IEEE Open Journal of Vehicular Technology

Received 4 May 2025; accepted 20 May 2025. Date of publication 23 May 2025;
date of current version 21 July 2025. The review of this article was coordinated by the Guest Editors for the Special Section on Orthogonal Time Frequency Space Modulation and Delay-Doppler Signal Processing.

Digital Object Identifier 10.1109/OJVT.2025.3573073

LDPC-Hadamard Code-Assisted OTFS in High-Mobility Scenarios

YI GONG [1] (Member, IEEE), ZEHAO LI [1], LICHUN YANG [2], LU TIAN [1] (Member, IEEE), JUN MIAO [3], DESHENG ZHANG [4,5], AND ZHAN XU [1] (Member, IEEE)

[1]Beijing Information Science and Technology University, Beijing 102206, China
[2]Shenzhen Penguin Network Technology Company, Shenzhen 518000, China
[3]China Academy of Space Technology, Beijing 100094, China
[4]Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China
[5]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 101408, China

CORRESPONDING AUTHORS: LU TIAN; JUN MIAO (e-mail: tianlu@bistu.edu.cn; miaojun@buaa.edu.cn).

This work was supported in part by the Beijing Natural Science Foundation under Grant 4242003, in part by the National Natural Science Foundation of China under Grant 62201067 and Grant 62201070, in part by Beijing Natural Science Foundation under Grant L223025, and in part by the Qin Xin Talents Cultivation Program of Beijing Information Science and Technology University under Grant QXTCP B202405.

ABSTRACT In high-speed mobile scenarios, Orthogonal Time Frequency Space (OTFS) effectively addresses the Doppler frequency shift. Channel coding enhances the performance of OTFS systems by adding redundancy and error correction capabilities. However, existing coding schemes still face limitations when dealing with complex conditions such as highly dynamic channel variations, delay spread, and Doppler effects. This paper proposes a Low-Density Parity-Check (LDPC)-Hadamard code to assist OTFS in combating complex channel conditions, thereby improving data transmission efficiency. In particular, the LDPC-Hadamard code leverages the sparsity of LDPC and the orthogonality of Hadamard codes to address the issue of rapidly varying channels in OTFS systems. Additionally, a low-complexity quasi-cyclic LDPC code construction algorithm is proposed to decrease the coding complexity. Otherwise, a Log-Likelihood Ratio Belief Propagation (LLR-BP) decoding algorithm is introduced, which reduces the computational complexity. Therefore, the proposed coding and decoding methods enable the OTFS to achieve better performance in low SNR conditions. Compared to conventional quasi-cyclic LDPC schemes, the proposed LDPC-Hadamard code-assisted OTFS system exhibits a 0.2 dB SNR gain and achieves a lower BER of 10^{-6} at 1.1 dB SNR, demonstrating enhanced robustness against Doppler-induced impairments in highly dynamic channels.

INDEX TERMS LDPC-Hadamard, OTFS, channel coding, delay doppler domain.

I. INTRODUCTION

Conventional Orthogonal Frequency Division Multiplexing (OFDM) exhibits suboptimal performance in high-mobility scenarios [1]. Doppler shifts cause frequency variations, making it difficult for the receiver to accurately demodulate the subcarriers and increasing the risk of Inter-Symbol Interference (ISI) [2]. However, Orthogonal Time Frequency Space (OTFS) modulation, as proposed in [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], signals by transforming them from the time-frequency (TF) domain to the delay-Doppler (DD) domain. This conversion effectively turns time-varying channels into approximately time-invariant channels, ensuring enhanced stability in the DD domain and robust resistance to Doppler shifts and channel fading. The process is implemented through two-dimensional transformations—the Inverse Symplectic Finite Fourier Transform (ISFFT) and the Heisenberg transform—which map information symbols onto the DD domain grid. Subsequently, these symbols are converted back to the TF domain for transmission. In [14], OTFS suppresses fast time-varying channel distortion induced by high-speed vehicle movement, enabling ultra-low bit error rate (BER) communication for vehicle-road-cloud collaboration. For UAV swarm networking, its Doppler spread resistance supports high-speed connectivity among dense

node clusters [15]. In satellite communication scenarios, OTFS effectively mitigates the coupling effect between long propagation delays and dynamic Doppler shifts, thereby enhancing transmission reliability in non-geostationary orbital constellations [16]. Experimental results demonstrate that, in contrast to traditional OFDM modulation, OTFS modulation effectively meets the communication requirements of high-mobility scenarios, as evidenced by [17], [18], [19], [20], [21].

Channel coding [22], [23], [24] plays a vital role in ensuring reliable data transmission. To achieve higher transmission rates and enhanced interference immunity in high-mobility scenarios, coding is commonly applied to enhance OTFS modulation and optimize system performance. By analyzing the conditional Pairwise Error Probability (PEP), the performance limit of the coded OTFS system is determined, demonstrating its superiority over coded OFDM in highly dynamic environments, as shown in [25]. In the OTFS system, convolutional coding enhances error correction performance in doubly selective channels, but its continuous nature may lead to residual interference in high-mobility scenarios. [26] proposes an iterative soft cancellation scheme combined with element-wise extrinsic information feedback to mitigate this interference. A scheme that combines Polar code and the OTFS system is proposed in [27]. Polar code transmits information from reliable channels utilizing the "polarization effect" which is effective in high Signal-to-Noise Ratio (SNR) environments. However, in low SNR environments, performance is impacted by the code rate and parameters of polar codes. The above limitation may result in insufficient flexibility to address multipath effects and dynamic channel changes, preventing the full utilization of the OTFS system's advantages, and thereby affecting overall error correction capability and transmission efficiency. Non-binary Low-Density Parity-Check (LDPC)-OTFS coding and modulation schemes have been proposed in [28]. However, the encoding and decoding processes of non-binary LDPC codes are often highly complex. In the OTFS system, non-binary signals can cause increased latency and exacerbate the effects of Doppler shifts, thereby limiting system performance.

Therefore, to address future communication scenarios, such as high-speed vehicles, high-speed railroads, airplanes, and satellite and deep-space communications, efficient and low-latency coding schemes need to be developed to guarantee reliable and real-time data transmission in highly dynamic environments [29], [30], [31], [32], [33]. Coding schemes effectively mitigate channel fading and interference while maximizing channel capacity to ensure efficient data transmission in fast-moving and complex channel conditions [34].

This paper offers the following main contributions:

- The paper proposes an LDPC-Hadamard code to assist OTFS in mitigating the Doppler shift. By leveraging the sparsity of LDPC codes and the orthogonality of Hadamard code, the proposed scheme improves data transmission efficiency in high-noise environments.
- An algorithm with low complexity is proposed to construct LDPC codes in a quasi-cyclic form. Aiming to address the difficulty in constructing LDPC-Hadamard code.
- A Log-Likelihood Ratio Belief Propagation (LLR-BP) decoding algorithm is proposed to optimize computational complexity and minimize delays during decoding.

The remainder of the paper is structured as follows: Section II introduces the OTFS system model with LDPC-Hadamard coding assistance. Section III presents the specific algorithm for LDPC-Hadamard code. The experiments are performed in Section IV to evaluate the performance of the LDPC-Hadamard-assisted OTFS system. Finally, we conclude the paper in Section V.

II. SYSTEM MODEL

In high-mobility scenarios, the Doppler shift is specifically resisted by OTFS modulation [35], [36], [37]. To effectively combat Doppler shift in high-mobility scenarios, this paper proposes a scheme utilizing LDPC-Hadamard codes to assist the OTFS system. The focus is on simulating a highly dynamic scenario using a time-frequency dual-selective channel [38], exploring the impact of LDPC-Hadamard coding in enhancing the OTFS modulation scheme. Further details are provided in Fig. 1.

First, a Cyclic Redundancy Check (CRC) [39] is performed on the message sequence u, generating a checksum that is appended to the original sequence, resulting in an extended message. The sequence is then processed by an LDPC-Hadamard encoder, producing a codeword c with strong error-correcting capabilities. The codeword is mapped to signal points in the complex plane using constellation modulation, converting the digital information into analog signals. OTFS modulation then maps the signal to a two-dimensional TF domain, reducing Doppler effects [40]. The modulated signal is then transmitted over a physical channel, where the receiving end performs OTFS demodulation, constellation demodulation, and LDPC-Hadamard decoding. Finally, perform a modulo-2 division on the received data to complete the CRC check.

A. OTFS MODULATION

Fig. 2 illustrates the OTFS modulation and demodulation process. Dividing the TF domain by T in time and Δf in frequency results in a discrete TF grid

$$\Lambda_{\mathrm{TF}} = \{(nT, m\Delta f), n = 0, \ldots, N-1, m = 0, \ldots, M-1 \tag{1}$$

where the number of subcarriers and the number of time intervals in OTFS are represented by N and M, respectively. The corresponding DD domain is denoted as

$$\Lambda_{\mathrm{DD}} = \left\{ \left(\frac{k}{NT}, \frac{l}{M\Delta f} \right), k = 0, \ldots, N-1, l = 0, \ldots, M-1 \right\}. \tag{2}$$

The signal symbol $x[k, l]$ is first mapped to the DD domain, then transformed to the TF domain signal $X[n, m]$ using the

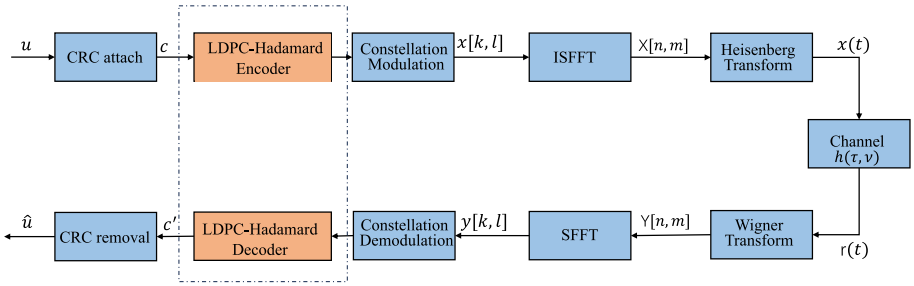

FIGURE 1. LDPC-Hadamard code-assisted OTFS system modeling.

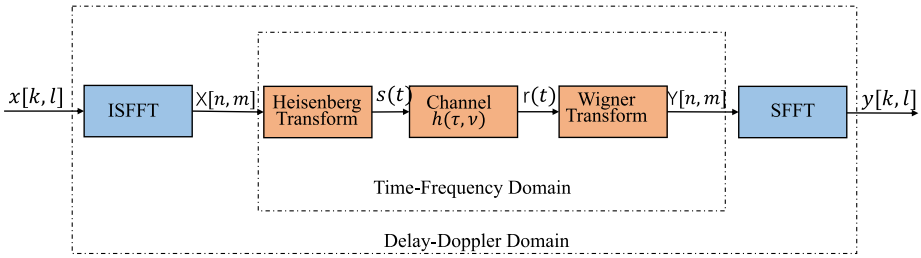

FIGURE 2. OTFS modulation and demodulation process.

ISFFT as shown in

$$X[n,m]=\frac{1}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1}x[k,l]e^{j2\pi\left(\frac{kn}{N}-\frac{ml}{M}\right)}. \tag{3}$$

Then, the signal $X[n, m]$ in the TF domain is converted to the TF signal $s(t)$ using the Heisenberg transform with $g_{tx}(t)$ as the pulse as

$$s(t)=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}X[n,m]g_{tx}(t-nT)e^{j2\pi m\Delta f(t-nT)}. \tag{4}$$

B. OTFS DEMODULATION

After experiencing the time-frequency double-selective channel $h(\tau, v)$, the receiver receives the signal $r(t)$, which can be expressed as

$$r(t)=\int\int h(\tau,v)s(t-\tau)e^{j2\pi v(t-\tau)}d\tau dv+z(t), \tag{5}$$

where $z(t)$ is the noise component. Subsequently, the Wigner transform of $r(t)$ is performed using $g_{rx}^{*}(t)$ as the received pulse. Sampling is then carried out with a time interval T and a frequency interval Δf, and the TF domain received signal $Y[n, m]$ is yielded.

$$Y(t,f)=\int g_{rx}^{*}(t'-t)r(t')e^{-j2\pi f(t'-t)}dt', \tag{6}$$

$$Y[n,m]=Y(t,f)|_{t=nT,f=m\Delta f}. \tag{7}$$

Finally, the receiver performs Symplectic Finite Fourier Transform (SFFT) on $Y[n, m]$ to obtain the DD domain signal as

$$y[k,l]=\frac{1}{NM}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}Y[n,m]e^{-j2\pi\left(\frac{nk}{N}-\frac{ml}{M}\right)}, \tag{8}$$

where $y[k, l]$ is the received signal, the receiver demodulates $y[k, l]$ and performs channel equalization to recover the original transmitted information.

III. DETAILED ENCODING AND DECODING STRUCTURE OF LDPC-HADAMARD CODES

A. HADAMARD CODE

The Hadamard matrix H_n is a square matrix composed of $\{+1, -1\}$, and it satisfies $H_nH_n^T=nI$, where I is the n-dimensional identity matrix. Since $nH_n^{-1}=H_n^T$, it follows that $H_n^TH_n=nI$. The orthogonality of the columns implies that the Hadamard matrix is equal to its transpose, meaning the transpose of the Hadamard matrix is also a Hadamard matrix. When $n=2^r$, the Hadamard matrix H_n can be recursively constructed starting from $H_1=[+1]$ as

$$H_n=\begin{bmatrix}+H_{n/2} & +H_{n/2}\\ +H_{n/2} & -H_{n/2}\end{bmatrix}, \tag{9}$$

where the order of the Hadamard matrix is called r.

1) HADAMARD ENCODING STRUCTURE

The Hadamard codewords c_n generated from information bits b_r can be expressed as

$$c_n=E(b_0,b_1,\ldots,b_r), \tag{10}$$

where E stands for Hadamard code encoder, $\{c(0) = b_0, c(2^i) = b_{i+1} \mid i = 0, 1, \ldots, r-1\}$ are the information bits in c_n and the remaining bits are generated as check bits.

Hadamard code can be obtained by circular recursion. when $n = 4$, the Hadamard matrix is denoted as

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{11}$$

Its generating matrix can be expressed as

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{12}$$

Given that $n = 2^r$ and $n = 4$, then $c_4 = b_2 G_4$. The Hadamard codeword c_n generated from the given information b_r. Therefore, the Hadamard codeword c_{2n} can be expressed as

$$c_{2n} = [c_n, b(r+1) \oplus b(0) \oplus c_n], \tag{13}$$

where $\oplus$ stands for binary addition.

2) HADAMARD DECODING STRUCTURE

c denotes the coded sequence, and y represents the received sequence. The posterior LLR of the message, after transmission through an AWGN channel with variance σ^2, is given by

$$L(c_i|\mathrm{y}) = \log \frac{\sum_{\mathrm{H}_{i,j}=\pm 1} \gamma(\pm \mathrm{h}^j)}{\sum_{\mathrm{H}_{i,j}=\mp 1} \gamma(\pm \mathrm{h}^j)}, \tag{14}$$

more details are given in Appendix A.

$$\gamma(\pm h^j) = \exp\left(\frac{1}{2}\langle \pm h^j, \tilde{L}\rangle\right), \tag{15}$$

$$\tilde{L} = \frac{2y}{\sigma^2}. \tag{16}$$

In order to compute $\gamma(\pm h^j)$, it is necessary to calculate the inner product $\langle \pm h^j, \tilde{L}\rangle$, where $\tilde{L}$ represents the elements in $H_n\tilde{L}$. Here, $\pm h^j$ represents the j-th column of the Hadamard matrix along with its sign. $\tilde{L}$ represents the received sequence after being scaled. $H_n\tilde{L}$ is efficiently computed with the Fast Hadamard Transform (FHT), allowing $\gamma(\pm h^j)$ to be easily obtained. Detailed information about FHT can be found in Appendix B.

B. LDPC CODE

1) LDPC ENCODING STRUCTURE

LDPC codes belong to the class of (n, k) linear block codes, where each group of k symbols from the information stream is linearly transformed to produce m redundant symbols, resulting in a codeword c of n symbols. The codeword length is $n = k + m$, where k represents the number of original input bits. Code rate $R = \frac{k}{n}$ represents the efficiency of information transmission. Additionally, the relationship between the generator matrix $G_{k\times n}$ and the codeword c is expressed as $c = sG$. An LDPC code's parity-check matrix is a sparse matrix, with most elements being zero and only a few ones. The sparse structure reduces the computational complexity in encoding and decoding while preserving strong error-correcting capabilities. LDPC code can also be represented using Tanner graphs [41]. The graphical model of LDPC codes makes it easier to understand both the construction and decoding algorithms.

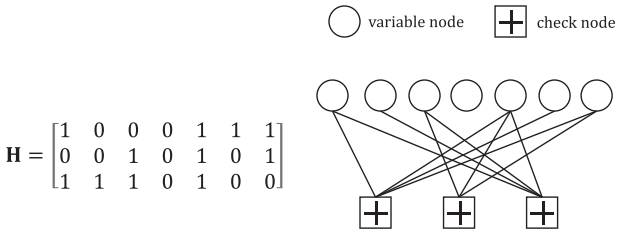

FIGURE 3. Tanner graph of LDPC code.

A Tanner graph divides all nodes into two distinct subsets. One subset, called Variable Nodes (VNs), is represented by circles and corresponds to the columns in the matrix. Another subset, called Check Nodes (CNs), is represented by squares and corresponds to the rows in the matrix. In the parity-check matrix, each row is linked to a check node and each column to a variable node, establishing the connections that form the Tanner graph, where the matrix entries define the relationships between nodes.

In the graph, check nodes are indicated by c_i, where i takes values from 0 to $m-1$. Similarly, variable nodes are represented as v_j, with j ranging from 0 to $n-1$. If the element in the parity-check matrix at row i and column j equals 1, it indicates a connection between c_i and s_j in the Tanner graph of the LDPC code, as illustrated in Fig. 3.

Progressive Edge Growth (PEG) Algorithm: The PEG algorithm is designed to optimize the parity-check matrix of LDPC codes by progressively adding edges between variables and check nodes. It prioritizes connections that increase the shortest cycle length in the Tanner graph, thereby reducing the occurrence of short cycles. By determining the shortest paths linking variable nodes to check nodes, an algorithm connects edges to the farthest nodes, reducing short cycles and improving error-correcting performance in LDPC codes.

The algorithm proceeds with the following steps:

Initialize variable nodes n, check nodes m, and the degree distribution of the variable nodes. For each variable node s_j $(j = 1, 2, \ldots, n)$, search for check nodes connected to s_j, i.e., adding edges to s_j in the Tanner graph.

a) Iterate $k = 0$ to $d_{s_j} - 1$, representing the number of nodes to be added to each column.
b) If $k = 0$, for adding the first edge to node s_j, simply find the check node c_i with the smallest degree from all

check nodes, and add the edge (c_i, s_j). If $k \neq 0$, traverse from node s_j and record N_{sj}^l.

c) If, during traversal, the size of N_{sj}^l stops increasing and is smaller than m, indicating an unreachable check node starting from s_j, select the check node c_i with the smallest degree from $\bar{N}_{s_j}^l$, and add the edge (c_i, s_j). Return (a), then continue with the next edge addition.

d) If both $\bar{N}_{s_j}^l \neq \emptyset$ and $\bar{N}_{s_j}^{l+1} \neq \emptyset$ are found during traversal, it indicates that all check nodes are reachable from s_j. Select the check node c_i with the smallest degree from $\bar{N}_{s_j}^l$, add the edge (c_i, s_j), return (a), and proceed with the next edge addition.

In the PEG algorithm, variable and check node degrees are organized in a non-decreasing order, ensuring that node degrees either remain constant or increase as the sequence progresses. For the variable node s_j, denoted by d_{s_j}, N_{sj}^l represents the set of check nodes accessible through tree expansion from s_j, while the complement set is denoted as $\bar{N}_{s_j}^l$. Additionally, the Tanner graph of LDPC codes generated by the PEG algorithm features relatively long shortest cycles, indicating that the parity-check matrix possesses a large girth. The girth g_p is defined by

$$g_p \geq 2\left(\left\lfloor \frac{\log\left(md_c^{max} - \frac{md_c^{max}}{d_s^{max}} - m + 1\right)}{\log\left[(d_s^{max} - 1)(d_c^{max} - 1)\right]} - 1 \right\rfloor + 2\right). \tag{17}$$

Here, the maximum degree distributions of the check nodes and bit nodes are represented by $d_c^{\max}$ and $d_s^{\max}$, respectively.

The PEG algorithm exhibits *exponential complexity* for non-binary LDPC codes over GF(q), due to the combinatorial search space of edge weights ($(q-1)^{d_v}$ combinations per variable node) and the dual constraints of structural/symbolic cycle avoidance. Symbolic cycles impose additional $\mathcal{O}(q)$ verification costs, rendering PEG intractable for $q \geq 16$ even with moderate code lengths.

Therefore, this paper proposes a low complexity **Quasi-Cyclic LDPC Code Construction Algorithm:** The parity-check matrix H of Quasi-cyclic LDPC codes is composed of multiple cyclically shifted submatrices, where the submatrices are identity matrices and zero matrices generated through specific cyclic shifts. Let p^i denote the ith cyclic shift of an L-order identity matrix, where $0 \leq i < L$, and p^∞ denote a zero matrix. The parity-check matrix H, with dimensions $mL \times nL$, is defined as

$$H = \begin{bmatrix} P^{\alpha_{11}} & P^{\alpha_{12}} & \cdots & P^{\alpha_{1n}} \\ P^{\alpha_{21}} & P^{\alpha_{22}} & \cdots & P^{\alpha_{2n}} \\ \vdots & \vdots & & \vdots \\ P^{\alpha_{m1}} & P^{\alpha_{m2}} & \cdots & P^{\alpha_{mn}} \end{bmatrix}, \tag{18}$$

where $\alpha_{ij} \in \{0, 1, \ldots, L-1, \infty\}$, $1 \leq i \leq m$, and $1 \leq j \leq n$.

Hence, the code C, associated with the parity-check matrix H, follows a cyclic structure. Constructing a quasi-cyclic LDPC code involves determining the cyclic shift matrices. Once these parameters are established, the parity-check matrix H is fully defined.

This paper achieves low-complexity LDPC code by leveraging the sparsity of the checksum matrix and preprocessing it by rearranging its rows and columns. The process divides the transformed checksum matrix H into six parts:

$$H = (H_I \mid H_P) = \begin{pmatrix} A & B & T \\ C & D & E \end{pmatrix}, \tag{19}$$

where $H_p = \begin{pmatrix} B & T \\ D & E \end{pmatrix}$ represents the checksum part, and T is the lower triangular array. Assume the codeword $c = (b, p_1, p_2)$, where the systematic bits are represented by b, and the checksum bits of lengths g and $m-g$ are represented by p_1 and p_2, respectively. The encoded codeword c satisfies the condition $Hc^T = 0^T$.

$$\begin{cases} p_1^{\mathrm{T}} = -\Phi^{-1}(-ET^{-1}A + C)s^{\mathrm{T}} \\ p_2^{\mathrm{T}} = -T^{-1}(As^{\mathrm{T}} + Bp_1^{\mathrm{T}}) \end{cases}, \tag{20}$$

where $\Phi = -ET^{-1}B + D$. With a reasonable number of steps, the coding complexity is $O(n + g^2)$. The encoding complexity approximates $O(n)$ when $g \ll n$. However, careful consideration of the approximate lower triangular matrix structure of the H-matrix is required for various g-values. Overemphasizing the choice of g-values for LDPC code may also negatively impact the performance.

Preprocessing the rows and columns of the check matrix transforms the sub-matrix T into a lower triangular array, reducing the computational complexity of T^{-1}. However, the preprocessing must be performed serially, which significantly increases coding delay. Additionally, the sparsity of Φ is not guaranteed after preprocessing. From (20), it is clear that selecting appropriate structures for the submatrices E, B, D, and T can reduce the computational complexity of Φ^{-1} and T^{-1}. Drawing from the analysis, this paper introduces a low-complexity method for constructing quasi-cyclic LDPC codes.

First, the LDPC code's checksum matrix H is divided into an information part H_I and a checksum part H_P, i.e., $H = [H_I \mid H_P]$, where the size of H_I is $mL \times kL$ (with $k = n - m$), and the size of H_P is $mL \times mL$. To achieve efficient encoding, H_P is constrained to be a lower triangular matrix.

H_I consists of $m \times k$ cyclic shift matrices or zero matrices, each of size $L \times L$. 0 and I represent the zero and identity matrices of size $L \times L$, respectively. p^{∂_i} represents the ∂_i-th ($\partial_i = 0, 1, 2, \ldots, L-1$) cyclic shift of the identity matrix.

$$T = \begin{bmatrix} I & 0 & \cdots & 0 & 0 \\ P^{\partial_2} & I & \cdots & 0 & 0 \\ 0 & P^{\partial_3} & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & P^{\partial_{m-1}} & I \end{bmatrix}, \tag{21}$$

where T is a lower bi-diagonal matrix, making it straightforward to compute T^{-1}. $ET^{-1}B = p^{\partial_m}[p^{(2,m-1)}\ p^{(3,m-1)}\ \ p^{(4,m-1)} \cdots p^{\partial_{m-1}}\ \ I]B = p^{(1,m)}$.

$$\text{From } \Phi = -ET^{-1}B + D,$$

$$\text{To } \Phi = -P^{(1,m)} + D.$$

Therefore, $\Phi = -ET^{-1}B + D = -P^{(1,m)} + D$, which forms a lower bi-diagonal matrix. The quasi-cyclic structure of Φ enables *linear-time inversion* via forward substitution (complexity $\mathcal{O}(n)$), in contrast to the $\mathcal{O}(n^3)$ cost for dense matrices. Additionally, the block-circulant property of $P^{(1,m)}$ and D in QC-LDPC codes allows storage-efficient representation (using shift values rather than full matrices) and fast encoding via FFT-based multiplications, reducing both time and space complexity to $\mathcal{O}(n \log n)$ and $\mathcal{O}(1)$ per block, respectively. The detailed encoding process is as follows:

Step	Description
Step 1	Calculate AS^T and CS^T.
Step 2	Calculate $\left[p_{(2,m)}, p_{(3,m)}, \ldots, pm\right] AS^T$.
Step 3	Calculate $ET^{-1}AS^T + CS^T$.
Step 4	Calculate $p_1^T = -\Phi^{-1}(ET^{-1}AS^T + CS^T)$.
Step 5	Calculate $p_2^T = -T^{-1}(AS^T + Bp_1^T) - T^{-1}(AS^T + Bp_1^T)$.

2) LDPC DECODING STRUCTURE

The BP algorithm: It is an iterative algorithm that enables fast decoding through parallel operations. During decoding, each node passes information to its neighboring nodes. Once the information is updated, the variable bits are evaluated. The process stops when either the decoding is correct or the maximum number of iterations is reached. If neither condition is met, the algorithm proceeds to the next iteration. Fig. 4 illustrates the flowchart of the BP algorithm.

The BP algorithm's iterative decoding process consists of the following steps:

a) *Initialization:* Let i denote a variable node, while j represents a check node. $P(x_i = a \mid y_i)$ represents the probability of $x_i = a$ given the received value y_i, where $a = \pm 1$, and f_i^0 and f_i^1 represent the probabilities of receiving $x_i = 1$ and $x_i = -1$, respectively. Thus,

$$f_i^0 = P(x_i = 1 \mid y_i) = \frac{1}{1 + e^{\frac{-2y_i}{\sigma^2}}}, \tag{22}$$

$$f_i^1 = P(x_i = -1 \mid y_i) = \frac{1}{1 + e^{\frac{2y_i}{\sigma^2}}}. \tag{23}$$

The initial information for variable node i is

$$q_{ij}^0 = f_i^0 q_{ij}^1 = f_i^1. \tag{24}$$

b) *Check Node Information Update:* For each check node j and its connected variable nodes $i \in V(j)$ (where $V(j)$ is the set of nodes linked to j), compute the messages passed from j to i:

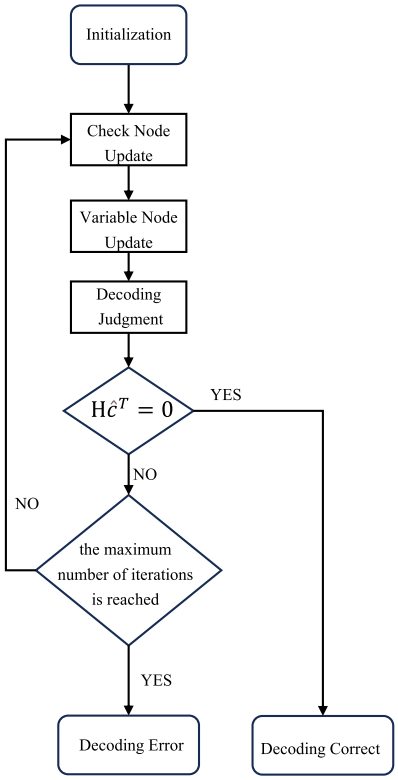

FIGURE 4. Flowchart of the BP Algorithm.

$$r_{ji}^0 = \frac{1}{2} + \frac{1}{2} \prod_{i' \in V(j) \backslash i} \left(1 - 2q_{i'j}^0\right), \tag{25}$$

$$r_{ji}^1 = \frac{1}{2} - \frac{1}{2} \prod_{i' \in V(j) \backslash i} \left(1 - 2q_{i'j}^1\right). \tag{26}$$

c) *Variable Node Information Update:* For each variable node i and its connected check nodes $j \in C(i)$ (where $C(i)$ is the set of nodes linked to i), compute the messages passed from i to j:

$$q_{ij}^0 = k_{ij} f_i^0 \prod_{j' \in C(i) \backslash j} r_{j'i}^0, \tag{27}$$

$$q_{ij}^1 = k_{ij} f_i^1 \prod_{j' \in C(i) \backslash j} r_{j'i}^1. \tag{28}$$

k_{ij} is a correction factor ensuring that $q_{ij}^0 + q_{ij}^1 = 1$.

d) *Decoding Judgment:* Compute the pseudo a posteriori probability information for all variable nodes:

$$q_i^0 = K_i f_i^0 \prod_{j' \in C(i)} r_{j'i}^0, \tag{29}$$

$$q_i^1 = K_i f_i^1 \prod_{j' \in C(i)} r_{j'i}^1, \tag{30}$$

K_i is the correction factor ensuring that $q_i^0 + q_i^1 = 1$. If $q_i^1 > q_i^0$, the received bit sequence is $\hat{c}_i = 1$, otherwise $\hat{c}_i = 0$. When $H\hat{c}^T = 0$ or the iteration limit is reached, the operation stops. If not, the iteration resumes from step (b).

In the previous section, the BP decoding algorithm for LDPC code was introduced. Based on probabilistic information, the BP algorithm includes multiplication operations during the iterations. As a result, computational complexity increases, and more processing time is required, making it less suitable for hardware implementation.

LLR-BP algorithm: This paper proposes a method to represent probabilistic information using LLR, which reduces the computational complexity of decoding.

The channel's initial message is represented as

$$L(p_i) = \ln\left(\frac{P_i(0)}{P_i(1)}\right) = \ln\left(\frac{\Pr(x_i = 0 \mid y_i)}{\Pr(x_i = 1 \mid y_i)}\right). \tag{31}$$

The message sent from a check node to a variable node is expressed as

$$L(r_{ji}) = \ln\left(\frac{r_{ji}(0)}{r_{ji}(1)}\right). \tag{32}$$

The message sent from a variable node to a check node is expressed as

$$L(q_{ij}) = \ln\left(\frac{q_{ij}(0)}{q_{ij}(1)}\right). \tag{33}$$

The combined message received by the variable node is represented as

$$L(q_i) = \ln\left(\frac{q_i(0)}{q_i(1)}\right). \tag{34}$$

The specific steps of the LLR-BP decoding algorithm are given below:

a) *Initialization:* Calculate the LLR of the initial probability passed from the channel to the variable node, $L(p_i)$, where $p_i(0) = 1 - p_i(1)$. Set the initial message from the variable node to the check node as

$$L^{(0)}(q_{ij}) = L(p_i). \tag{35}$$

b) *Check Node Information Update:*

$$L^{(l)}(r_{ji}) = 2\tanh^{-1}\left(\prod_{i' \in V(j)\backslash i} \tanh\left(\frac{1}{2}L^{(l-1)}(q_{i'j})\right)\right). \tag{36}$$

c) *Variable Node Information Update:*

$$L^{(l)}\left(q_{ij}\right) = L(p_i) + \prod_{j' \in C(i)\backslash j} L^{(l)}(r_{j'i}). \tag{37}$$

d) *Decoding Judgment:*

$$L^{(l)}\left(q_i\right) = L(p_i)_i + \prod_{j' \in C(i)} L^{(l)}(r_{j'i}). \tag{38}$$

If $L^{(l)}(q_i) > 0$, then $\hat{c}_i = 0$, otherwise $\hat{c}_i = 1$. If $H\hat{c}^T = 0$ or the maximum number of iterations is reached, the iteration process terminates. If the stopping condition is not satisfied, the iteration continues from step (b).

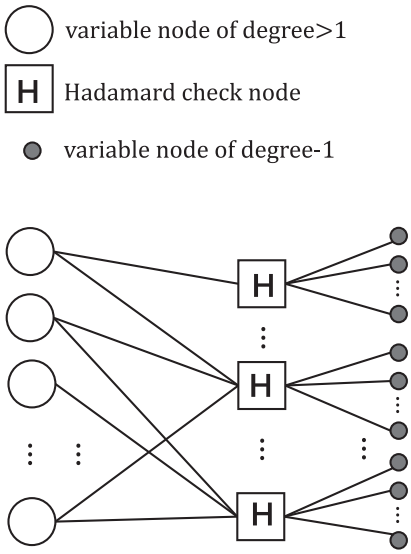

FIGURE 5. The Tanner graph of the LDPC-Hadamard code.

C. LDPC-HADAMARD CODE

This paper proposes a concatenated LDPC-Hadamard code with a dual-layer structure: the outer code is LDPC and the inner code is Hadamard. The core innovation lies in combining the sparse graph coding properties of LDPC with the strong error correction capability of Hadamard codes. Specifically, the Tanner graph constructed from the LDPC parity-check matrix (H matrix) dynamically partitions check nodes into two constraint types:

1) LDPC constraints acting on variable nodes with degree > 1.
2) Hadamard orthogonal constraints applied to leaf nodes with degree 1.

In Fig. 5, our hierarchical mapping strategy facilitates the re-encoding of checkpoint-related bits into Hadamard codewords, with parity equations constructed through linear combinations in the Hadamard transform domain. The encoding process involves two phases: First, generating LDPC codewords, followed by Hadamard expansion encoding on bits covered by each checkpoint. The final joint codeword inherits both the long-code error correction gain of LDPC and the short-code burst noise resistance of Hadamard. Key parameters, such as the order of Hadamard codes, are designed to align with the connection degrees of LDPC check nodes, achieving an optimal trade-off between complexity and performance.

Thus, LDPC-Hadamard codes are a type of concatenated coding, where Hadamard serves as the inner code and LDPC as the outer code. The structure allows the strengths of both codes to be combined for enhanced performance. Consequently, the decoding algorithm for the LDPC-Hadamard

TABLE 1. Simulation Parameters

Parameter	Value
OTFS	N=16, M=128
Channel coding	LDPC-Hadamard/quasi-cyclic LDPC
Channel model	AWGN/3GPP TDL-C [42]
Modulation techniques	BPSK/16QAM/64QAM/256QAM
Speed (km/h)	120, 300, 500
Hadamard order	r=4

code can be designed. During the process, information is iteratively exchanged between the decoders for variable nodes and Hadamard-constrained check nodes. The LLR-BP decoding algorithm handles the variable nodes, while Hadamard code decoding is applied to the check nodes. Upon the completion of the exchange, the decoding process concludes.

IV. SIMULATION RESULTS

This section provides simulation results to assess the performance of the LDPC-Hadamard code and the OTFS system enhanced by LDPC-Hadamard coding. The simulations were conducted using MATLAB software and covered several different experimental scenarios, with Monte Carlo simulations ($N_{\mathrm{MC}} = 10^6$ trials) employed to estimate the expected results. Table 1 summarizes the simulation parameters used. The OTFS system's DD domain resource allocation sets $N = 16$ subcarriers and $M = 128$ time slots, with subcarrier spacing $\Delta f = 15$ kHz, symbol duration $T = 66.7$ μs, and total bandwidth $B = N\Delta f = 240$ kHz. Speeds of 120 km/h, 300 km/h, and 500 km/h are chosen to represent cars, high-speed trains, and airplanes, simulating various high-mobility scenarios. LDPC-Hadamard codes use 4^{th}-order Hadamard matrices for the balance of complexity and performance, ease of implementation, and proven error correction capabilities in practical applications and standards [43]. The 3GPP TDL-C model simulates multipath effects through delay and power variations (with $L = 8$ delay taps, maximum delay spread $\tau_{\max} = 3.2$ μs, and exponential decay factor $\alpha = 0.8$), capturing frequency selectivity. It also accounts for Doppler shifts (Jakes spectrum model) and temporal changes, making it ideal for modeling time-frequency doubly selective channels [44]. Converting the check node constraints in the Tanner graph of LDPC codes to Hadamard constraints increases the redundancy, resulting in greater resilience to interference during transmission. In the coding process, each information bit is expanded into a larger symbol space, enhancing error correction performance but reducing the overall code rate, leading to simulations being conducted at a very low code rate.

A. EXIT CHART OF QUASI-CYCLIC LDPC CODE

Fig. 6 illustrates the information exchange between the variable and check nodes of the quasi-cyclic LDPC code constructed, as shown by the Extrinsic Information Transfer Chart (EXIT) [45]. Specific steps are provided in Appendix C. In Fig. 6, the red curve represents the information transfer characteristics of check nodes, and the blue curve represents the information transfer characteristics of variable nodes. By observing the "tunnel" (the gap between variable and check node curves), it becomes clear that the two curves gradually converge under high SNR conditions, showing that the decoder eventually reaches the correct solution. Fig. 6 demonstrates that the LDPC code constructed efficiently updates information during iterative decoding, showing good convergence. Furthermore, it indicates that the code achieves low BER and strong decoding performance under high SNR conditions.

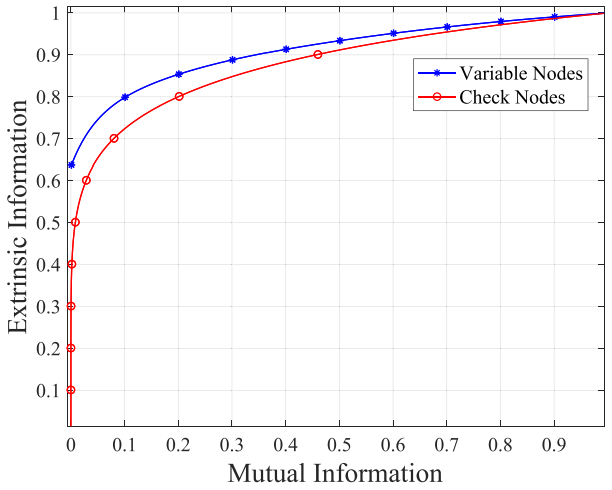

FIGURE 6. EXIT chart of quasi-cyclic LDPC code.

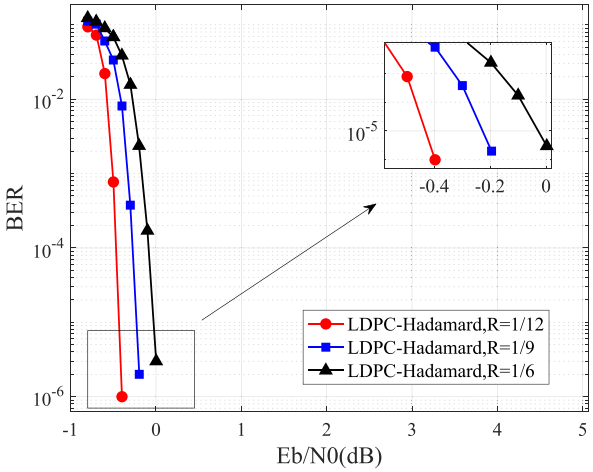

FIGURE 7. LDPC-Hadamard Performance in AWGN, BPSK, R=1/12, 1/9, 1/6.

B. PERFORMANCE COMPARISON OF LDPC-HADAMARD CODES WITH DIFFERENT CODE RATES

Fig. 7 shows the BER of the LDPC-Hadamard code under different code rates with BPSK modulation in an AWGN channel, where the red, blue, and black curves correspond to code rates of 1/12, 1/9, and 1/6, respectively. Different code

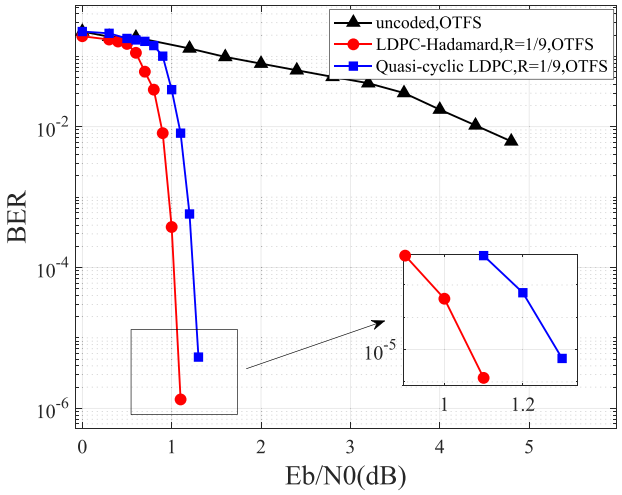

FIGURE 8. Performance of uncoded OTFS, LDPC-Hadamard OTFS, and Quasi-cyclic OTFS in 3GPP TDL-C, BPSK, R=1/9.

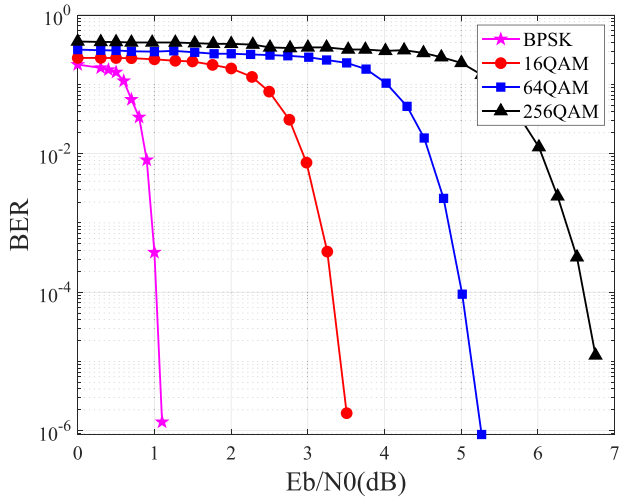

FIGURE 9. LDPC-Hadamard OTFS in 3GPP TDL-C with BPSK, 16QAM, 64QAM, 256QAM.

rates are designed as rate-compatible codes through puncturing techniques. Simulation results show that the punctured LDPC-Hadamard code still exhibits good performance. With a code rate of $R = 1/12$, the LDPC-Hadamard code achieves a BER of 10^{-6} at -0.4 dB, providing a 0.2 dB SNR gain compared to $R = 1/9$, and a 0.4 dB gain compared to $R = 1/6$. Replacing the SPC code in the LDPC code with a Hadamard code accelerates convergence. At very low code rates, the LDPC-Hadamard code exhibits superior performance, particularly regarding error rate reduction.

C. PERFORMANCE COMPARISON OF UNCODED AND CODED OTFS SYSTEMS WITH LDPC VARIANTS

Fig. 8 compares the performance of LDPC-Hadamard codes, quasi-cyclic LDPC codes, and uncoded OTFS systems (represented by red, blue, and black curves, respectively) under 3GPP TDL-C channels using BPSK modulation. The simulation parameters are set as follows: $N = 16$ subcarriers, $M = 128$ time slots, speed $V = 120$ km/h, and code rate $R = 1/9$. Simulation results show that the BER of the uncoded OTFS system is significantly higher than that of the coded systems, indicating that coding provides substantial noise immunity and error correction improvements for the OTFS system. Besides, in the lower E_b/N_0 range (0 dB to 1.3 dB), the BER drops rapidly as E_b/N_0 increases, demonstrating the strong noise resistance of the Hadamard code structure at low SNRs. Especially at an E_b/N_0 of 1.1dB, the LDPC-Hadamard coded OTFS system shows approximately a 0.2dB gain over the quasi-cyclic LDPC coded OTFS system.

D. PERFORMANCE OF LDPC-HADAMARD CODE-ASSISTED OTFS SYSTEM UNDER DIFFERENT MODULATION TECHNIQUES

Fig. 9 shows the performance of the LDPC-Hadamard coded OTFS system under different modulation schemes. The pink, red, blue, and black curves represent the performance of the BPSK, 16QAM, 64QAM, and 256QAM modulation schemes, respectively, using the following parameters: $N = 16$, $M = 128$, $V = 120$ km/h, $R = 1/12$, and a 3GPP TDL-C channel. Fig. 9 indicates a gradual decrease in BER as the SNR increases. The LDPC-Hadamard-assisted OTFS system, utilizing 16QAM modulation, demonstrates excellent noise resistance, achieving low BER at lower SNR levels. At $E_b/N_0 = 3.5$ dB, a BER of 10^{-6} is reached, indicating highly reliable transmission performance. In contrast, 64QAM and 256QAM exhibit inferior BER performance compared to 16QAM at low SNR. However, the required SNR increase for 64QAM and 256QAM is moderate, enabling both modulation schemes to meet system BER requirements at slightly higher SNR levels while offering higher data rates. Consequently, 64QAM and 256QAM are well-suited for applications requiring high bandwidth and data rates.

E. BER PERFORMANCE OF LDPC-HADAMARD AND QUASI-CYCLIC LDPC CODES IN OTFS SYSTEMS UNDER VARYING MOBILITY SPEEDS

Fig. 10 illustrates the BER of LDPC-Hadamard and quasi-cyclic LDPC coded OTFS systems under 64QAM modulation across different mobility speeds (120 km/h, 300 km/h, and 500 km/h) over a 3GPP TDL-C channel. The red and blue curves represent the LDPC-Hadamard and quasi-cyclic LDPC coded systems, respectively, with different marker shapes indicating various speeds. Fig. 10 demonstrates that regardless of the mobility speed, the LDPC-Hadamard-assisted OTFS system consistently maintains an extremely low BER under low SNR conditions, even at higher speeds. The simulation results show that the LDPC-Hadamard code consistently outperforms the quasi-cyclic LDPC code in BER at all speeds, achieving significantly lower BER at the

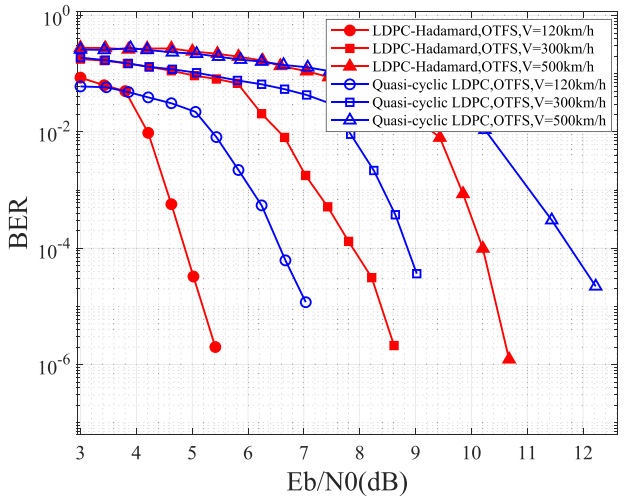

FIGURE 10. Comparison of LDPC-Hadamard and Quasi-cyclic LDPC in OTFS (3GPP TDL-C, 120/300/500 km/h, R=1/9, 64QAM).

same SNR. Therefore, the LDPC-Hadamard code demonstrates enhanced robustness in assisting the OTFS system, making it a promising candidate for high-speed and low-SNR communication scenarios where reliable performance is critical.

V. CONCLUSION

This paper proposes an LDPC-Hadamard code to enhance OTFS system performance in high-speed mobility scenarios. Additionally, a quasi-cyclic LDPC construction algorithm is introduced to reduce encoding complexity, and an LLR-BP decoding algorithm is proposed to minimize computational costs without compromising accuracy. Simulation results show that the proposed LDPC-Hadamard code achieves a BER of 10^{-6} at 1.1dB SNR under high Doppler shifts, outperforming conventional quasi-cyclic LDPC schemes by 0.2dB SNR gain in time-varying channels. However, Current code design assumes block-fading channels, which degrade when Doppler spread exceeds 15% of OTFS frame duration. This paper focuses on the design of binary LDPC codes and regular Hadamard codes. Future work will explore using non-binary LDPC and irregular Hadamard codes to enhance system adaptability and scalability.

APPENDIX A
THE DECODING ALGORITHM FOR HADAMARD CODE

The LLR of the posterior probability for the Hadamard code is

$$\begin{aligned} L(c_i|\mathrm{y}) &= \log \frac{\Pr(c_i=+1 \mid \mathrm{y})}{\Pr(c_i=-1 \mid \mathrm{y})} \\ &= \log \frac{\Pr(\mathrm{y} \mid c_i=+1)}{\Pr(\mathrm{y} \mid c_i=-1)} \\ &= \log \frac{\sum_{c_i=+1} \Pr\left(\mathrm{y} \mid \mathrm{c}\right)}{\sum_{c_i=-1} \Pr\left(\mathrm{y} \mid \mathrm{c}\right)} \\ &= \log \frac{\sum_{\mathrm{H}_{i,j}=+1} e^{-\frac{\left\|\mathrm{h}^j-\mathrm{y}\right\|^2}{2\sigma^2}} + \sum_{\mathrm{H}_{i,j}=1} e^{-\frac{\left\|-\mathrm{h}^j-\mathrm{y}\right\|^2}{2\sigma^2}}}{\sum_{\mathrm{H}_{i,j}=-1} e^{-\frac{\left\|\mathrm{h}^j-\mathrm{y}\right\|^2}{2\sigma^2}} + \sum_{\mathrm{H}_{i,j}=+1} e^{-\frac{\left\|-\mathrm{h}^j-\mathrm{y}\right\|^2}{2\sigma^2}}} \\ &= \log \frac{\sum_{\mathrm{H}_{i,j}=\pm 1} \gamma\left(\pm \mathrm{h}^j\right)}{\sum_{\mathrm{H}_{i,j}=\mp 1} \gamma\left(\pm \mathrm{h}^j\right)}. \end{aligned} \tag{39}$$

APPENDIX B
FAST HADAMARD TRANSFORM

The Hadamard matrix-vector product frequently operates in the APP decoding algorithm for Hadamard code. For an nth-order Hadamard matrix, calculate $y = x \cdot H_n$ as typically performed. Leveraging Hadamard matrix properties can greatly reduce computational complexity. Then,

$$\begin{aligned} \mathrm{y} &= \begin{bmatrix} x' & x'' \end{bmatrix} \begin{bmatrix} H_{n/2} & H_{n/2} \\ H_{n/2} & -H_{n/2} \end{bmatrix} \\ &= \begin{bmatrix} x'H_{n/2} & x''H_{n/2} \end{bmatrix} \begin{bmatrix} I_{n/2} & I_{n/2} \\ I_{n/2} & -I_{n/2} \end{bmatrix}. \end{aligned} \tag{40}$$

Similarly, $H_{n/2}$ can be recursively selected. Thus, the process is similar to the Fast Fourier Transform (FFT), which we refer to as the Fast Hadamard Transform.

APPENDIX C
EXIT CHART

EXIT chart is based on visually analyzing the information exchanged by the iterative decoder during each iteration. It quantifies the external information transfer in each decoder module (e.g., variable and check nodes in LDPC code) to analyze the decoder's convergence performance. Specific steps of the EXIT diagram:

a) *Initialize mutual information:* Start the first iteration of the variable node decoder by assuming an initial set of external information (mutual information values).
b) *Variable node decoder computation:* The variable node decoder computes new extrinsic information based on the initial input, generating the mutual information curve that illustrates its operation under different signal-to-noise ratios or prior conditions. The extrinsic information from the variable nodes is then passed to the check node decoder for the next iteration.
c) *Check node decoder computation:* The check node decoder calculates new extrinsic information based on the data from the variable node decoder, which is used to generate the mutual information curve for the check nodes. The mutual information curve shows how the check nodes adjust their decoding output under different input conditions. The extrinsic information from the check nodes is then passed back to the variable node decoder for the next iteration.
d) *Iterative update:* In each iteration, variable nodes and check nodes alternately calculate and exchange information, updating the mutual information curves. As the

curves gradually converge, the decoder's ability to find the correct decoding result can be assessed.

ACKNOWLEDGMENT

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.

REFERENCES

[1] T. Y. Al-Naffouri, K. Z. Islam, N. Al-Dhahir, and S. Lu, "A model reduction approach for OFDM channel estimation under high mobility conditions," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2181–2193, Apr. 2010.

[2] S. A. Matin and L. B. Milstein, "OFDM system performance, variability and optimality with design imperfections and channel impediments," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 381–397, Jan. 2021.

[3] Z. Wei et al., "Orthogonal time-frequency space modulation: A promising next-generation waveform," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 136–144, Aug. 2021.

[4] W. Yuan, Z. Wei, S. Li, J. Yuan, and D. W. K. Ng, "Integrated sensing and communication-assisted orthogonal time frequency space transmission for vehicular networks," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 6, pp. 1515–1528, Nov. 2021.

[5] W. Yuan, Z. Wei, J. Yuan, and D. W. K. Ng, "A simple variational Bayes detector for orthogonal time frequency space (OTFS) modulation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7976–7980, Jul. 2020.

[6] Z. Wei, W. Yuan, S. Li, J. Yuan, and D. W. K. Ng, "Off-grid channel estimation with sparse Bayesian learning for OTFS systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7407–7426, Sep. 2022.

[7] T. P. Zieliński et al., "Wireless OTFS-based integrated sensing and communication for moving vehicle detection," *IEEE Sensors J.*, vol. 24, no. 5, pp. 6573–6583, Mar. 2024.

[8] W. Yuan et al., "New delay doppler communication paradigm in 6G era: A survey of orthogonal time frequency space (OTFS)," *China Commun.*, vol. 20, no. 6, pp. 1–25, 2023.

[9] M. Li, W. Liu, and J. Lei, "A review on orthogonal time–frequency space modulation: State-of-art, hotspots and challenges," *Comput. Netw.*, vol. 224, 2023, Art. no. 109597.

[10] Q. Li, Y. Gong, F. Meng, Z. Li, L. Miao, and Z. Xu, "Residual learning based channel estimation for OTFS system," in *Proc. 2022 IEEE/CIC Int. Conf. Commun. China*, 2022, pp. 275–280.

[11] Y. Gong, Q. Li, L. Liu, F. Meng, X. Hou, and Z. Xu, "Exploring the performance of the DL-based OTFS detection scheme in internet of vehicles," *Veh. Commun.*, vol. 43, 2023, Art. no. 100626.

[12] K. Zhu, F. Zhang, L. Jiao, B. Xue, and L. Zhang, "Client selection for federated learning using combinatorial multi-armed bandit under long-term energy constraint," *Comput. Netw.*, vol. 250, 2024, Art. no. 110512.

[13] Y. Gong, Q. Li, F. Meng, X. Li, and Z. Xu, "ViterbiNet-based signal detection for OTFS system," *IEEE Commun. Lett.*, vol. 27, no. 3, pp. 881–885, Mar. 2023.

[14] Y. Ma, G. Ma, B. Ai, J. Liu, N. Wang, and Z. Zhong, "OTFCS-modulated waveform design for joint grant-free random access and positioning in C-V2X," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 1, pp. 103–119, Jan. 2024.

[15] A. Nordio, C. F. Chiasserini, and E. Viterbo, "Joint communication and sensing in OTFS-based UAV networks," *IEEE Trans. Veh. Technol.*, 2025.

[16] J. Shi, J. Hu, Y. Yue, X. Xue, W. Liang, and Z. Li, "Outage probability for OTFS based downlink LEO satellite communication," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3355–3360, Mar. 2022.

[17] L. Gaudio, G. Colavolpe, and G. Caire, "OTFS vs. OFDM in the presence of sparsity: A fair comparison," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4410–4423, Jun. 2022.

[18] X. Huang, A. Farhang, and R.-R. Chen, "Channel estimation and turbo equalization for coded OTFS and OFDM: A comparison," *IEEE Wireless Commun. Lett.*, vol. 12, no. 9, pp. 1613–1617, Sep. 2023.

[19] L. Xiao, S. Li, Y. Qian, D. Chen, and T. Jiang, "An overview of OTFS for Internet of Things: Concepts, benefits, and challenges," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7596–7618, May 2022.

[20] W. Yuan, Z. Wei, S. Li, R. Schober, and G. Caire, "Orthogonal time frequency space modulation—Part III: ISAC and potential applications," *IEEE Commun. Lett.*, vol. 27, no. 1, pp. 14–18, Jan. 2023.

[21] W. Yuan, S. Li, Z. Wei, J. Yuan, and D. W. Kwan Ng, "Bypassing channel estimation for OTFS transmission: An integrated sensing and communication solution," in *Proc. 2021 IEEE Wireless Commun. Netw. Conf. Workshops*, 2021, pp. 1–5.

[22] E. Erdemir, T.-Y. Tung, P. L. Dragotti, and D. Gündüz, "Generative joint source-channel coding for semantic image transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 8, pp. 2645–2657, Aug. 2023.

[23] C. Yue, V. Miloslavskaya, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, "Efficient decoders for short block length codes in 6G URLLC," *IEEE Commun. Mag.*, vol. 61, no. 4, pp. 84–90, Apr. 2023.

[24] G. Xie, C. Yang, Y. Feng, G. Liu, and B. Dai, "Secure finite block-length coding schemes for reconfigurable intelligent surface aided wireless channels with feedback," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2931–2946, May 2023.

[25] S. Li, J. Yuan, W. Yuan, Z. Wei, B. Bai, and D. W. K. Ng, "Performance analysis of coded OTFS systems over high-mobility channels," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6033–6048, Sep. 2021.

[26] X. Li and T. Yang, "Iterative soft cancellation with element-wise extrinsic information for coded OTFS system," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2889–2894, Feb. 2024.

[27] Z. Liu, J. Lei, and B. Shu, "Research on polar-OTFS technology in high-speed mobile communication scenarios," in *Proc. 13th Int. Conf. Commun., Circuits Syst.*, 2024, pp. 244–249.

[28] H. Tian, D. Zhao, and H. Jia, "Research on nonbinary LDPC-OTFS scheme in high mobile communication scenarios," in *Proc. 2021 Int. Conf. Eng. Emerg. Technol.*, 2021, pp. 1–6.

[29] J. Wu, W. Chen, and A. Ephremides, "Achieving extremely low latency: Incremental coding for real-time applications," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4453–4467, Aug. 2023.

[30] D. Han, J. Park, Y. Lee, H. V. Poor, and N. Lee, "Block orthogonal sparse superposition codes for ultra-reliable low-latency communications," *IEEE Trans. Commun.*, vol. 71, no. 12, pp. 6884–6897, Dec. 2023.

[31] C. Yue, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, "Channel coding and decoding schemes for URLLC," in *Ultra-Reliable Low- Latency Communications (URLLC) Theory Practice: Advances in 5G Beyond*. Hoboken, NJ, USA: Wiley, 2023, pp. 119–168.

[32] P.-W. Zhang, F. C. Lau, and C.-W. Sham, "Protograph-based LDPC hadamard codes," *IEEE Trans. Commun.*, vol. 69, no. 8, pp. 4998–5013, Aug. 2021.

[33] G. Yue, L. Ping, and X. Wang, "Generalized low-density parity-check codes based on Hadamard constraints," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 1058–1079, Mar. 2007.

[34] W. Ullah, F. Yang, and D. N. K. Jayakody, "OTFS modulated massive MIMO with 5G NR LDPC coding: Trends, challenges and future directions," *Comput. Netw.*, vol. 254, 2024, Art. no. 110751.

[35] Y. Ma et al., "Characteristics of channel spreading function and performance of OTFS in high-speed railway," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 7038–7054, Oct. 2023.

[36] M. Qian, F. Ji, Y. Ge, M. Wen, X. Cheng, and H. V. Poor, "Block-wise index modulation and receiver design for high-mobility OTFS communications," *IEEE Trans. Commun.*, vol. 71, no. 10, pp. 5726–5739, Oct. 2023.

[37] Y. Gong, Q. Li, F. Meng, X. Li, and Z. Xu, "Data-driven deep learning for OTFS detection," *China Commun.*, vol. 20, no. 1, pp. 88–101, 2023.

[38] D. Zhou et al., "OTFS-based robust MMSE precoding design in over-the-air computation," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13932–13937, Sep. 2024.

[39] C. Borrelli, "IEEE 802.3 cyclic redundancy check," Xilinx, Inc., Application Note XAPP209, 2001. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp209.pdf

[40] Z. Ding, R. Schober, P. Fan, and H. V. Poor, "OTFS-Noma: An efficient approach for exploiting heterogenous user mobility profiles," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7950–7965, Nov. 2019.

[41] Y. Liu, W. Tang, and D. G. Mitchell, "Efficient implementation of a threshold modified min-sum algorithm for LDPC decoders," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 67, no. 9, pp. 1599–1603, Sep. 2020.
[42] I. Akila, S. Uma, and P. Poojitha, "Performance analysis of OTFS scheme for TDL and CDL 3GPP channel models," in *Proc. Expert Clouds Appl.: Proc. ICOECA*, 2022, pp. 737–747.
[43] G. Goelman, "Fast 3D T2-weighted MRI with Hadamard encoding in the slice select direction," *Magn. Reson. Imag.*, vol. 18, no. 8, pp. 939–945, 2000.
[44] H. Yin, X. Wei, Y. Tang, and K. Yang, "Diagonally reconstructed channel estimation for MIMO-AFDM with inter-doppler interference in doubly selective channels," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 14066–14079, Oct. 2024.
[45] H. Li, H. Xu, C. Chen, and B. Bai, "Efficient construction of quasi-cyclic LDPC codes with multiple lifting sizes," *IEEE Commun. Lett.*, vol. 28, no. 4, pp. 754–758, Apr. 2024.

YI GONG (Member, IEEE) received the B.S. degree in information and engineering from the Xi'an University of Posts and Telecommunications, Xi'an, Shannxi province, China, in 2013, the M.S. and Ph.D. degree from the School of Information and Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2016 and 2020, respectively, and the dual Ph.D. degree from University of Technology Sydney, Sydney, Australia, in 2022. She is currently with the School of Information Communication Engineering, Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University, Beijing, China, where she is also an Associate Professor. Her research interests include non-linear MIMO, OTFS, and blockchain in wireless communication.

ZEHAO LI received the B.S. degree from the North China Institute of Aerospace Engineering, Langfang, China, in 2021. He is currently working toward the M.S. degree with Beijing Information Science and Technology University, Beijing, China. He is also with the School of Information Communication Engineering, Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University. His research interests include OTFS and channel coding.

LICHUN YANG received the B.S. degree in communication engineering from Northwestern Polytechnical University, Xi'an, China, in 2006, and the M.S. degree in optical communication from the Xi'an University of Posts and Telecommunications, in 2009. From 2009 to 2013, he was a Regional Director with ZTE Corporation. Since 2014, he has been a Co-founder, the Vice President, and CMO with Shenzhen Penguin Network Technology Company, Ltd. He is currently with Shenzhen Penguin Network Technology Company, Shenzhen, China.

LU TIAN (Member, IEEE) received the B.E. and Ph.D. degrees in the major of information and communication engineering from the Beijing Institute of Technology, Beijing, China, in 2011 and 2017, respectively. From 2017 to 2020, she was an Engineer with the China Academy of Electronics and Information Technology for the development of satellite communication systems. She is currently an Assistant Professor with the Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University, Beijing, China. Her research interests include interference suppression and covert communication.

JUN MIAO received the Ph.D. degree in flight vehicle design and engineering from Beihang University (BUAA), Beijing, China, in 2019. He is currently with the Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing. His research interests include spacecraft system simulation and automatic control.

DESHENG ZHANG was born in Anhui, China, in 1980. He received the B.S., M.S., and Ph.D. degrees in communication and information system from the Beijing Institute of Technology, Beijing, China, in 2003, 2006, and 2010, respectively. In 2010, he joined the Institute of Electrics, Chinese Academy of Sciences (IECAS). He is currently an Associate Researcher with Aerospace Information Research Institute, Chinese Academy of Science, Beijing. His research interests include design and evaluation of microwave integrated systems and component devices and array signal processing system.

ZHAN XU (Member, IEEE) received the B.S. degree in information engineering and the M.S. and Ph.D. degrees in communications and information systems from the Beijing Institute of Technology, Beijing, China, in 2004, 2006, and 2010, respectively. He is currently with the School of Information and Communication Engineering and Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University, Beijing. His research interests include wireless communications and digital signal processing.