Article

Exploration of Channel Coding Techniques in OTFS Systems: Convolutional, Turbo, LDPC, and Polar Codes

Zehao Li [1], Yi Gong [2,*] , Xinru Li [1], Quan Wang [2] and Zhan Xu [1]

[1] Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University, China; zehao.li@bistu.edu.cn (Z.L.), xinru.li@bistu.edu.cn (X.L.), xuzhan@bistu.edu.cn (Z.X.)
[2] Key Lab of Semiconductor Materials Science, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China; wangquan@semi.ac.cn
* Correspondence: gongyi@bistu.edu.cn

Received: 9 May 2024; Revised: 18 June 2024; Accepted: 20 June 2024; Published: 2 July 2024

Abstract: Orthogonal Time Frequency Space (OTFS) modulation has been verified in high-mobility scenarios as an effective solution to address the Doppler shift effect. To further improve the data transmission efficiency of the system, this paper delves into different channel coding techniques in the Delay-Doppler (DD) domain under OTFS modulation. This paper evaluates the system performance of various encoding schemes under multiple user mobility rates, including Convolutional codes, Turbo codes, Low-Density Parity-Check (LDPC) codes, and Polar codes. Additionally, this paper investigates the impact of adopting different modulation constellation mapping schemes on the system's Bit Error Rate (BER) and explores strategies for enhancing system performance in high-speed data transmission scenarios. The simulation results show that the code-based system reduces the BER by about 17–35% compared to the uncoded OTFS system. In this case, the LDPC code system has a 10 dB Signal-to-Noise Ratio (SNR) gain. The simulation results demonstrate that combining coding techniques with OTFS modulation can significantly enhance the performance of communication systems in highly dynamic environments, with LDPC and Turbo codes showing notable advantages in performance improvement. The findings of this paper not only highlight the importance of choosing the right coding scheme and provide valuable references for the design of high-speed mobile communication systems in the DD domain.

Keywords: OTFS; channel coding; convolutional coding; turbo coding; LDPC coding; polar coding

1. Introduction

With the commercial deployment of 5G communication technology and the preliminary exploration of 6G research [1], data transmission efficiency in high-speed mobile environments is critical. Scenarios such as urban high-speed trains, Telematics, and Unmanned Aerial Vehicles (UAVs) all face the challenge of failing to receive signals as expected due to high-speed movement. Specifically, the Doppler shift phenomenon and signal attenuation are caused by excessive speed [2].

Channel coding and modulation techniques are key technologies at the physical layer. These methods jointly dictate the efficiency and reliability of information transmission in channels under restrictive conditions. Compared to traditional Orthogonal Frequency Division Multiplexing (OFDM), Orthogonal Time Frequency Space (OTFS) modulation offers significant advantages in handling the Doppler effects caused by high-speed mobility [3]. OTFS modulation exploits the Delay-Doppler (DD) domain to map signals, achieving a two-dimensional decomposition of the channel, effectively utilizing the sparse nature of wireless channels. This mapping allows each information symbol to be uniformly distributed across the entire time-frequency resource. Consequently, each symbol experiences the full variation of the channel, enhancing the system's robustness to Doppler effects and providing consistent reception quality in rapidly changing channel conditions [4].

Existing research discusses the impact of different OTFS modulation techniques on channel estimation. For instance, researchers proposed an uplink OTFS-SCMA (Sparse Code Multiple Access) channel estimation technique based on Convolutional Sparse Coding (CSC) under the uplink scenario [5]. This technique enhances channel estimation accuracy by exploiting the properties of standard propagation channel models and the CSC

model, transforming the unknown path sparsity into known user sparsity. However, discussions on different coding techniques in OTFS are relatively less common. An OTFS modulation and signal detection scheme based on Autoencoders (AE), optimized through deep learning for the transmission and reception processes, was developed, as detailed in [6]. This approach involves partitioning the DD grid into multiple sub-blocks and associating a one-hot encoding vector with each sub-block to reduce coding complexity. [7] and [8] respectively showcase the performance characteristics and application scenarios of LDPC-OTFS encoding and modulation schemes and non-binary LDPC-OTFS encoding and modulation schemes.

These two papers provide valuable insights on improving system reliability in high-mobility communication environments and analyze the utility and efficiency of their respective schemes in these scenarios. However, [7] and [8] each evaluate only one type of encoding scheme, making it impossible to derive optimal coding characteristics under OTFS from these works, or to provide additional practical clues for assessing OTFS coding efficiency. Therefore, the existing research is insufficient for studying data transmission efficiency in high-mobility scenarios. Addressing this issue, this work evaluates the impact of convolutional [9], turbo [10], LDPC [11], and polar [12] codes on OTFS performance and provides specific recommendations.

The rest of this paper is structured as follows: Section 2 introduces the OTFS system model with an encoding module. Section 3 details the coding principles of convolutional, turbo, LDPC, and polar codes and their application in OTFS. Section 4 conducts simulations to study the impact of these codes on OTFS performance. Conclusions are drawn in Section 5.

2. System Model

In this section, we introduced the OTFS model with an encoding module. OTFS modulation can effectively cope with the effects of high-speed mobile environments and time-varying multipath conditions by coding in the DD domain. This modulation technique is particularly well-suited for dynamically complex communication scenarios. Additionally, coding techniques significantly improve the spectral efficiency and link quality of communication systems, overcoming the adaptability issues of traditional modulation in high-speed scenarios. Therefore, this paper focuses on OTFS modulation under such conditions for an in-depth discussion, aiming to demonstrate its performance advantages in practical applications further.

In scenarios involving high-mobility vehicles or drones, the signal experiences Doppler shifts that significantly impact the performance of communication systems. Therefore, the impact of different channel codes on the OTFS system is evaluated under a time-frequency dual-selective channel capable of simulating high dynamic scenarios, as proposed in [13].

Figure 1 shows an OTFS model with an encoding module. At the transmitter, the message sequence u first undergoes Cyclic Redundancy Check (CRC) processing, is sent to the channel coder, and then undergoes constellation mapping. Additionally, a pre-processing block is incorporated into the transmitter before the conventional OTFS modulator, while a post-processing block is integrated after the conventional demodulator. Then, the received signal is demodulated through constellation mapping and sent to the corresponding channel decoder for decoding and decision output, followed by a final CRC check.

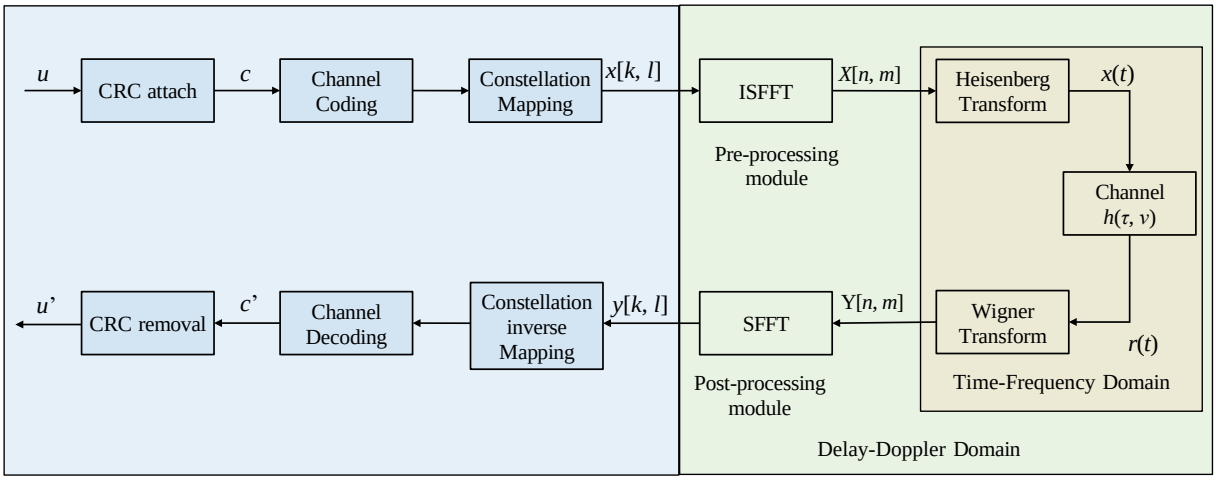

Figure 1. OTFS system model with channel encoding.

Specifically, a predefined generator polynomial is appended to the end of the message sequence u, and then the data block containing the CRC is sent to a specific channel encoder to obtain the encoded sequence c. This is

followed by constellation mapping to obtain the delay-Doppler data symbols $x[k, l]$, where $k = 0, 1, \ldots, N-1$ and $l = 0, 1, \ldots, M-1$. These transmit symbols are multiplexed in an OTFS frame of size $N \times M$, where N and M represent the number of resource units along the Doppler and delay dimensions, respectively.

The pre-processing module consists of the Inverse Symplectic Finite Fourier Transform (ISFFT) and the transmit window function $W_{tx}[n, m]$, and maps the transmit symbols in the DD domain to the time-frequency domain $X[n, m]$. Where the ISFFT is expressed as

$$X[n,m] = \frac{1}{\sqrt{MN}} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} x[k,l] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)} \tag{1}$$

where $n = 0, \ldots, N-1$ and $m = 0, \ldots, M-1$. Next, the time-frequency signal is obtained by multiplying the transmit window function $W_{tx}[n, m]$ by the signal $x[k, l]$ in

$$X[n,m] = Wtx[n,m] \cdot \frac{1}{MN} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} x[k,l] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)} \tag{2}$$

After pre-processing, the time-frequency domain symbol $X[n, m]$ is mapped to the time-domain signal $x(t)$ by the Heisenberg transform, where $g_{tx}(t)$ is a transmit shaping filter with a sampling period of T and a frequency interval of Δf. The Heisenberg transform is expressed as

$$x(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} X[n,m] g_{tx}(t - nT) e^{j2\pi\Delta f(t-nT)} \tag{3}$$

At the receiver, the signal first passes through the double-selective channel $h(\tau, \nu)$ denoted by

$$y(t) = \iint h(\tau,\nu) x(t-\tau) e^{j2\pi\nu(t-\tau)} \mathrm{d}\nu \mathrm{d}\tau + w(t) \tag{4}$$

where $y(t)$ denotes the received signal and $w(t)$ denotes the additive noise at the receiver. $h(\tau, \nu)$ represents the impulse response of the DD domain channel as

$$h(\tau,\nu) = \int h(\tau,t) e^{j2\pi\nu t} \mathrm{d}t \tag{5}$$

Then, time-frequency demodulation is performed using the Wigner transform. First, the mutual ambiguity function is determined as

$$A_{g_{rx},y}(t, f) = \int g_{rx}^{*}(t-t') r(t') e^{-j2\pi f(t-t')} \mathrm{d}t' \tag{6}$$

where $g_{rx}(t)$ is the receive pulse-shaping filter and $*$ represents the complex conjugate operation. Then, the Wigner transformed mutual ambiguity function is sampled at the specified time and frequency points $(nT, m\Delta f)$ as

$$Y[n,m] = A_{g_{rx},y}(t, f)\big|_{t=nT,\, f=m\Delta f} \tag{7}$$

where $n = 0, \ldots, N-1$ and $m = 0, \ldots, M-1$.

The post-processing module maps the time-frequency domain signal to the DD domain signal and performs demodulation. It is realized by the receiver window function $W_{rx}[n, m]$ and Symplectic Finite Fourier Transform (SFFT), where the windowed time-frequency signal $Y[n, m]$ is expressed as

$$Y[n,m] = W_{rx}[n,m] \cdot A_{g_{rx},y}(t, f)\big|_{t=nT,\, f=m\Delta f} \tag{8}$$

Subsequently, the DD domain signal is demodulated by SFFT transform as

$$y[k,l] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} Y[n,m] e^{-j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)} \tag{9}$$

where $y[k, l]$ is the received signal, which is then mapped by the demodulation constellation and sent to the channel decoder for decoding judgment output. Finally, a CRC check is performed to ensure the integrity and accuracy of the transmitted data.

3. Channel Coding in OTFS System

To fully utilize the bandwidth utilization in high-mobility scenarios [14], researchers should combine channel coding with OTFS systems. This subsection first describes the application and impact of channel coding on OTFS systems and then describes the coding and decoding details.

3.1. Convolutional Codes

3.1.1. Introduction to Convolutional Codes

Convolutional codes encode the input data sequence through a fixed set of encoders to generate a sequence of code words with redundancy. The encoders consist of shift registers and polynomials that perform a polynomial product operation on the input sequence to generate the output. This technique provides multiple temporal and spectral replicas for each data symbol in an OTFS system, enhancing resistance to time-frequency variations. Specifically, convolutional codes' redundancy and repetition characteristics enable the mapping of time-frequency units onto the OTFS grid, thus counteracting the Doppler effects in highly dynamic environments.

3.1.2. Encoding

Taking a (2,1,2) convolutional encoder as an example, as shown in Figure 2, if a new information element $\mathbf{u}_i$ is inputted into the encoder for each unit of time, and the data in the memory is shifted one bit to the right, then $\mathbf{u}_i$ and the previous two information elements $\mathbf{u}_{i-1}$ and $\mathbf{u}_{i-2}$ inputted at each time unit is operated according to the rules determined in the figure to obtain the two checksums $\mathbf{c}_{i,1}$ and $\mathbf{c}_{i,2}$ at this time, forming a codeword $\mathbf{c}_i = (\mathbf{c}_{i,1}, \mathbf{c}_{i,2})$ and sent to the channel:

$$\mathbf{c}_{i,1} = \mathbf{u}_i + \mathbf{u}_{i-2} \tag{10}$$

$$\mathbf{c}_{i,2} = \mathbf{u}_i + \mathbf{u}_{i-1} + \mathbf{u}_{i-2} \tag{11}$$

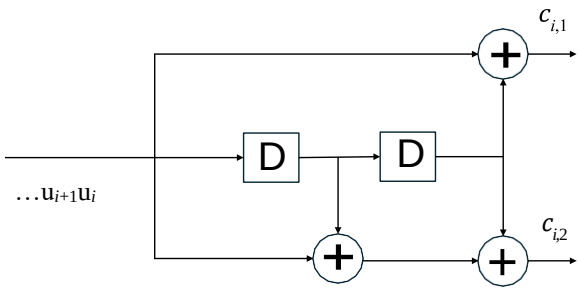

Figure 2. Convolutional encoder (2,1,2).

The input information element for the next time unit is $\mathbf{u}_{i+1}$, and its corresponding two checksums are

$$\mathbf{c}_{i+1,1} = \mathbf{u}_{i+1} + \mathbf{u}_{i-1} \tag{12}$$

$$\mathbf{c}_{i+1,2} = \mathbf{u}_{i+1} + \mathbf{u}_i + \mathbf{u}_{i-1} \tag{13}$$

Forming the second codeword $\mathbf{c}_{i+1} = (\mathbf{c}_{i+1,1}, \mathbf{c}_{i+1,2})$ and input it into the channel, and so on.

3.1.3. Decoding

Convolutional code decoding includes the Viterbi Algorithm [15] and the Bahl, Cocke Jelinek and Raviv (BCJR) algorithm. Here is a brief introduction to the Viterbi algorithm. Assuming that the channel outputs sequence R, the decoder outputs sequence C, and all possible codewords C_i ($i = 1, 2, \ldots, 2^k$).

Maximum likelihood decoding is a decoding rule that aims to select the most probable codeword C_i from among 2^k codewords such that $P(C_i)$ is maximized if the received message R is known. where $P(C|R)$ is called the likelihood function. If the probability of sending each codeword $P(C_i)$ is the same and $P(R)$ is independent of the decoding method, then there is a Bayesian formula:

$$P(C_i|R) = \frac{P(C_i)P(R|C_i)}{P(R)} \tag{14}$$

$$\max P(C_i \mid R) \Rightarrow \max P(R \mid C_i) \tag{15}$$

Since x is monotonically related to $\log_b x$, the maximum likelihood function can be expressed as

$$\max_{i=1,2,\ldots,2^k} \log_b P(R \mid C_i) \tag{16}$$

$\log_b P(R|C_i)$ represents the logarithmic likelihood function.

3.2. Turbo Codes

3.2.1. Introduction to Turbo Codes

Turbo codes enhance the randomness and error-correcting capabilities of encoded data through serial convolutional encoders and interleaving techniques. The interleaver effectively disperses error bits. In the OTFS system, the iterative decoding feature of this forward error correction coding allows for a more precise allocation of time-frequency resources with each iteration. It effectively combats the Doppler effect and selective time-frequency fading in the channel. This method gradually boosts error correction performance by exchanging soft information during the decoding process [16].

3.2.2. Encoding

Figure 3 shows the coding process of Turbo. The input information sequence is the same for both of the component codes. The information sequence $\mathbf{u}_k$ is transmitted directly to the multiplexer as the system output $\{x_k^S\}$ while being fed to the first recursive systematic convolutional encoder (RSC1) for encoding. The interleaved sequence $\{\mathbf{u}_n\}$ obtained by $\{\mathbf{u}_k\}$ passing through interleaver I is fed into the RSC2. Where $n = I(k)$, $0 \le n, k \le N - 10$. $I(\cdot)$ is the interleaving mapping function. The check sequences of RSC1 and RSC2 are $\{x_k^{1P}\}$ and $\{x_k^{2P}\}$, respectively, which are then passed through the puncture matrix to obtain $\{x_k^P\}$. Next multiplexed with the $\{x_k^s\}$ to form the codeword sequence $\{\mathbf{c}_k\}$.

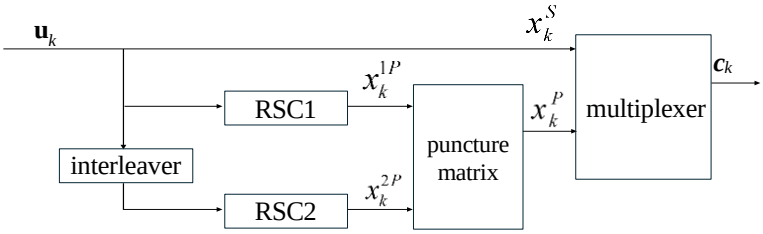

Figure 3. Turbo codes encoder structure.

3.2.3. Decoding

Figure 4 illustrates a typical iterative decoding structure for Turbo codes, where the decoder uses the same interleaver as the encoder, and the deinterleaver serves as the inverse of the interleaving process. Let the soft information received by the decoder of soft input/output decoding (SISO) be ($y_{1,s}$, $y_{2,s}$, $y_{3,s}$, . . ., $y_{N,s}$), the soft information of the check sequence is ($y_{1,p}$, $y_{2,p}$, $y_{3,p}$, . . . , $y_{N,p}$). The prior information L_a is

$$L_a(\mathbf{u}_k) = \ln \frac{p(\mathbf{u}_k = 1)}{p(\mathbf{u}_k = 0)} \tag{17}$$

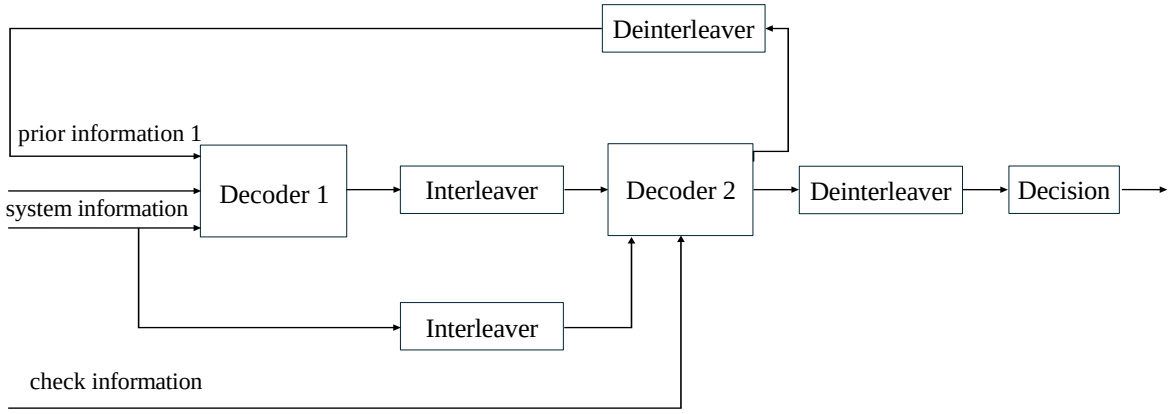

Figure 4. Iterative decoding structure of Turbo codes.

The posterior probability log-likelihood ratio (LLR) $L(\mathbf{u}_k)$ is

$$L(\mathbf{u}_k) = \ln \frac{p(\mathbf{u}_k = 1 \left| y_1^N \right.)}{p(\mathbf{u}_k = 0 \left| y_1^N \right.)} \tag{18}$$

The system information, check information, and prior information of Turbo codes are processed by the component decoder and interleaving, and then enter the next iteration operation. When the number of iterations is completed or other decision conditions are met, the final decoded output sequence is obtained by de-interleaving and decision.

3.3. LDPC Codes

3.3.1. Introduction to LDPC Codes

LDPC codes utilize a sparse parity-check matrix to construct the codewords, and this matrix design permits efficient coding, easy decoding processes, and a high degree of parallel processing. LDPC codes are well suited for highly dynamic environments [17], and are therefore well suited for OTFS modulation systems. It reduces the resource consumption in the signal modulation and coding phases and can adapt to high Doppler shift and time-varying channel conditions. This provides robust error correction performance and ensures data integrity and accuracy in the presence of multipath propagation and channel fading.

3.3.2. Encoding

LDPC encoding is completed through the parity-check matrix **H** and the generator matrix **G**. For an (n, k) LDPC code with a code length of n, a number of information bits k, and a number of redundant bits m ($m = n - k$), the **H** matrix can be transformed into a systematic form through Gaussian elimination:

$$\mathbf{H}_{m \times n} = \left[\mathbf{I}_m \mathbf{P}_{m \times k} \right] \tag{19}$$

where $\mathbf{I}_m$ is an m identity matrix. Based on duality, the generator matrix can be expressed as

$$\mathbf{G}_{k \times n} = \left[\mathbf{P}_{k \times m}^T \mathbf{I}_k \right] \tag{20}$$

By employing the generator matrix for encoding, the encoded codeword $\mathbf{c}_{1 \times k}$ can be expressed as

$$\mathbf{c}_{1 \times k} = \mathbf{u}_{1 \times k} \cdot \mathbf{G}_{k \times n} \tag{21}$$

3.3.3. Decoding

LDPC decoding algorithms are typically categorized into two main types according to the decision method used: hard decision decoding and soft decision decoding [18]. This paper utilizes soft decision algorithms: Belief Propagation (BP) and Sum-Product Algorithm (SPA). The SPA algorithm is briefly described here. Define $N(j)$ as the set of variable nodes, each of which is connected to the check node VN_j , and $N(i)$ as the set of check nodes, each of which is connected to the variable node CN_i. During decoding, the message passed from VN_j to CN_i is $L_{j \to i}$, and the message from CN_i to VN_j is $L_{i \to j}$. Here, L_j is the LLR calculated based on the channel received value

y_j. L_j^{total} is the total LLR for each symbol at the end of each iteration, and $\hat{\upsilon}_j$ is the decoding result after each iteration.

Sum-Product Algorithm

Step 1: Initialization L_j

Step 2: Update of CNs:

$$L_{i\to j} = 2\tanh^{-1}\left(\prod_{j'\in N(i)-(j)} \tanh\left(\frac{1}{2}L_{j\to i}\right)\right) \tag{22}$$

Step 3: Update of VNs:

$$L_{j\to i} = L_j + \sum_{i'\in N(j)-(i)} L_{i'\to j} \tag{23}$$

Step 4: Calculate the total LLR:

$$L_j^{\text{total}} = L_j + \sum_{i\in N(j)} L_{i\to j} \tag{24}$$

Step 5: Test for Stopping Criteria:

$$\hat{\upsilon}_j = \begin{cases} 1, & L_j^{\text{total}} < 0 \\ 0 & \text{else} \end{cases} \tag{25}$$

3.4. Polar Codes

3.4.1. Introduction to Polar Codes

The principle of Polar codes encoding utilizes channel polarization to divide channels into high and low-capacity groups [19]. Information bits are allocated to the high-capacity "good channels" to ensure reliable transmission, while known frozen bits are placed on the low-capacity "bad channels". A key aspect of Polar codes is their ability to optimize the encoding strategy based on channel conditions. This is especially useful in OTFS systems, where the specific delay and Doppler effects are considered to select the most suitable subset of channels for transmission, significantly enhancing the accuracy and efficiency of data transfer.

3.4.2. Encoding

The encoding process of Polar codes involves placing the information bits into the "good channels" that have been identified after channel polarization, while the predetermined frozen bits, typically set to zero, are placed into the "bad channels." This forms the encoding input sequence $u_1^N = \{u_A, u_{Ac}\}$, where A represents the channel indices where information bits are placed. The codeword $x_1^N = u_1^N G_N$ is then obtained by multiplying the input sequence u_1^N by the generator matrix G_N.

In practical encoding, the generator matrix for Polar codes is generated as follows

$$G_N = B_N F^{\otimes n} \tag{26}$$

where B_N denotes bit reversal. The codeword can be obtained by multiplying the bit-reversed sequence with $F^{\otimes n}$.

3.4.3. Decoding

Arikan introduced the Polar codes decoding algorithm, the Successive Cancellation (SC) decoding algorithm. During the polarization process, the polarization of subsequent channels depends on their preceding channels. During decoding, the polarized channel indexed by i is denoted as

$$W_N^{(i)}\left(y_1^N, u_1^{i-1}\right) \triangleq \sum_{u_{i+1}^N \in X_{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N \left| u_1^N\right.) \tag{27}$$

where y_1^N represents the received signal, and u_1^{i-1} denotes the sequence preceding the bit. The estimation $\hat{\boldsymbol{u}}_1$ can be determined based on

$$\hat{u}_i = \begin{cases} h_i(y_1^N, \hat{u}_1^{i-1}), & i \in A \\ u_i & i \in A^c \end{cases} \tag{28}$$

If i belongs to A^c, this means the bit is frozen and is automatically assigned the predetermined frozen bit value. Conversely, if i is in A, it signifies that the bit is an information bit, with the decision being made by

$$hi(y_1^n, \hat{u}_1^{i-1}) = \begin{cases} 0, & L_1^{(i)}(y_1^N, \hat{u}_1^{(i-1)}) \geq 0 \\ 1, & L_1^{(i)}(y_1^N, \hat{u}_1^{(i-1)}) < 0 \end{cases} \tag{29}$$

Next, define the LLR, where the value of LLR is recursively calculated with the help of functions *f* and *g*, defining the functions *f* and *g* as

$$f(a,b) = \ln\left(\frac{1+e^{a+b}}{e^a + e^b}\right) \tag{30}$$

$$g(a,b,u_s) = (-1)u_s a + b \tag{31}$$

where $a, b \in R$ and $u_s \in \{0, 1\}$. The recursion terminates when $N = 1$, which means it has reached the end of the channel *W*, at which point $L_1^{(1)} = \ln\frac{W(y_i|0)}{W(y_i|1)}$.

4. Simulation Results

4.1. Simulation Parameter Setting

In this section, simulation results are shown to evaluate the impact of different channel coding strategies on the performance of the OTFS system. This work implements a complete physical layer link simulation of an OTFS system containing different coding schemes using MATLAB simulation software and Monte Carlo simulation for performance evaluation. Table 1 summarizes the simulation parameters we used, with $N = 16$ and $M = 128$ defining the structure and size of the time-frequency grid used for mapping data symbols. Random data symbols are generated using MATLAB and modulated using 16QAM, 64QAM, and 256QAM constellation modulation schemes, followed by SFFT modulation. The 3GPP TDL-C channel model is chosen to represent the multipath and fading effects of the real-world environment. To simulate mobility speeds such as those of automobiles, high-speed railroads and airplanes, three speed levels of 30 km/h, 120 km/h and 300 km/h were selected. At the receiver, demodulation is performed through ISFFT. Within the framework of the OTFS system, we assume the ideal channel estimation, which means that the channel $H[n, m]$ is known. A minimum variance equalizer is employed for signal equalization, and the BER is calculated to evaluate system performance.

Table 1. The specific simulation parameters.

Parameter	Value
OTFS	$N = 16$ (Number of Time Slots), $M = 128$ (Number of Frequency Slots)
Cycle prefix length	16
Channel coding	5G NR LDPC/Turbo/Polar/Convolutional
Channel Model	3GPP TDL-C [20]
Symbol mapping	16QAM/64QAM/256QAM
Speed (km/h)	30, 120, 300

Next, this section evaluates the impact of different coding schemes on the OTFS system and investigates how different decoding algorithms affect the BER. Ultimately, the paper comprehensively analyzes the BER performance of coding strategies within the OTFS system, employing different modulation constellation mapping schemes under simulations with varying mobility speeds. These works provide profound insights into assessing the performance of the OTFS system in dynamic environments.

4.2. Comparison of Encoding Schemes under Different Movement Rates

Figure 5 exhibits the effect of four coding schemes on the BER of the OTFS system under different mobility speeds (30 km/h, 120 km/h, 300 km/h). The simulation utilizes a 64QAM modulation scheme and employs several coding and decoding strategies: LDPC codes with the Min-Sum decoding algorithm, Polar codes with the CRC-List-SC decoding algorithm, Turbo codes with the Linear-Log-MAP decoding algorithm, and Convolutional codes with the Max-Log-MAP decoding algorithm. All coding schemes are set with a code rate of 1/2 and a maximum of 16 iterations. Figure 5 shows that at low SNR, the performance of the uncoded OTFS system closely parallels that of the encoded OTFS system. However, as SNR increases, the system using codes reduces the BER by about

17–35% compared to the uncoded OTFS system. In this case, the LDPC code system has 10 dB SNR gain. With higher mobility speeds, multipath interference worsens, negatively affecting BER. LDPC codes consistently achieve the lowest BER across all speeds, particularly at high SNR. Polar codes struggle at low SNR but improve markedly as SNR increases. Meanwhile, the performance of Convolutional codes remains moderate across all SNR levels. Overall, channel coding substantially enhances the performance of OTFS systems.

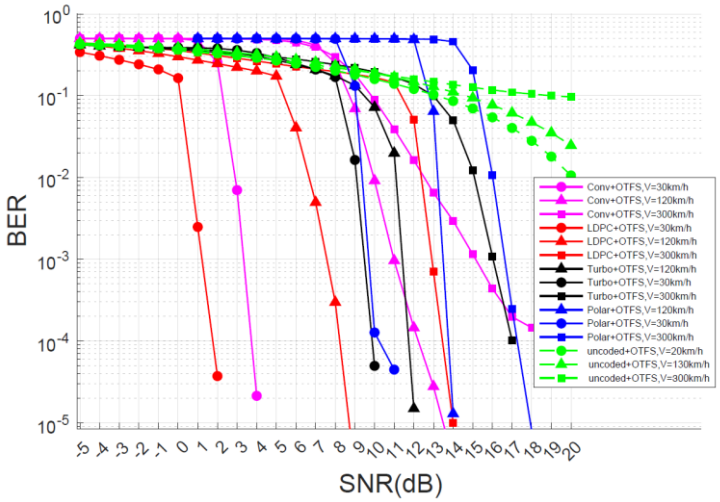

Figure 5. BER comparison for different mobility rates, V = 30 km/h, 120 km/h, 300 km/h.

4.3. Comparison of Different Decoding Algorithms

Figure 6 illustrates the comparison of BER for different coding schemes using different decoding algorithms. This work is conducted in an OTFS system at a 120 km/h speed using 64QAM modulation. Convolutional codes utilize Log-MAP/Max-Log-MAP decoding algorithms, Turbo codes employ Log-MAP/Linear-Log-MAP/Max-Log-MAP algorithms, LDPC codes are decoded using Sum-Product/PWLMin-Sum/Min-Sum algorithms, and Polar codes use Log-MAP/Max-Log-MAP algorithms. We observe that the BER performance of all decoding algorithms achieves significant gains at an SNR range of 20–25 dB. This demonstrates their exceptional error-correcting capabilities while maintaining communication quality. The decoding algorithms proposed in the paper effectively support system performance, ensuring efficient data transmission in high-speed mobility environments.

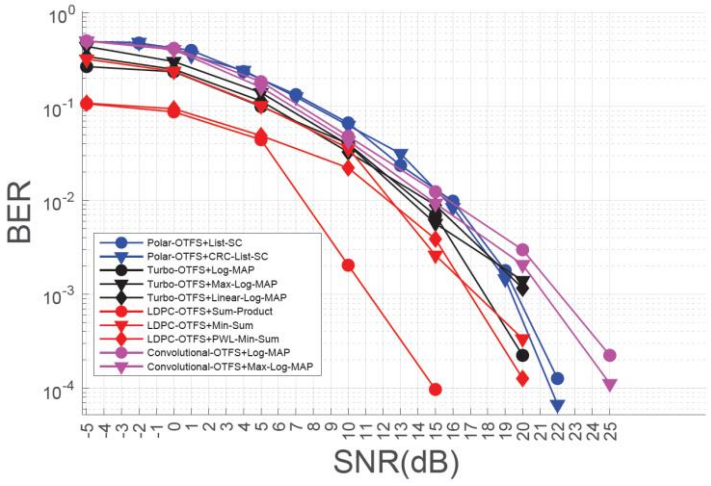

Figure 6. BER comparison for different decoding algorithms.

4.4. Comparison of Different Modulation Constellation Schemes

Figures 7–9 sequentially display the impact of four coding schemes combined with different modulation constellations on BER in an OTFS system at mobility speeds of 30 km/h, 120 km/h, and 300 km/h, respectively. Simulation results indicate that as the modulation order increases, the BER slope for all coding schemes flattens, revealing the vulnerability of higher-order QAM modulation at the same SNR levels. Despite this, the performance

of LDPC and Turbo codes remains exceptional, particularly LDPC codes, which maintain a sufficiently low BER even at SNR = 15 dB with 256QAM high-order modulation. In contrast, Polar codes only achieve low BER at high SNRs. Convolutional codes perform the worst, especially at higher modulation orders. These findings demonstrate the significant impact of modulation order on BER and the performance advantages of LDPC and Turbo codes in complex modulation environments.

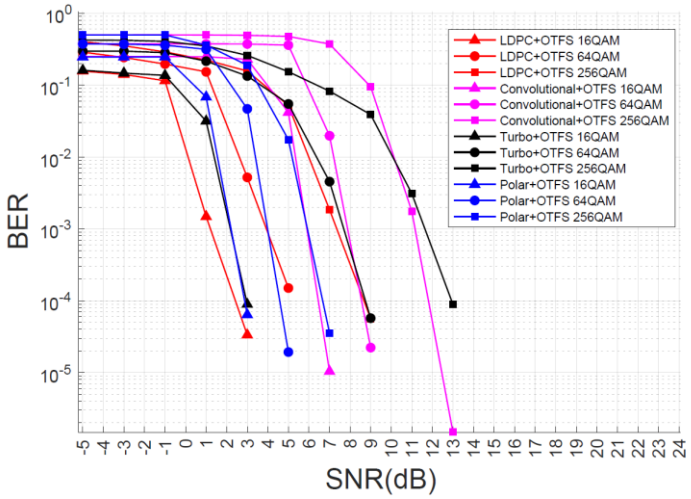

Figure 7. BER comparison for different modulation constellations, 16QAM/64QAM/256QAM, V = 30 km/h.

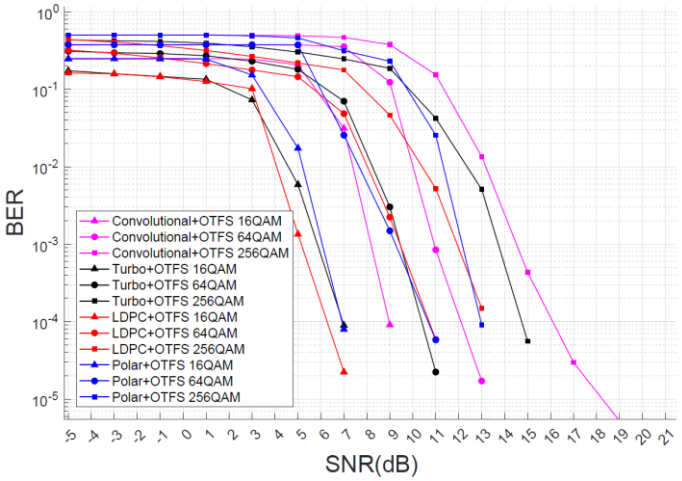

Figure 8. BER comparison of modulation constellations, 16QAM/64QAM/256QAM, V = 120 km/h.

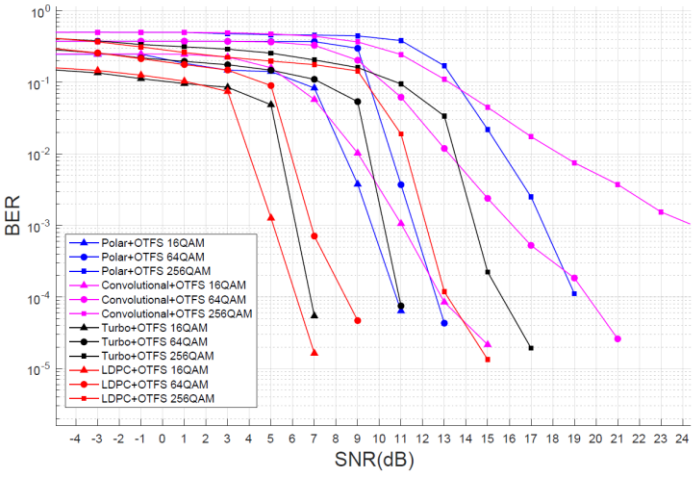

Figure 9. BER comparison of modulation constellations, 16QAM/64QAM/256QAM, V = 300 km/h.

5. Conclusion

This paper conducts a thorough simulation analysis to comprehensively evaluate the performance of different channel coding techniques in OTFS modulation schemes. The research results indicate that all four channel coding methods effectively improve the performance of the OTFS system, with Turbo and LDPC codes demonstrating similar and optimal performance within the OTFS system. This paper has only completed simulations in software and has not been implemented in hardware. Future work will explore the performance of these coding techniques under different OTFS modulation parameters and how to optimize algorithms in hardware implementations to balance performance and cost. This work will provide important theoretical and practical guidance for applying OTFS technology in future communication systems.

Author Contributions: Zehao Li: Conceptualization, Methodology, Software, Writing—Original Draft Preparation; Yi Gong: Data Curation, Writing—Original Draft Preparation; Xinru Li: Visualization, Investigation; Quan Wang: Supervision; Zhan Xu: Validation, Writing—Reviewing and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Beijing Natural Science Foundation (4242003, L223025) and Qin Xin Talents Cultivation Program of Beijing Information Science and Technology University (QXTCP B202405).

Data Availability Statement: Not applicable.

Acknowledgments: This work is supported by Beijing Natural Science Foundation (4242003, L223025) and Qin Xin Talents Cultivation Program of Beijing Information Science and Technology University (QXTCP B202405). The authors would like to thank the Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University, for their support of this paper.

Conflicts of Interest: All authors declare no conflict of interest.

References

1. Gong, Y.; Li, X.; Meng, F.; Liu, F.; Guizani, M.; Xu, Z. Towards Green RF Chain Design for Integrated Sensing and Communications: Technologies and Future Directions. *IEEE Commun. Maga.* **2024**, accepted.
2. Wang, X.; Shi, Y.; Xin, W.; Wang, T.; Yang, G.; Jiang, Z. Channel Prediction with Time-Varying Doppler Spectrum in High-Mobility Scenarios: A Polynomial Fourier Transform Based Approach and Field Measurements. *IEEE Trans. Wireless Commun.* **2023**, *22*, 7116–7129.
3. Yuan, W.; Li, S.; Wei, Z.; Cui, Y.; Jiang, J.; Zhang, H.; Fan, P. New delay Doppler Communication Paradigm in 6G Era: A Survey of Orthogonal Time Frequency Space (OTFS). *China Commun.* **2023**, *20*, 1–25.
4. Gong, Y.; Li, Q.; Meng, F.; Li, X.; Xu, Z. Data-Driven Deep Learning for OTFS Detection. *China Commun.* **2023**, *20*, 88–101.
5. Thomas, A.; Deka, K.; Raviteja, P.; Sharma, S. Convolutional Sparse Coding Based Channel Estimation for OTFS-SCMA in Uplink. *IEEE Trans. Commun.* **2022**, *70*, 5241–5257.
6. Park, J.; Hong, J.P.; Kim, H.; Jeong, B.J. Auto-Encoder Based Orthogonal Time Frequency Space Modulation and Detection with Meta-Learning. *IEEE Access* **2023**, *11*, 43008–43018.
7. Zhang, C.; Xing, W.; Yuan, J.; Zhou, Y. Performance of LDPC Coded OTFS Systems over High Mobility Channels. *ZTE Commun.* **2021**, *19*, 45.
8. Tian, H.; Zhao, D.; Jia, H. Research on Nonbinary LDPC-OTFS Scheme in High Mobile Communication Scenarios. In Proceedings of 2021 International Conference on Engineering and Emerging Technologies (ICEET), Istanbul, Turkey, 27–28 October 2021; pp. 1–6.
9. Elias, P. Coding for Noisy Channels. In *IRE WESCON Convention Record*; IEEE Operations Center: Piscataway, NJ, USA, 1955; Volume 2, pp. 94–104.
10. Berrou, C.; Glavieux, A.; Thitimajshima, P. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. 1. In Proceedings of ICC'93-IEEE International Conference on Communications, Geneva, Switzerland, 23–26 May 1993; pp. 1064–1070.
11. MacKay, D.J.; Neal, R.M. Near Shannon Limit Performance of Low Density Parity Check Codes. *Electron. Lett.* **1997**, *33*, 457–458.
12. Arikan, E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binaryinput memoryless channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073.
13. Gong, Y.; Li, Q.; Meng, F.; Li, X.; Xu, Z. ViterbiNet-Based Signal Detection for OTFS System. *IEEE Commun. Lett.* **2023**, *27*, 881–885.
14. Mohammed, S.K. Time-Domain to Delay-Doppler Domain Conversion of OTFS Signals in very High Mobility Scenarios. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6178–6183.

15. Schiavone, R.; Garello, R.; Liva, G. Performance Improvement of Space Missions Using Convolutional Codes by CRC-Aided List Viterbi Algorithms. *IEEE Access* **2023**, *11*, 55925–55937.
16. Liu, Y.; Guan, Y.L.; González, D. Turbo BEM OTFS Receiver with Optimized Superimposed Pilot Power. *IEEE Trans. Commun.* **2023**, *72*, 601–617.
17. Xu, Y.; Xiong, H.; Yang, G.; Zhou, H. Carrier Frequency Synchronization of LDPC Coded Transmissions in High Dynamic Environments. *IEEE Trans. Veh. Technol.* **2024**, 1–13. doi: 10.1109/TVT.2024.3383891
18. Ding, X.; Zhou, K.; Li, G.; Yang, K.; Gao, X.; Yuan, J.; An, J. Customized Joint Blind Frame Synchronization and Decoding Methods for Analog LDPC Decoder. *IEEE Trans. Commun.* **2023**, *72*, 756–770.
19. Niu, K.; Zhang, P.; Dai, J.; Si, Z.; Dong, C. A Golden Decade of Polar Codes: From Basic Principle to 5G Applications. *China Commun.* **2023**, *20*, 94–121.
20. Hasegawa, F.; Taira, A.; Noh, G.; Hui, B.; Nishimoto, H.; Okazaki, A.; Okamura, A.; Lee, J.; Kim, I. High-speed train communications standardization in 3GPP 5G NR. *IEEE Commun. Stand. Mag.* **2018**, *2*, 44–52.