Low-Complexity OTFS Channel Equalization Based on CLU-MMSE

Jia Haoxiang[1], Zhao Danfeng[1], Xin Yu[2], Hua Jian[2]
(1. College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202601004
https://kns.cnki.net/kcms/detail/34.1294.TN.20260302.1718.004.html, published online March 3, 2026
Manuscript received: 2024-03-18

Abstract: In view of the high computational complexity of traditional linear equalization algorithms in Orthogonal Time Frequency Space (OTFS) systems, a minimum mean square error (MMSE) channel equalization algorithm based on Matrix Chunking Lower and Upper Triangular Decomposition (CLU) is proposed. The proposed algorithm derives the structural properties of the chunked MMSE equalization matrix by leveraging the block diagonal structure of the Cyclic Prefix OTFS (CP-OTFS) time-domain channel matrix and the quasi-band structure of its constituent block matrices. On this basis, triangular decomposition combined with forward and backward substitution is used to avoid matrix inversion. This approach significantly reduces the complexity of the MMSE algorithm without sacrificing its performance.

Keywords: OTFS; equalization algorithm; MMSE; Lower and Upper Triangular Decomposition

Citation (Format 1): Jia H X, Zhao D F, Xin Y, et al. Low-complexity OTFS channel equalization based on CLU-MMSE [J]. ZTE Communications, 2026, 24(1): 16 - 24. DOI: 10.12142/ZTECOM.202601004
Citation (Format 2): H. X. Jia, D. F. Zhao, Y. Xin, et al., "Low-complexity OTFS channel equalization based on CLU-MMSE," *ZTE Communications*, vol. 24, no. 1, pp. 16 - 24, Mar. 2026. doi: 10.12142/ZTECOM.202601004.

1 Introduction

Orthogonal Time Frequency Space (OTFS) is a novel multicarrier modulation technique that characterizes the time-frequency dual-selective channel as an approximately time-invariant channel in the Delay-Doppler (DD) domain through a two-dimensional time-frequency extension[1], making it suitable for highly dynamic scenarios.

However, in actual communication environments, OTFS systems still suffer from inter-code interference, inter-subcarrier interference, and Doppler interference, which require channel equalization techniques to ensure system reliability[2 - 6]. Since the equivalent channel matrix dimension of OTFS in the DD domain is much higher than that of Orthogonal Frequency Division Multiplexing (OFDM), the complexity of channel equalization increases significantly. Therefore, the study of equalization algorithms with low complexity and high performance is key to the development of OTFS modulation systems.

Existing channel equalization techniques for OTFS can be classified into linear and nonlinear types based on design criteria. Linear equalization offers advantages such as simple structure and easy implementation, making it widely used in communication systems. Among various linear equalization algorithms, the minimum mean square error (MMSE) is the most commonly used. However, for an OTFS system with N symbols and M subcarriers, the size of the equivalent channel matrix in both the time domain and the DD domain is $MN \times MN$, and the complexity of the conventional MMSE with matrix inversion is as high as $O((MN)^3)$. To address this problem, various low-complexity linear equalization algorithms based on the properties of the OTFS channel matrix have been proposed. Under ideal pulse conditions, Surabhi et al. [7] proposed an MMSE equalization algorithm with linear complexity by exploiting the doubly circulant property of the OTFS delay-Doppler domain channel matrix. This algorithm reduces the complexity of MMSE to logarithmic levels, but its performance deteriorates severely under practical rectangular pulse waveforms. To address this issue, Tiwari et al. [8] proposed an MMSE algorithm with logarithmic complexity under rectangular pulse waveforms by utilizing the quasi-banded property of the Reduced Cyclic Prefix (RCP)-OTFS time-domain channel matrix, avoiding large matrix inversion through matrix decomposition. However, this algorithm heavily depends on the channel matrix structure and cannot be directly applied to Cyclic Prefix OTFS (CP-OTFS) systems, which offer better compatibility with OFDM. Based on the above analysis, most existing low-complexity linear equalization algorithms for OTFS are MMSE variants developed in different domains, and most rely on the assumption of ideal pulse or RCP-OTFS. Therefore, it is important to further explore practical and effective low-complexity linear equalization algorithms that can be applied

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. KY10800230005.

to CP-OTFS for the implementation of OTFS technology.

Motivated by this, this paper proposes a low-complexity Matrix Chunking Lower and Upper Triangular Decomposition (CLU)-MMSE equalization algorithm for CP-OTFS systems. The algorithm employs matrix chunking combined with lower-upper (LU) decomposition of banded matrices, leveraging the block-diagonal structure of CP-OTFS time-domain channel matrices.

2 OTFS Fundamentals

2.1 OTFS System Model

Traditional wireless communication signals, such as those in OFDM systems, are typically analyzed and processed in the time-frequency domain. In contrast, OTFS modulation introduces the concept of the DD domain, and realizes the mutual conversion between the DD and time-frequency (TF) domains through the two-dimensional Symplectic Finite Fourier Transform (SFFT), as shown in Fig. 1.

In Fig. 1, M and N denote the number of subcarriers and symbols per OTFS frame, respectively. T is the duration of a single symbol, which is the reciprocal of the subcarrier spacing Δf. The bandwidth occupied by an OTFS frame is $B = M\Delta f$, and the frame duration is $T_f = NT$.

The bitstream signals $\boldsymbol{b} = [b_1, b_2, \cdots, b_k]$ generated at the transmitter side are mapped to a transmit signal vector $\boldsymbol{a} = [a_1, a_2, \cdots, a_{MN}]$ via a modulator. If the constellation size is A, the number of bitstream symbols is $K = MN\log_2 A$. By arranging the transmit signal vector $\boldsymbol{a}$ sequentially on the DD-domain grid, the DD-domain OTFS complex signal matrix $\boldsymbol{x}_{\mathrm{DD}} \in \mathbb{C}^{M\times N}$ is obtained. The OTFS modulation is then performed on this matrix. Fig. 2 illustrates the complete block diagram of the OTFS communication system.

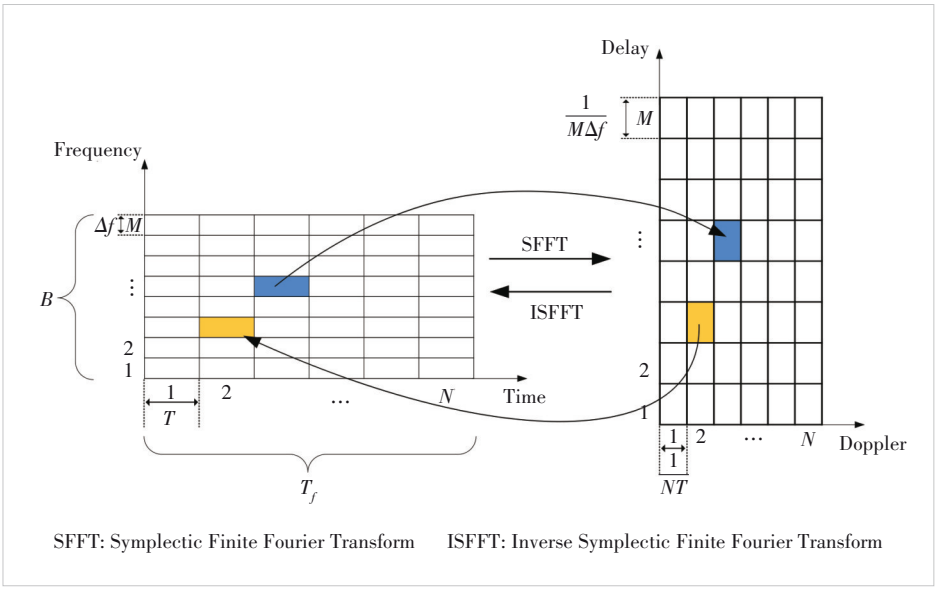


Figure 1. Schematic diagram of the relationship between channel resource transformations in the TF and DD domains

At the transmitter, the transmit symbol $x_{\mathrm{DD}}[k,l]$ at the k-th Doppler and l-th delay grid point in the DD domain undergoes a 2D Inverse Symplectic Finite Fourier Transform (ISFFT) [9], which transforms it into the TF domain symbols $X_{\mathrm{TF}}[n,m]$, where $0 \leqslant n \leqslant N-1$ and $0 \leqslant m \leqslant M-1$. The transformation is given by:

$$X_{\mathrm{TF}}[n,m] = \frac{1}{\sqrt{NM}} \sum_{k=0}^{N-1}\sum_{l=0}^{M-1} x_{\mathrm{DD}}[k,l]\, \mathrm{e}^{\mathrm{j}2\pi\left(\frac{nk}{N}-\frac{ml}{M}\right)} \tag{1}$$

The TF-domain symbol $X_{\mathrm{TF}}[n,m]$ is then converted to the time-domain signal $s(t)$ via the Heisenberg transform, which can be regarded as the process of adding a window to the TF-domain signal after an M-point inverse Fourier transform, typically implemented via the Inverse Fast Fourier Transform (IFFT). The resulting signal is:

$$s(t) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} X_{\mathrm{TF}}[n,m]\, g_{\mathrm{tx}}(t-nT)\mathrm{e}^{\mathrm{j}2\pi m\Delta f(t-nT)} \tag{2}$$

where g_{tx} denotes the transmit shaping pulse with duration $[0, T]$, repeated N times per frame.

The time domain signal arrives at the receiving end through the wireless channel, yielding the received signal $r(t)$:

$$r(t) = \iint h(\tau,\nu)s(t-\tau)\mathrm{e}^{\mathrm{j}2\pi\nu(t-\tau)}\mathrm{d}\nu\,\mathrm{d}\tau + n(t) \tag{3}$$

where $h(\tau,\nu)$ represents the DD domain channel impulse response, and $n(t)$ is additive white Gaussian noise (AWGN) component. $\mathcal{CN}(\boldsymbol{n};0,\sigma_n^2)$, which is an operator indicating that the variable n obeys a complex Gaussian distribution, is calculated as:

$$\mathcal{CN}(n;\mu,\sigma^2) = \frac{1}{\pi\sigma^2}\exp\left(-\frac{|n-\mu|^2}{\sigma^2}\right) \tag{4}$$

Consider a mobile terminal moving at velocity v with carrier frequency f_c. For the i-th propagation path, let θ_i denote the angle of arrival relative to the moving direction, d_i the path length, and c the speed of light. The corresponding delay and Doppler shift are given by:

$$\tau_i = \frac{d_i}{c} \tag{5}$$

$$\nu_i = f_c\frac{v\cos\theta_i}{c} \tag{6}$$

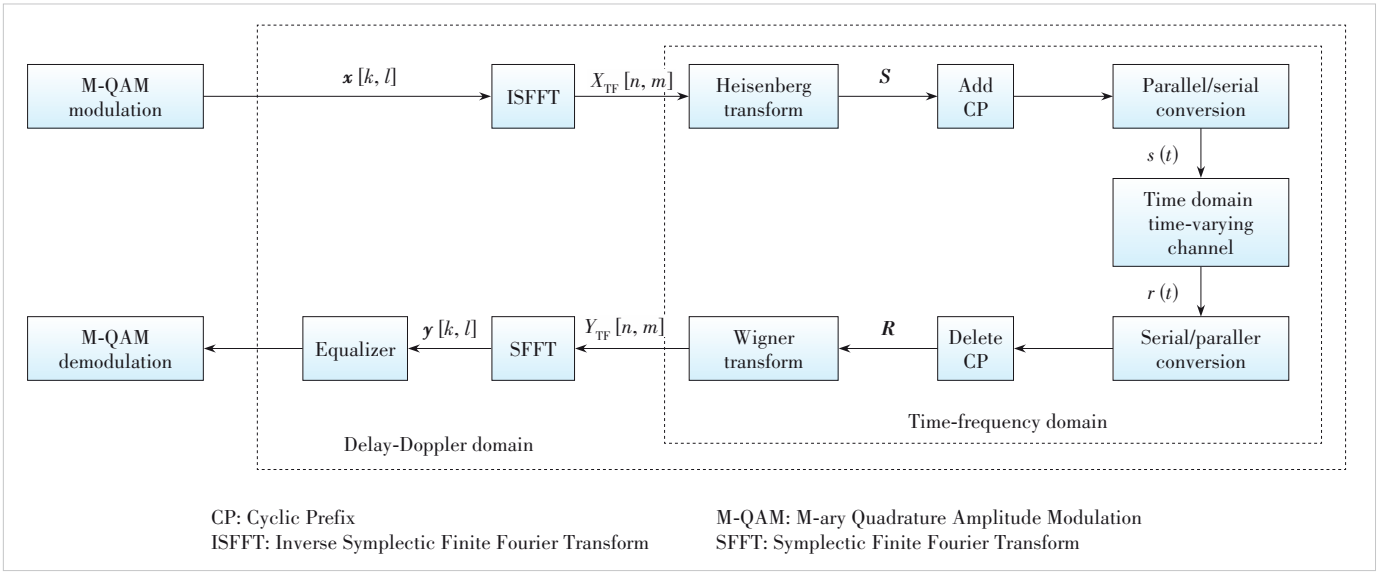


Figure 2. Block diagram of an OTFS communication system

This leads to an expression for the channel impulse response in the DD domain:

$$h(\tau,\nu)=\sum_{i=1}^{P}h_i\delta(\tau-\tau_i)\delta(\nu-\nu_i) \tag{7}$$

where P is the total number of paths and h_i is the complex gain of the i-th path.

The path delays and Doppler shifts in Eqs. (5) and (6) can be obtained by projecting them into the DD domain with resolution of $1/M\Delta f$ and $1/NT$, respectively:

$$\tau_i=\frac{l_{\tau_i}}{M\Delta f},\quad \nu_i=\frac{k_{\nu_i}+\kappa_{\nu_i}}{NT} \tag{8}$$

where l_{τ_i} is the integer delay tap, k_{ν_i} is the integer Doppler taps, and κ_{ν_i} is the fractional Doppler taps with $\kappa_{\nu_i}\in(-0.5,0.5]$. The indices l_{τ_i} and k_{ν_i} represent the path's delay index and Doppler indices in the DD domain, respectively. In general, the delay resolution is sufficient to approximate each path's delay to the nearest sampling point; therefore, there is no need to consider the fractional delay.

Sampling the received signal in Eq. (3) yields the discrete-time signal:

$$r_u=\sum_{i=1}^{P}h_i s(u-l_i)\mathrm{e}^{\mathrm{j}2\pi k_i(u-l_i)}+n_u \tag{9}$$

Unlike time-invariant multipath channels, time-varying multipath channels introduce significant Doppler shifts $(\theta_i)^{k_i-l_i}$, which cause inter-carrier interference (ICI) in the frequency domain and inter-Doppler interference (IDI) in the DD domain. These interference components severely degrade the bit error rate (BER) performance and greatly increase the difficulty of channel estimation and signal detection. For a channel with one direct path and two reflection paths, the received signal r_u is the superposition of these paths, each weighted by its corresponding path gain and shifted by its propagation delay. The direct path arrives first, so its delay can be regarded as zero. The received signal r_u can therefore be expressed as:

$$r_u=h_1s_1(\theta_1)^{k_1}+h_2s_2(\theta_2)^{k_2-l_2}+h_3s_3(\theta_3)^{k_3-l_3}+n_u \tag{10}$$

At the receiver, the received signal is passed through a matched filter to obtain the cross-ambiguity function $A_{g_{rx},y}(t,f)$. Sampling this function at $t=nT$ and $f=m\Delta f$ yields the discrete TF domain received signal $Y_{\mathrm{TF}}[n,m]$, expressed as

$$Y_{\mathrm{TF}}(t,f)=A_{g_{rx},y}(t,f)\triangleq\int g_{rx}^{*}(t-t')r(t')\mathrm{e}^{-\mathrm{j}2\pi f(t-t')}\mathrm{d}t' \tag{11}$$

$$Y_{\mathrm{TF}}[n,m]=Y_{\mathrm{TF}}(t,f)\big|_{t=nT,\ f=m\Delta f} \tag{12}$$

where $g_{rx}(t)$ denotes the receive filter. The operations in Eqs. (11) and (12) together constitute the Wigner transform. Applying the SFFT to the output of the Wigner transform yields the DD domain signal $y_{\mathrm{DD}}[k,l]$ as:

$$y_{\mathrm{DD}}[k,l]=\frac{1}{\sqrt{NM}}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}Y_{\mathrm{TF}}[n,m]\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{nk}{N}-\frac{ml}{M}\right)} \tag{13}$$

The resulting $y_{\mathrm{DD}}[k,l]$ is the demodulated OTFS signal, which serves as the input to the subsequent signal detection stage.

2.2 CP-OTFS

Fig. 3 illustrates the complete implementation of a CP-OTFS

transmitter. As shown, the TF-domain and time-domain processing in CP-OTFS is identical to that in OFDM, except for the addition of a CP to each symbol. This structural similarity renders CP-OTFS highly compatible with OFDM. Consequently, CP-OTFS can be implemented directly on existing OFDM systems by simply adding pre- and post-processing modules, facilitating the rapid deployment of OTFS technology.

The time-domain channel matrix for CP-OTFS can be expressed as[10]:

$$\boldsymbol{H}_t^{\mathrm{CP}} = \begin{bmatrix} \boldsymbol{H}_1 & & & \\ & \boldsymbol{H}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{H}_N \end{bmatrix} \tag{14},$$

where $\boldsymbol{H}_n \in \mathbb{C}^{M\times M}$ is the n-th chunked submatrix, calculated as

$$\boldsymbol{H}_n = \sum_{i=1}^{P} h_i \boldsymbol{\Pi}_M^{l_{\tau_i}} \boldsymbol{\Delta}^{k_{\nu_i},n} \tag{15},$$

where $\boldsymbol{\Pi}_M^{l_{\tau_i}}$ is the forward cyclic shift matrix and $\boldsymbol{\Delta}^{k_{\nu_i},n}$ is the diagonal phase matrix of dimension M, defined as:

$$\boldsymbol{\Pi} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}_{M\times M}$$

$$\boldsymbol{\Delta}^{(k_i)} = \begin{bmatrix} \mathrm{e}^{\mathrm{j}2\pi k_{\nu_i}\frac{[(n-1)M(0)]}{M}} & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \mathrm{e}^{\mathrm{j}2\pi k_{\nu_i}\frac{[(n-1)M-1]}{M}} \end{bmatrix}_{M\times M} \tag{16}.$$

In Eq. (14), the omitted blocks of $\boldsymbol{H}_t^{\mathrm{CP}}$ are zero matrices.

From Eq. (14) and Eq. (16), it can be seen that the time-domain channel matrix of CP-OTFS is a block diagonal matrix with N submatrices. Therefore, the structure of the CP-OTFS time-domain matrix under rectangular pulse conditions can be obtained as shown in Fig. 4.

Consequently, the time-domain channel matrix of CP-OTFS under rectangular pulse shaping assumes the block-diagonal form illustrated in Fig. 4 and given by:

$$\boldsymbol{H}_t^{\mathrm{CP}} = \mathrm{diag}\left\{\boldsymbol{H}_1^{\mathrm{CP}},\cdots,\boldsymbol{H}_N^{\mathrm{CP}}\right\} \tag{17},$$

where each diagonal block $\boldsymbol{H}_i^{\mathrm{CP}} \in \mathbb{C}^{M\times M}$, $1 \leqslant i \leqslant N$, is generally distinct.

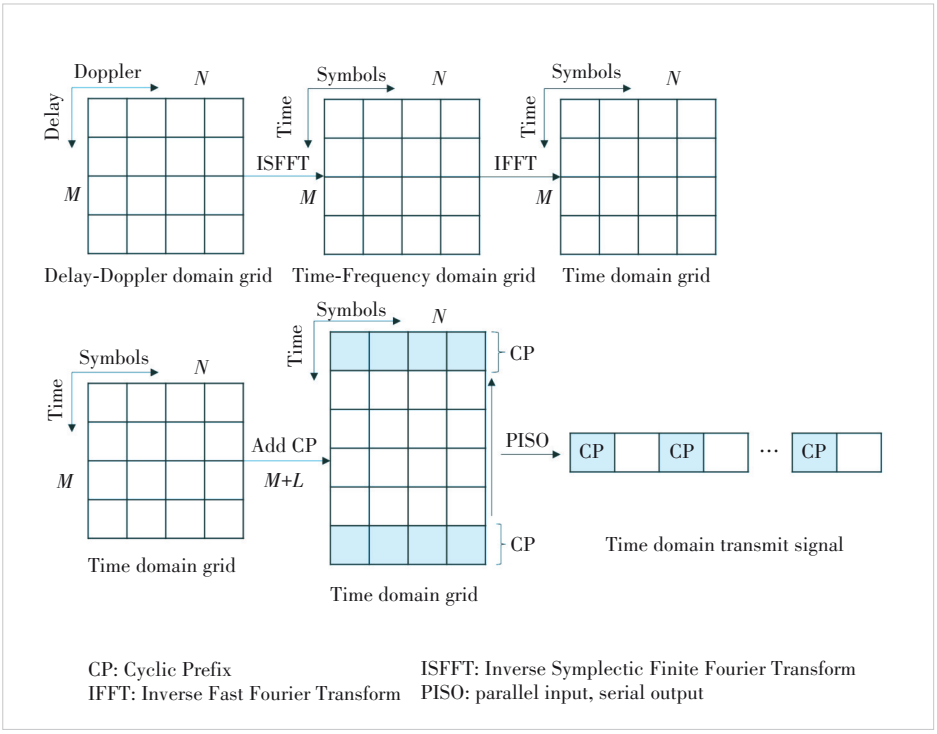


Figure 3. Schematic diagram of a CP-OTFS transmitter

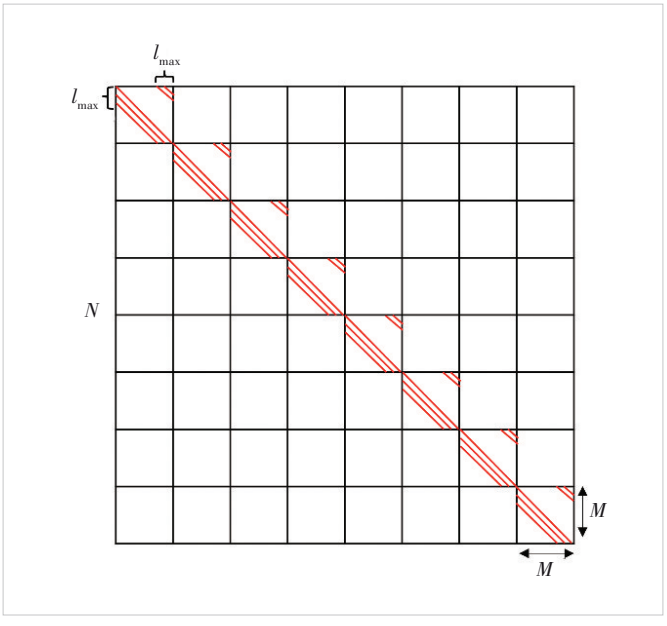


Figure 4. Schematic of CP-OTFS time domain channel matrix structure

3 Channel Equalization Algorithm Based on CLU-MMSE

3.1 MMSE Equalization Algorithm

Let the number of OTFS symbols be N and the number of subcarriers be M. The input-output relationship between the transmitter and receiver can be expressed as:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{18}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^{MN \times 1}$ denote the transmitted and received symbol vectors in the time or DD domain, $\boldsymbol{H} \in \mathbb{C}^{MN \times MN}$ is the equivalent channel matrix in the corresponding domain, and $\boldsymbol{n} \in \mathbb{C}^{MN \times 1}$ is the additive white Gaussian noise vector.

Equalization aims to recover the transmitted signal by processing the received signal, e.g., through interference cancellation. The equalized output signal can be expressed as:

$$\hat{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{y} \tag{19}$$

where $\boldsymbol{G} \in \mathbb{C}^{MN \times MN}$ is the linear transformation equalization matrix.

MMSE equalization derives the equalization matrix by minimizing the mean square error between the transmitted signal and the equalized signal. The estimated signal at the MMSE equalization output is given by:

$$\begin{aligned}\hat{\boldsymbol{x}}_{\text{MMSE}} &= \boldsymbol{G}_{\text{MMSE}}\boldsymbol{y} = \left(\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I}_{MN}\right)^{-1}\boldsymbol{H}^H\boldsymbol{y} = \\ &\boldsymbol{x} + \left(\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I}_{MN}\right)^{-1}\boldsymbol{H}^H\boldsymbol{n}\end{aligned} \tag{20}$$

where σ^2 denotes the noise variance.

Direct matrix inversion in MMSE involves large-scale matrices, resulting in a computational complexity of $\mathrm{O}((MN)^3)$ complex multiplications.

3.2 CLU-MMSE Equalization Algorithm

Substituting Eq. (17) into Eq. (20) yields the time-domain MMSE equalization matrix for CP-OTFS:

$$\begin{aligned}&\boldsymbol{G} = ((\boldsymbol{H}_t^{\text{CP}})^H\boldsymbol{H}_t^{\text{CP}} + \sigma^2\boldsymbol{I}_{MN})^{-1}(\boldsymbol{H}_t^{\text{CP}})^H = \\ &\left(\begin{bmatrix}(\boldsymbol{H}_1^{\text{CP}})^H & & \\ & (\boldsymbol{H}_2^{\text{CP}})^H & \\ & & \ddots \\ & & & (\boldsymbol{H}_N^{\text{CP}})^H\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{H}_1^{\text{CP}} & & \\ & \boldsymbol{H}_2^{\text{CP}} & \\ & & \ddots \\ & & & \boldsymbol{H}_N^{\text{CP}}\end{bmatrix} + \sigma^2\boldsymbol{I}_{MN}\right)^{-1}\cdot(\boldsymbol{H}_t^{\text{CP}})^H = \\ &\begin{bmatrix}(\boldsymbol{H}_1^{\text{CP}})^H\boldsymbol{H}_1^{\text{CPp}} + \sigma^2\boldsymbol{I}_M & & & \\ & (\boldsymbol{H}_2^{\text{CP}})^H\boldsymbol{H}_2^{\text{CP}} + \sigma^2\boldsymbol{I}_M & & \\ & & \ddots & \\ & & & (\boldsymbol{H}_N^{\text{CP}})^H\boldsymbol{H}_N^{\text{CP}} + \sigma^2\boldsymbol{I}_M\end{bmatrix}^{-1}\cdot(\boldsymbol{H}_t^{\text{CP}})^H = \\ &\begin{bmatrix}(\boldsymbol{G}_1')^{-1} & & \\ & (\boldsymbol{G}_2')^{-1} & \\ & & \ddots \\ & & & (\boldsymbol{G}_N')^{-1}\end{bmatrix}\cdot\begin{bmatrix}(\boldsymbol{H}_1^{\text{CP}})^H & & \\ & (\boldsymbol{H}_2^{\text{CP}})^H & \\ & & \ddots \\ & & & (\boldsymbol{H}_N^{\text{CP}})^H\end{bmatrix}\end{aligned} \tag{21}$$

where $\boldsymbol{G}_i' = (\boldsymbol{H}_i^{\text{CP}})^H\boldsymbol{H}_i^{\text{CP}} + \sigma^2\boldsymbol{I}_M \in \mathbb{C}^{M \times M}$, $i = 1, \cdots, N$.

The result of substituting Eq. (21) into Eq. (19) in a chunked matrix form is given by:

$$\hat{\boldsymbol{x}}_i = (\boldsymbol{G}_i')^{-1}(\boldsymbol{H}_i^{\text{CP}})^H\boldsymbol{y}_i \tag{22}$$

where $\boldsymbol{y}_i, \hat{\boldsymbol{x}}_i \in \mathbb{C}^{M \times 1}$, $i = 1, \cdots, N$, are the i-th subvectors of the time-domain received vector $\boldsymbol{y}_t$ and the equalized estimate vector $\hat{\boldsymbol{x}}$, respectively.

From Eq. (21), the time-domain MMSE equalization matrix $\boldsymbol{G}$ is block diagonal, with each submatrix sharing the same structure. Since the calculation of $\boldsymbol{G}$ requires the inverse of each $\boldsymbol{G}_i'$, the structural properties of the matrix $\boldsymbol{G}_i', i = 1, \cdots, N$ are derived first. Substituting Eq. (14) into Eq. (21), the matrix $\boldsymbol{G}_i'$ can be expressed as:

$$\begin{aligned}&\boldsymbol{G}_i' = (\boldsymbol{H}_i^{\text{CP}})^H\boldsymbol{H}_i^{\text{CP}} + \sigma^2\boldsymbol{I}_M = \\ &\sum_{p=1}^{P} h_p\boldsymbol{\Delta}^{-k_p}\boldsymbol{\Pi}^{-l_p}\sum_{s=1}^{P} h_s'\boldsymbol{\Delta}^{k_s}\boldsymbol{\Pi}^{l_s} + \sigma^2\boldsymbol{I}_M = \\ &\sum_{\substack{p=1 \\ p=s}}^{P}\left(\left|h_p\right|^2 + \sigma^2\right)\boldsymbol{I}_M + \sum_{\substack{p=1 \\ p\neq s}}^{P}\sum_{s=1}^{P} h_p h_s'\boldsymbol{\Pi}^{l_s - l_p}\boldsymbol{\Delta}^{k_s - k_p}\end{aligned} \tag{23}$$

where P is the number of channel paths, h_p and h_s' are the complex channel coefficients of different paths, and l_i and k_i, $i = 1, \cdots, P$, are the delay taps and Doppler taps corresponding to each path, respectively.

Let $l_{\max}$ denote the maximum delay tap and β denote the ceiling of the maximum Doppler tap. Then, the range of $(l_s - l_p)$ is $[-l_{\max}, l_{\max}]$. Therefore, the maximum shifts represented by $\boldsymbol{\Pi}^{l_s - l_p}$ can shift up to $l_{\max}$ positions to the left or right, indicating that the submatrix $\boldsymbol{G}_i'$ is a quasi-banded matrix with a bandwidth of $(2l_{\max} + 1)$, as shown in Fig. 5. The inversion of $\boldsymbol{G}_i'$ can be efficiently performed using matrix factorization algorithms, as described in the following.

First, the LU decomposition of $\boldsymbol{G}_i'$ is illustrated in Fig. 5. Let $Q = M - l_{\max}$; this decomposition process can be expressed in matrix form as:

$$\begin{aligned}&\boldsymbol{G}_i' = \begin{bmatrix}(\boldsymbol{A})_{Q\times Q} & (\boldsymbol{B})_{Q\times l_{\max}} \\ (\boldsymbol{C})_{\alpha\times Q} & (\boldsymbol{D})_{l_{\max}\times l_{\max}}\end{bmatrix} = \\ &\begin{bmatrix}(\boldsymbol{L}_A)_{Q\times Q} & (\boldsymbol{0})_{Q\times l_{\max}} \\ (\boldsymbol{E})_{l_{\max}\times Q} & (\boldsymbol{R})_{l_{\max}\times l_{\max}}\end{bmatrix}\times\begin{bmatrix}(\boldsymbol{U}_A)_{Q\times Q} & (\boldsymbol{F})_{Q\times l_{\max}} \\ (\boldsymbol{0})_{l_{\max}\times Q} & (\boldsymbol{T})_{l_{\max}\times l_{\max}}\end{bmatrix} = \boldsymbol{L}_i\boldsymbol{U}_i\end{aligned} \tag{24}$$

where $\boldsymbol{L}_i$ and $\boldsymbol{U}_i$ are the LU decomposition matrices of $\boldsymbol{G}_i'$, and $\boldsymbol{L}_A$ and $\boldsymbol{U}_A$ are the LU decomposition matrices of the chunking

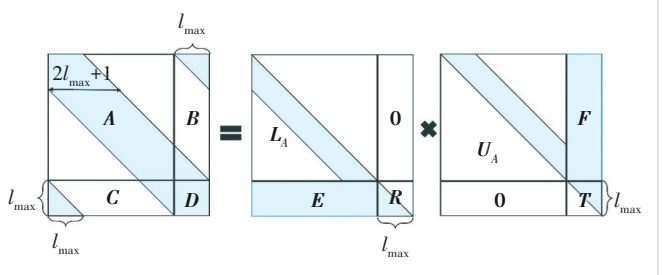


Figure 5. Schematic diagram of the LU decomposition of the chunked matrix

matrix $\boldsymbol{A}$.

The following matrix relationship can be obtained from the above equation:

$$\boldsymbol{A} = \boldsymbol{L}_A \boldsymbol{U}_A \tag{25}$$

$$\boldsymbol{F} = \left(\boldsymbol{L}_A\right)^{-1} \boldsymbol{B} \tag{26}$$

$$\boldsymbol{E} = \boldsymbol{C}\left(\boldsymbol{U}_A\right)^{-1} \tag{27}$$

$$\boldsymbol{RT} = \boldsymbol{D} - \boldsymbol{EF} \tag{28}$$

Therefore, to compute the LU decomposition of $\boldsymbol{G}_i'$, one needs to solve for the matrices $\boldsymbol{L}_A$, $\boldsymbol{U}_A$, $\boldsymbol{E}$, $\boldsymbol{F}$, $\boldsymbol{R}$, and $\boldsymbol{T}$ separately.

From Fig. 5, $\boldsymbol{A} \in \mathbb{C}^{Q \times Q}$ is a standard banded matrix with a non-zero element bandwidth of $2l_{\max} - 1$ and a half-bandwidth of $l_{\max}$. The LU decomposition of the standard banded matrix can be performed using a low-complexity LU decomposition algorithm. The algorithm proceeds in Q steps. For the k-th step ($k = 0,\cdots, Q-1$), only the elements in an $l_{\max} \times l_{\max}$ rectangular window that is active along the diagonal are involved in the operation. The first row and the first column of this window in the k-th step correspond to the k-th row and the k-th column of $\boldsymbol{A}$, respectively. At this point, the k-th column of $\boldsymbol{L}_A$ and the k-th row of $\boldsymbol{U}_A$ can be obtained using Eqs. (29)–(31):

$$\boldsymbol{L}_A(k,k) = 1 \tag{29}$$

$$\boldsymbol{L}_A(k+i,k) = \boldsymbol{A}(k+i,k)/\boldsymbol{A}(k,k) \quad 1 \leqslant i < m \tag{30}$$

$$\boldsymbol{U}_A(k,k+j) = \boldsymbol{A}(k,k+j) \quad 0 \leqslant j < m \tag{31}$$

In the rectangular window of $\boldsymbol{A}$, the elements not directly involved in the current step are updated using Eqs. (30) and (31):

$$\boldsymbol{A}(k+i,k+j) = \boldsymbol{A}(k+i,k+j) - \boldsymbol{L}_A(k+i,k)\boldsymbol{U}_A(k,k+j) \tag{32}$$

where $0 < i < m$, $0 < j < m$.

As k increases, the $l_{\max} \times l_{\max}$ rectangular window moves along the diagonal of $\boldsymbol{A}$ element by element. The corresponding columns and rows of $\boldsymbol{L}_A$ and $\boldsymbol{U}_A$ are computed sequentially, ultimately completing the decomposition.

Using Eq. (28) to solve for $\boldsymbol{R}$ and $\boldsymbol{T}$, we observe that $\boldsymbol{R}$ and $\boldsymbol{T}$ are lower and upper triangular matrices in small dimensions, respectively. They can be directly obtained via LU decomposition of $(\boldsymbol{D} - \boldsymbol{EF})$ using standard Gaussian elimination.

To find the matrix $\boldsymbol{F}$, we first rewrite Eq. (26) as:

$$\boldsymbol{L}_A \boldsymbol{F} = \boldsymbol{B} \tag{33}$$

Since $\boldsymbol{L}_A$ is a known banded lower triangular matrix, Eq. (33) can be transformed into a system of linear equations and solved for $\boldsymbol{F}$ using recursive relations, as shown in Algorithm 1. Similarly, Eq. (27) can be rewritten as $(\boldsymbol{U}_A^T)\boldsymbol{E}^T = \boldsymbol{C}^T$, so $\boldsymbol{E}$ can also be obtained using Algorithm 1.

Algorithm 1. Inversion of standard banded lower triangular matrices and matrix multiplication

Inputs: banded lower triangular matrix $\boldsymbol{L} \in \mathbb{C}^{Q \times Q}$, matrix $\boldsymbol{B} \in \mathbb{C}^{Q \times l_{\tau_{\max}}}$, bandwidth $l = l_{\max}$
Output: $\boldsymbol{F} = (\boldsymbol{L})^{-1}\boldsymbol{B}$
1: for $k = 0$ to $l - 1$ do
2: $\quad \boldsymbol{F}_{0,k} = \boldsymbol{B}_{0,k}/\boldsymbol{L}_{0,0}$
3: for $i = 1$ to $l - 1$ do
4: $\quad \boldsymbol{F}_{i,k} = \boldsymbol{B}_{i,k}/\boldsymbol{L}_{i,i} - \sum_{j=1}^{i} \boldsymbol{B}_{i,i-j}\boldsymbol{F}_{i-j,k}$
5: end for
6: for $i = l$ to $Q - 1$ do
7: $\quad \boldsymbol{F}_{i,k} = \boldsymbol{B}_{i,k}/\boldsymbol{L}_{i,i} - \sum_{j=1}^{l} \boldsymbol{B}_{i,i-j}\boldsymbol{F}_{i-j,k}$
8: end for
9: end for

The LU decomposition matrices $\boldsymbol{L}_i$ and $\boldsymbol{U}_i$ of $\boldsymbol{G}_i'$ can be obtained after computing each chunk submatrix.

Substituting Eq. (24) into Eq. (22) yields:

$$\hat{\boldsymbol{x}}_i = \overbrace{\left(\boldsymbol{U}_i\right)^{-1}\underbrace{\left(\boldsymbol{L}_i\right)^{-1}\underbrace{\left(\boldsymbol{H}_i^{\mathrm{CP}}\right)^H \boldsymbol{y}_i}_{r_1}}_{r_2}}^{r_3}, \quad i = 1,\cdots,N \tag{34}$$

This equation can be computed in three steps. The first step computes $\boldsymbol{r}_1$, exploiting the sparsity of $\boldsymbol{H}_i^{\mathrm{CP}}$ to reduce complexity. The second step computes $\boldsymbol{r}_2$ by leveraging the banded lower triangular matrix structure of $\boldsymbol{L}_i$. In this step, each element of $\boldsymbol{r}_2$ is obtained recursively using forward substitution, as detailed in Algorithm 2. The third step computes $\boldsymbol{r}_3$ using backward substitution, leveraging the upper triangular structure of $\boldsymbol{U}_i$, as shown in Algorithm 3.

Algorithm 2. Forward substitution for banded lower triangular matrices

Inputs: banded lower triangular matrix $\boldsymbol{L} = \boldsymbol{L}_i \in \mathbb{C}^{M \times M}$, vector $\boldsymbol{r}^{(1)} \in \mathbb{C}^{M \times 1}$, bandwidth $l = l_{\max}$, dimension parameter Q
Output: $\boldsymbol{r}^{(2)} = \boldsymbol{L}^{-1}\boldsymbol{r}^{(1)}$
1: $\quad \boldsymbol{r}_0^{(2)} = \boldsymbol{r}_0^{(1)}$
2: for $k = 1$ to $l - 1$ do
3: $\quad \boldsymbol{r}_k^{(2)} = \boldsymbol{r}_k^{(1)} - \sum_{i=1}^{k} \boldsymbol{L}_{k,k-i}\boldsymbol{r}_{k-i}^{(2)}$
4: end for
5: for $k = l$ to $Q - 1$ do
6: $\quad \boldsymbol{r}_k^{(2)} = \boldsymbol{r}_k^{(1)} - \sum_{i=1}^{l} \boldsymbol{L}_{k,k-i}\boldsymbol{r}_{k-i}^{(2)}$
7: end for
8: for Q to $M - 1$ do
9: $\quad \boldsymbol{r}_k^{(2)} = \boldsymbol{r}_k^{(1)} - \sum_{i=1}^{M-1} \boldsymbol{L}_{k,k-i}\boldsymbol{r}_{k-i}^{(2)}$

10: end for

Algorithm 3. Backward substitution for banded upper triangular matrices

Inputs: banded upper triangular matrix $\boldsymbol{U} = \boldsymbol{U}_i \in \mathbb{C}^{M \times M}$, vector $\boldsymbol{r}^{(2)} \in \mathbb{C}^{M \times 1}$, bandwidth $l = l_{\max}$, dimension parameter Q
Output: $\boldsymbol{r}^{(3)} = \boldsymbol{U}^{-1}\boldsymbol{r}^{(2)}$
1: $\boldsymbol{r}^{(3)}_{(M-1)} = \boldsymbol{r}^{(2)}_{(M-1)}/\boldsymbol{U}_{M-1,M-1}$
2: for $k = M-2$ to $M-2l$ do
3: $\boldsymbol{r}^{(3)}_k = \left(\boldsymbol{r}^{(2)}_k/\boldsymbol{U}_{k,k}\right) - \sum_{i=1}^{M-k-1}\boldsymbol{U}_{k,k+i}\boldsymbol{r}^{(3)}_{k+i}$
4: end for
5: for $k = M-2l-1$ to 0 do
6: $\boldsymbol{r}^{(3)}_k = \left(\boldsymbol{r}^{(2)}_k/\boldsymbol{U}_{k,k}\right) - \sum_{i=1}^{l}\boldsymbol{U}_{k,k+i}\boldsymbol{r}^{(3)}_{k+i} - \sum_{r=M-l}^{M-1}\boldsymbol{U}_{k,r}\boldsymbol{r}^{(3)}_r$
7: end for

Both Algorithms 2 and 3 are derived by utilizing the connection between matrix LU decomposition and linear equation systems. This approach reduces the overall complexity of the MMSE algorithm by replacing complex and large matrix inversion operations with simple recursive subtraction and numerical multiplication.

By performing the above operations on N matrices $\boldsymbol{G}'_i$, the N time-domain estimation subvectors $\hat{\boldsymbol{x}}_i$ are computed and merged into the time-domain estimate $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_1^{\mathrm{T}},\cdots,\hat{\boldsymbol{x}}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{MN\times 1}$ in column-wise order.

According to Eq. (2), the signal matrix in the time-frequency domain is transformed via the Heisenberg transform to obtain the time-domain transmit signal matrix $\boldsymbol{S}$:

$$\boldsymbol{S} = \boldsymbol{G}_{\mathrm{tx}}\boldsymbol{F}_M^H(\boldsymbol{F}_M\boldsymbol{X}_{\mathrm{DD}}\boldsymbol{F}_N^H) = \boldsymbol{G}_{\mathrm{tx}}\boldsymbol{X}_{\mathrm{DD}}\boldsymbol{F}_N^H \tag{35}$$

where $\boldsymbol{S} \in \mathbb{C}^{M\times N}$, and $\boldsymbol{G}_{\mathrm{tx}} \in \mathbb{C}^{M\times M}$ is the matrix representation of the transmit window function $g_{\mathrm{tx}}(t)$:

$$\boldsymbol{G}_{\mathrm{tx}} = \mathrm{diag}\left(\left[g_{\mathrm{tx}}[0], g_{\mathrm{tx}}[T/M],\cdots, g_{\mathrm{tx}}[(M-1)T/M]\right]\right) \tag{36}$$

where $\mathrm{diag}(\cdot)$ forms a diagonal matrix from the given vector.

The matrix form of a rectangular pulse waveform can be expressed as a unit matrix. Under this condition, the modulation process at the transmitter side in Eq. (35) can be simplified as:

$$\boldsymbol{S} = \boldsymbol{G}_{\mathrm{tx}}\boldsymbol{F}_M^{\mathrm{H}}(\boldsymbol{F}_M\boldsymbol{X}_{\mathrm{DD}}\boldsymbol{F}_N^{\mathrm{H}}) = \boldsymbol{G}_{\mathrm{tx}}\boldsymbol{X}_{\mathrm{DD}}\boldsymbol{F}_N^{\mathrm{H}} = \boldsymbol{X}_{\mathrm{DD}}\boldsymbol{F}_N^{\mathrm{H}} \tag{37}$$

From Eq. (37), the equalized time-domain signal is transformed to the DD domain as:

$$\hat{\boldsymbol{x}}_{\mathrm{DD}} = \mathrm{vec}(\mathrm{vec}^{-1}_{M\times N}(\hat{\boldsymbol{x}})\boldsymbol{F}_N) \tag{38}$$

Fig. 6 shows the overall flowchart of the CLU-MMSE algorithm for the CP-OTFS system.

4 Analysis of Simulation Results

4.1 Complexity Analysis

Using the number of complex multiplications as a metric, the computational complexity of the proposed CLU-MMSE algorithm for CP-OTFS under rectangular pulses is summarized as follows:

1) The time-domain channel matrix of CP-OTFS is chunked by Eq. (17) and the N matrices $\mathbf{G}'_i$ are computed using Eq. (23). This step requires $\left((P^2-P)(\beta+1)MN + P^2N\right)$ complex multiplication.

2) According to Eq. (24), the LU decomposition of N matrices $\mathbf{G}'_i$ is performed, and the LU factors of the standard banded matrix $\boldsymbol{A}$ are obtained using Eqs. (29)–(30). This step requires $\left((l_{\max}^2 - l_{\max})MN\right)$ complex multiplication operations.

3) For the N matrices $\mathbf{G}'_i$, Algorithm 1 is used to compute $\boldsymbol{E}$ and $\boldsymbol{F}$ from Eqs. (26) and (27). Standard LU decomposition is then applied to Eq. (28) to obtain $\boldsymbol{R}$ and $\boldsymbol{T}$. This step requires a total of $\left((2l_{\max}^2 + 3l_{\max})MN - (5l_{\max}^3 + 3l_{\max}^2 + 1)N\right)$ complex multiplications.

4) Using Eq. (34), $\boldsymbol{r}_1$ is calculated, and $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ are computed according to Algorithms 2 and 3 to obtain the time-domain estimate. This step requires $\left((4l_{\max}+1)MN - (7l_{\max}^2 + 5l_{\max} - 2P)N/2\right)$ complex multiplication operations.

5) The DD-domain estimate is obtained from Eq. (38) by performing an $M\times N$-point Fast Fourier Transform, which requires $(MN\log_2 N)/2$ complex multiplications.

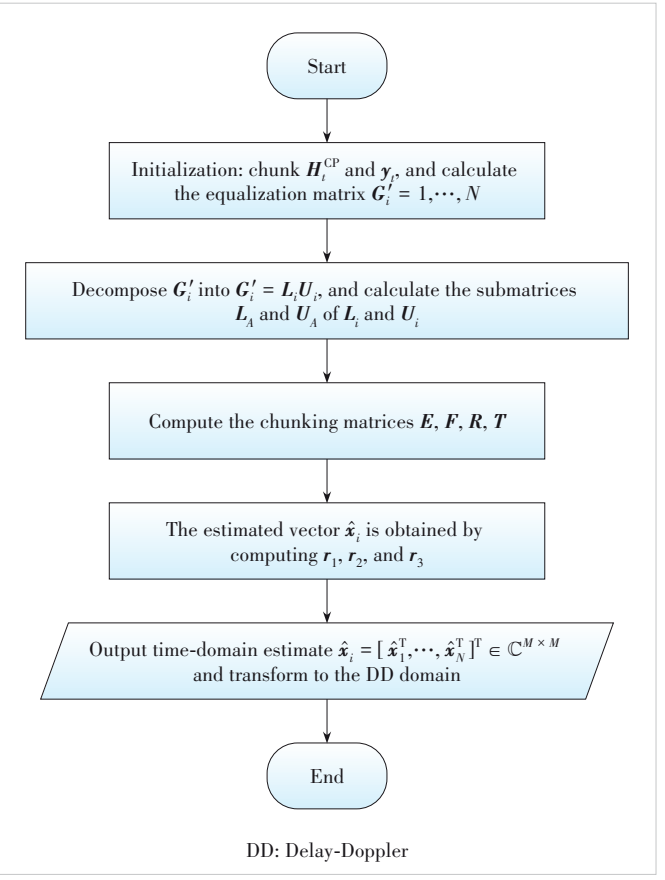


Figure 6. Flowchart of the proposed CLU-MMSE algorithm

In summary, the total complexity of the proposed CLU-MMSE algorithm for CP-OTFS is on the order of $O(MN\log_2 N)$ for large values of MN.

Table 1 compares the computational complexity of several linear equalization algorithms. As shown, the proposed algorithm achieves significantly lower complexity than the traditional matrix-inversion-based MMSE and zero forcing (ZF) algorithms. Moreover, unlike the MMSE equalization algorithm based on the doubly circulant characteristic[11], the proposed algorithm does not rely on the ideal bi-orthogonal pulse assumption, making it more practical.

4.2 Simulation Parameters

To evaluate the bit error rate (BER) performance of the proposed equalization algorithm, simulations are conducted for uncoded OTFS modulation. The Extended Vehicular A (EVA) channel model from LTE is adopted. The detailed simulation parameters are listed in Table 2.

4.3 Simulation Performance

Figs. 7 and 8 compare the BER performance of the proposed algorithm with that of traditional matrix-inversion-based ZF and MMSE algorithms, as well as the MMSE algorithm based on the doubly circulant characteristic. It can be observed that, across different modulation formats and varying values of M and N, the proposed algorithm achieves BER performance nearly identical to that of the traditional matrix-inversion-based MMSE, while significantly outperforming the traditional matrix-inversion-based ZF and the MMSE based on the doubly circulant characteristic.

Table 1. Computational complexity comparison of linear equalization algorithms

Algorithm name	Complexity order
Traditional matrix-inverse ZF	$O((MN)^3)$
Traditional matrix-inverse MMSE	$O((MN)^3)$
MMSE based on doubly circulant characteristic	$O(MN\log_2 MN)$
Proposed CLU-MMSE	$O(MN\log_2 N)$

CLU: Matrix Chunking Lower and Upper Triangular Decomposition
MMSE: minimum mean square error
ZF: zero forcing

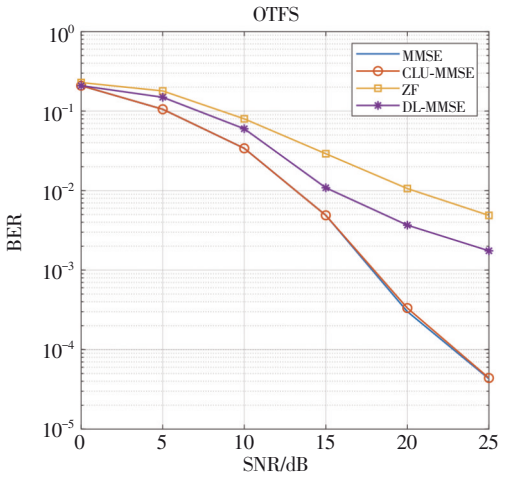
Table 2. Simulation parameters

Parameter	Value
Carrier frequency	4 GHz
Subcarrier spacing (Δf)	15 kHz
Number of subcarriers (M)	32, 16
Number of symbols (N)	16
Modulation type (A)	4QAM, 16QAM
Pulse waveform	Rectangular pulse
Signal path	EVA
Relative movement speed	500 km/h

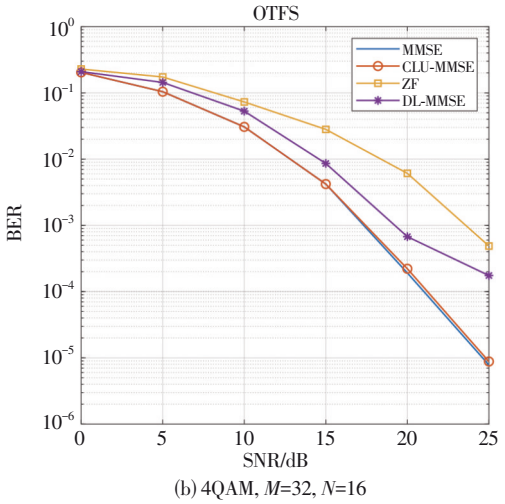
EVA: Extended Vehicular A

5 Conclusions

The traditional OTFS equalization algorithm based on matrix inversion suffers from high computational complexity. In this paper, we address this issue by adopting the time-domain channel matrix with a chunked band structure for equalization. Specifically, the time-domain MMSE equalization matrix is chunked and decomposed multiple times via LU decomposition. Through the factorization into lower and upper triangular matrices, computations involving matrix inversion or matrix-vector/matrix-matrix multiplications can be reformulated as solving linear sys-



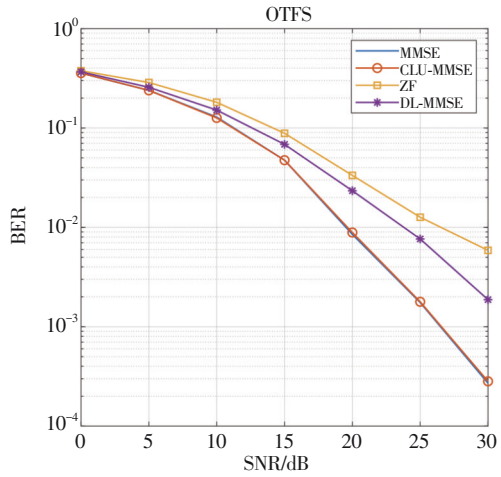
(a) 4QAM, M=N=16



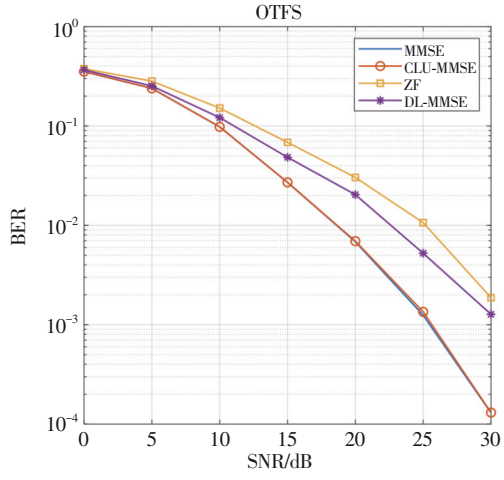
(b) 4QAM, M=32, N=16

BER: bit error rate
CLU: Matrix Chunking Lower and Upper Triangular Decomposition
DL: deep learning
MMSE: minimum mean square error
OTFS: Orthogonal Time Frequency Space
QAM: quadrature amplitude modulation
SNR: signal-to-noise ratio
ZF: zero forcing

Figure 7. BER performance comparison under 4QAM modulation



(a) 16QAM, M=N=16



(b) 16QAM, M=32, N=16

BER: bit error rate
CLU: Matrix Chunking Lower and Upper Triangular Decomposition
DL: deep learning
MMSE: minimum mean square error
OTFS: Orthogonal Time Frequency Space
QAM: quadrature amplitude modulation
SNR: signal-to-noise ratio
ZF: zero forcing

Figure 8. BER performance comparison under 16QAM modulation

tems, thereby avoiding explicit inversion. A recursive algorithm based on forward-backward substitution is then employed to compute each element of the linear equations sequentially. Simulation results demonstrate that the proposed algorithm achieves comparable reliability while reducing the computational complexity of the traditional OTFS-MMSE algorithm, thereby avoiding complex operations such as large-scale matrix multiplication and inversion.

References

[1] He X, Jia H X, Sun Y T, et al. Low-complexity iterative equalization for OTFS based on alternating minimization [J]. Journal of systems engineering and electronics, 2023, 34(4): 851 – 860. DOI: 10.23919/JSEE.2023.000089
[2] Raviteja P, Hong Y, Viterbo E, et al. Effective diversity of OTFS modulation [J]. IEEE wireless communications letters, 2020, 9(2): 249 – 253. DOI: 10.1109/LWC.2019.2951758
[3] Surabhi G D, Augustine R M, Chockalingam A. On the diversity of uncoded OTFS modulation in doubly-dispersive channels [J]. IEEE transactions on wireless communications, 2019, 18(6): 3049 – 3063. DOI: 10.1109/TWC.2019.2909205
[4] Li S Y, Yuan J H, Yuan W J, et al. Performance analysis of coded OTFS systems over high-mobility channels [J]. IEEE transactions on wireless communications, 2021, 20(9): 6033 – 6048. DOI: 10.1109/TWC.2021.3071493
[5] Cheng J Q, Jia C L, Gao H, et al. OTFS based receiver scheme with multi-antennas in high-mobility V2X systems [C]//Proc. IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2020: 1 – 6. DOI: 10.1109/iccworkshops49005.2020.9145313
[6] Shan Y R, Wang F G, Hao Y X, et al. Doppler rate estimation for OTFS via large-scale antenna array [J]. ZTE communications, 2025, 23(1): 115 – 122. doi: 10.12142/ZTECOM.202501015
[7] Surabhi G D, Chockalingam A. Low-complexity linear equalization for OTFS modulation [J]. IEEE communications letters, 2020, 24(2): 330 – 334. DOI: 10.1109/LCOMM.2019.2956709
[8] Tiwari S, Das S S, Rangamgari V. Low complexity LMMSE receiver for OTFS [J]. IEEE communications letters, 2019, 23(12): 2205 – 2209. DOI: 10.1109/LCOMM.2019.2945564
[9] Sun Y T, Jia H X, He X, et al. A low complexity OTFS detection algorithm based on GA-MP [J]. Telecommunication engineering, 2024, 64(2): 288 – 294. DOI: 10.20079/j.issn.1001-893x.220818005
[10] Xiao L X, Li S, Qian Y, et al. An overview of OTFS for Internet of Things: concepts, benefits, and challenges [J]. IEEE Internet of Things journal, 2022, 9(10): 7596 – 7618. DOI: 10.1109/JIOT.2021.3132606
[11] Das S S, Rangamgari V, Tiwari S, et al. Time domain channel estimation and equalization of CP-OTFS under multiple fractional Dopplers and residual synchronization errors [J]. IEEE access, 2021, 9: 10561 – 10576. DOI: 10.1109/ACCESS.2020.3046487

Biographies

Jia Haoxiang (jiahaoxiang@hrbeu.edu.cn) received his BS degree in communication engineering from Shandong University of Science and Technology, China in 2018. He is working toward his PhD degree at the School of Information and Communication Engineering, Harbin Engineering University, China. His main research interests include high-performance coding and modulation schemes.

Zhao Danfeng received his PhD degree in communication systems from Harbin Engineering University, China in 2006, where he is currently a professor. His research interests include network coding, underwater acoustic sensor networks, and high-performance coding and modulation.

Xin Yu graduated from Beijing University of Posts and Telecommunications, China in 2003. He is currently a senior engineer and senior expert in technical pre-research at ZTE Corporation, whose research direction is wireless communication technology. He first proposed the FB-OFDM and GFB-OFDM waveform schemes and has published dozens of papers on waveform technologies. He is now mainly responsible for the pre-research of 6G candidate new waveform technologies.

Hua Jian received his master's degree from Harbin Engineering University, China. He is currently an intermediate engineer at ZTE Corporation. His research interests include phase noise modeling, compensation scheme design, waveform modulation, and other technologies in terahertz communication scenarios.