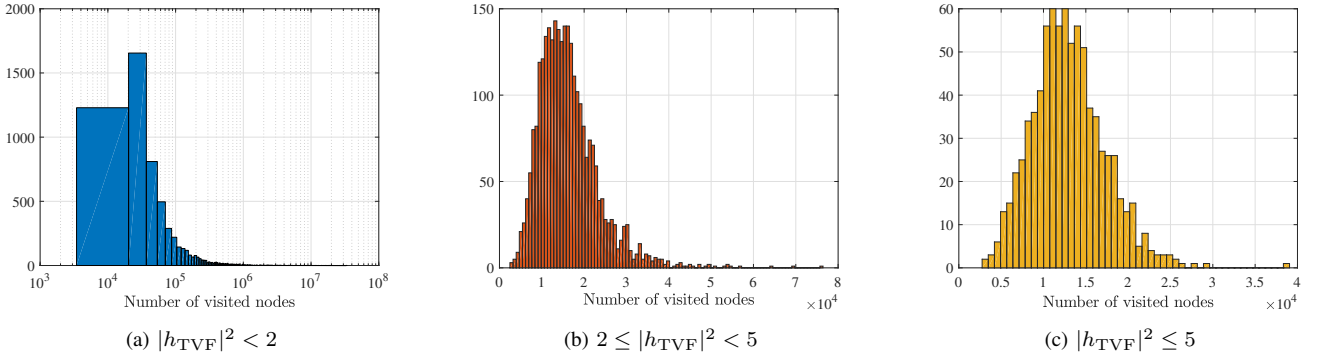


Fig. 8. Histograms of number of nodes visited under different TVF channel gain ranges.

TABLE VII

COMPARISON OF THE MAXIMUM, MINIMUM, AND MEAN NUMBER OF NODES VISITED BY THE STS-SD ALGORITHM UNDER DIFFERENT TVF CHANNEL GAIN RANGES.

Channel Gain	Samples	Number of visited nodes		
		Maximum	Minimum	Mean
$ h_{TVF} ^2 < 2$	6,399	33,554,430	3,450	3.5278×10^5
$2 \leq h_{TVF} ^2 < 5$	2,816	76,434	2,466	1.6303×10^4
$ h_{TVF} ^2 \geq 5$	815	39,064	2,754	1.2997×10^4
$ h_{TVF} ^2$	10,030	33,554,430	2,466	2.3070×10^5

lead to the conclusion that the complexity of the SD soft-output detector might become affordable if the SNR takes on higher values.

Yet, since the soft-output SD detector requires finding the minimum Euclidean distances of the antipodal solutions, the number of visited nodes can be substantially higher than that for the hard-output SD detector. For $\text{SNR} = 0$ dB, the increase in the mean number of visited nodes is 966.85% for the AWGN channel, 845.77% for the TIFS channel, and 98.38% for the TVF channel. The number of nodes visited also increases with the deterioration of the system conditioning, i.e., the increase in distortion caused by the channel. The signal compression also increases the number of visited nodes for the hard and soft-output SD detectors. Therefore, the computational complexity is higher for the TIFS and TVF channels than for the AWGN channel.

Fig. 7 shows the number of nodes visited by the STS-SD algorithm as a function of the TVF channel gain, assuming an average SNR of 5 dB and the system parameters for frequency compression listed in Table II. As described in Table III, the TVF channel follows a complex Gaussian distribution; therefore, its power gain has an exponential distribution. Thus, most channel gain values concentrate on low values, close to zero. Additionally, histograms of visited nodes were generated for different ranges of channel gain values, as shown in Fig. 8. To improve the visualization of the visited nodes distribution, the axis related to the number of visited nodes in Fig. 8a is on a logarithmic scale. Table VII summarizes the maximum, minimum, and mean values of visited nodes obtained for each channel gain range. For gains less than 2, the maximum number of nodes in the tree was reached. This indicates that the TVF channel, combined with noise, compromised the FTN-GFDM signal to the extent that it required the STS-SD algorithm to traverse the entire search

tree. The joint analysis of Figs. 7 and 8, and Table VII provides a comprehensive understanding of the behavior of the STS-SD algorithm concerning the TVF channel gain. In particular, it can be observed that lower channel gains result in a more extensive exploration of the search tree, indicating a significant sensitivity to the channel conditions. It is important to note that a TVF channel can lead to an outage state on the receiver side. In this scenario, the STS-SD will traverse many nodes but with a low probability of detecting the transmitted signal. In a practical system, this situation can be avoided by declaring an outage and dropping the content of the FTN-GFDM block, thereby reducing complexity and energy consumption.

V. CONCLUSION

In scenarios where massive MIMO and UDN are not feasible, spectral efficiency can be increased by the choice of the waveform. The FTN-GFDM waveform is flexible and capable of improving spectral efficiency over conventional GFDM and of increasing the data rate by compressing subsymbols or subcarriers without BER performance loss. The results obtained in this paper show that FTN-GFDM combined with polar codes improves waveform spectrum efficiency without introducing significant performance loss.

The SD detector can achieve BER performance similar to that achieved by the optimal detector but with less complexity and polar codes can be combined with FTN-GFDM to provide gains in BER performance with affordable complexity, improving the overall system performance with both soft and hard output SD detectors.

The soft-output SD outperforms the hard-output SD at the cost of the visitation of more nodes during tree search and, hence, higher computational complexity. Although more complex, the soft-output SD detector can have an affordable complexity mainly if the available SNR can assume high values. Thus, soft-output SD can be an interesting solution in scenarios that prioritize high spectral efficiency but the conventional high efficient techniques cannot be exploited. Also, when the complexity of soft-output cannot be supported, hard-output SD can be a good compromise. Therefore, SD allows implementations in practical systems with flexible trade-offs between complexity and performance. In future work, we intend to analyze the integration of the soft decision SD

with other state-of-the-art FEC schemes, such as low density parity check (LDPC), to achieve the best BER performance over several channel conditions. We also aim to develop other detectors that are sub-optimal and, therefore, cannot achieve the same BER performance as STS-SD or MLSE but have a very low complexity compared to these approaches.

REFERENCES

- [1] G. Liu *et al.*, "Vision, requirements and network architecture of 6G mobile network beyond 2030," *China Commun.*, vol. 17, no. 9, pp. 92–104, 2020.
- [2] L. Mendes *et al.*, "Enhanced remote areas communications: The missing scenario for 5G and beyond 5G networks," *IEEE Access*, vol. 8, pp. 219859–219880, 2020.
- [3] N. Michailow *et al.*, "Generalized frequency division multiplexing for 5th generation cellular networks," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3045–3061, Sept 2014.
- [4] D. Zhang, M. Matthé, L. L. Mendes, and G. Fettweis, "A study on the link level performance of advanced multicarrier waveforms under MIMO wireless communication channels," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2350–2365, 2017.
- [5] M. Matthé *et al.*, "Precoded GFDM transceiver with low complexity time domain processing," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, p. 138, 2016.
- [6] A. Nimr, M. Chafii, M. Matthe, and G. Fettweis, "Extended GFDM framework: OTFS and GFDM comparison," in *Proc. IEEE GLOBE-COM*, 2018.
- [7] J. E. Mazo, "Faster-than-Nyquist signaling," *Bell Syst. Technical J.*, vol. 54, no. 8, pp. 1451–1462, 1975.
- [8] J. B. Anderson, F. Rusek, and V. Öwall, "Faster-than-Nyquist signaling," *Proc. IEEE*, vol. 101, no. 8, pp. 1817–1830, 2013.
- [9] M. Mello, L. Mendes, and T. Barbosa, "Spectrum efficient GFDM based on faster than Nyquist signaling," *JCIS*, vol. 35, no. 1, pp. 349–356, 2020.
- [10] M. B. Mello and L. L. Mendes, "Low-complexity detection algorithms applied to FTN-GFDM systems," *IEEE Access*, vol. 10, pp. 101683–101696, 2022.
- [11] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [12] E. Şaşoğlu, E. Telatar, and E. Arikan, "Polarization for arbitrary discrete memoryless channels," in *Proc. IEEE IT Workshop*, 2009.
- [13] H. Gamage, N. Rajatheva, and M. Latva-aho, "Channel coding for enhanced mobile broadband communication in 5G systems," in *Proc. EuCNC*, 2017.
- [14] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of polar code constructions for the AWGN channel," 2015, *arXiv:1501.02473*.
- [15] R. Wang and R. Liu, "A novel puncturing scheme for polar codes," *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2081–2084, 2014.
- [16] D. Hui, S. Sandberg, Y. Blankenship, M. Andersson, and L. Grosjean, "Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE," *IEEE Veh. Technol. Mag.*, vol. 13, no. 4, pp. 60–69, 2018.
- [17] K. Chen, K. Niu, and J. R. Lin, "List successive cancellation decoding of polar codes," *Electron. Lett.*, vol. 48, no. 9, p. 500, 2012.
- [18] C. Condo, F. Ercan, and W. J. Gross, "Improved successive cancellation flip decoding of polar codes based on error distribution," in *Proc. IEEE WCNC Workshop*, 2018.
- [19] M. Matthé *et al.*, "Precoded GFDM transceiver with low complexity time domain processing," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, p. 138, 2016.
- [20] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, 2008.
- [21] C. Studer, M. Wenk, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Performance and implementation aspects," in *Proc. ACSSC*, 2006.
- [22] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.
- [23] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, 2005.
- [24] J. Fink, S. Roger, A. Gonzalez, V. Almenar, and V. M. Garcia, "Complexity assessment of sphere decoding methods for MIMO detection," in *Proc. IEEE ISSPIT*, 2009.
- [25] F. Zhao and S. Qiao, "Radius selection algorithms for sphere decoding," in *Proc. C3S2E*, 2009.
- [26] N. J. Higham, "Cholesky factorization," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, pp. 251–254, 09 2009.
- [27] E. Bedeer, H. Yanikomeroglu, and M. H. Ahmed, "Reduced complexity optimal detection of binary faster-than-Nyquist signaling," in *Proc. ICC*, 2017.
- [28] K. H. Rosen, J. G. Michaels, J. L. Gross, J. W. Grossman, and D. R. Shier, Eds., *Handbook of Discrete and Combinatorial Mathematics*, 2nd ed. CRC Press, 2010.



Mariana B. Mello received the B.Sc. and M.Sc. degrees in Electrical Engineering from the National Institute of Telecommunications (Inatel), Brazil in 2015 and 2019, respectively. Today she is pursuing her Ph.D. degree at Inatel in the field of Artificial Intelligence applied to mobile communication systems. Her research interests include multi-carrier modulation techniques for futures wireless systems, sphere decoder algorithms, and artificial intelligence schemes applied to communications.



Luciano L. Mendes received the B.Sc. and M.Sc. degrees from Inatel, Brazil, in 2001 and 2003, respectively, and the Doctor degree from Unicamp, Brazil, in 2007, all in Electrical Engineering. Since 2001, he has been a Professor with Inatel. He has led several research projects funded by FAPEMIG, FINEP, and BNDES. From 2013 to 2015, he was a Visiting Researcher with the Technical University of Dresden in the Vodafone Chair Mobile Communications Systems, where he has developed his postdoctoral program sponsored by CNPq. Since 2015, he has acted as the Research Coordinator of the Radiocommunication Reference Center with Inatel, which is a research project aiming at addressing the main challenges for the Brazilian society regarding mobile, satellite, and terrestrial communications. His main research interests include solutions for the physical layer of the future fifth generation of mobile communication systems (5G), a research field where he has published several papers in the last years. He has also been a CNPq level 2 Research Fellow since 2016. In 2017, he was elected Research Coordinator of the 5G Brazil Project, an association involving industries, telecom operators, and academia which aims for funding and build an ecosystem toward 5G, improving the discussions about the Brazilian needs for this network and how Brazil can contribute with the international standardization.



Daniely G. Silva received her B.Sc. and M.Sc. degrees from Inatel, Brazil, in 2007 and 2013, respectively, and her Ph.D. degree from Unifei, Brazil, in 2019, all in Electrical Engineering. Since 2007, she has worked in projects related to digital communication systems and microelectronics. Currently, she is Researcher in Radiocommunication Reference Center where she has been working with research and development for 5G/6G networks and Open RAN architecture. Also, she is a professor at Inatel in the software defined radio subject.



Paulo Ricardo B. Silva received his B.Sc. in Electrical Engineering in 2012 from Instituto Mauá de Tecnologia, Brazil, and his M.Sc. degrees and Ph.D. degrees in Forward Error Correction from the Federal University of Santa Catarina, Brazil, in 2015 and 2019, respectively. He did his PostDoc in SAR (synthetic aperture radars) for ultrawideband images at the ITA, Brazil, in particular in change detection using information theoretical distances. He now works at CPQD, Brazil, as a Machine Learning Researcher, especially in the fields of intelligent networks with applied artificial intelligence and forecasting.



Tiago C. Barbosa received the B.Sc. degree in Electrical Engineering from Pontifical Catholic University, Pocos de Caldas, MG, Brazil, in 2007 and M.Sc in microelectronics from Federal University of Itajuba, Itajuba, MG, Brazil in 2010. Currently, he is Researcher in National Institute of Telecommunications in Brazil. His research interests are VLSI circuit architecture design, specifically in error correcting codes.