Delay-Doppler Channel Estimation in DZT-OTFS via Deep Learning in Time-Frequency Domain

Sandesh Rao Mattu and A. Chockalingam
Department of ECE, Indian Institute of Science, Bangalore

***Abstract*—Orthogonal time-frequency space (OTFS) modulation has been shown to be robust in channels with high Doppler spreads. Conventional approach of OTFS signaling involves two steps, viz., conversion from delay-Doppler (DD) domain to time-frequency (TF) domain and then to time domain (TD) for transmission. A more direct approach converts the DD domain symbols to TD directly using the inverse Zak transform in one step, which is performance-wise better for large channel spreads. In this paper, we consider discrete Zak transform based OTFS (DZT-OTFS) and propose a deep learning based low-complexity channel estimation algorithm for fractional DD channels. The proposed approach learns the delay-Doppler matrix (DDM) through training rather than analytically computing it explicitly, and this drastically reduces complexity. A key novelty in the proposed approach is that learning is carried out in the TF domain for DD domain channel estimation. This is motivated by the observation that the values in the channel matrix in TF domain has a smaller swing compared to that in DD domain, which is more favorable for training. Simulation results show that the proposed TF learning based channel estimation achieves almost the same performance as that of a state-of-the-art algorithm in the literature but at a significantly lesser complexity, making the proposed approach practically appealing.**

***Index Terms*—OTFS modulation, discrete Zak transform, DD channel estimation, deep learning, time-frequency learning, fractional delay-Doppler.**

I. Introduction

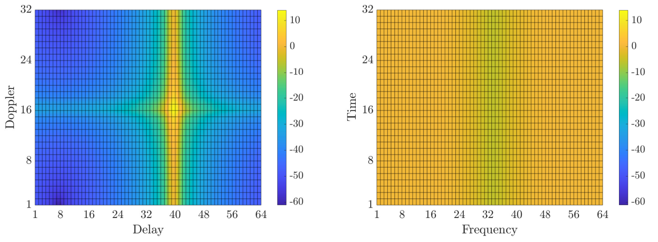
Orthogonal time-frequency space (OTFS) modulation is a promising modulation scheme for next generation wireless systems owing to its resilience to high Doppler channels [1]-[3]. In OTFS, information symbols are multiplexed in the delay-Doppler (DD) domain and the channel is also viewed in the DD domain. In conventional OTFS, the information symbols in the DD domain are converted to time domain for transmission in two steps, viz., conversion from DD domain to TF domain using inverse symplectic finite Fourier transform (ISFFT) and from TF domain to time domain using Heisenberg transform [1]-[8]. Corresponding inverse transforms are carried out at the receiver to convert the received time domain signal to DD domain where signal detection is carried out.

More recently, an alternate way to realize OTFS that converts the DD domain symbols directly to time domain using the inverse Zak transform at the transmitter [9],[10] and time domain signal to DD domain using the Zak transform at the receiver has been shown to achieve better performance compared to the conventional two-step OTFS in large Doppler spreads [11],[12]. Drawing parallel to the discrete implementation of OFDM using discrete Fourier transform, discrete implementation of the Zak based OTFS system can be realized using discrete Zak transform (DZT) [13],[14]. Further, this implementation can be achieved using the efficient discrete Fourier transform of the sub-sampled sequence [13], which makes this approach computationally efficient. The input-output relation for DZT based OTFS (DZT-OTFS) is derived in [14]. The bit error performance of DZT-OTFS has been investigated in [15],[16], with the assumption of perfect channel knowledge at the receiver. Motivated by the above, in this paper, we focus on DD channel estimation for DZT-OTFS systems in channels with fractional DDs. For this, we adopt a learning approach in the TF domain.

Recently, deep learning techniques have been widely used in the field of wireless communications. Advances in the hardware tailor made for training have resulted in lower testing and deployment times. Deep neural networks (DNN) are being used for the implementation of several wireless transceiver functionalities including signal detection [17], beam tracking [18], and channel prediction [19]. DNN based approaches show robustness to model mismatches along with providing complexity advantages [18].

In this paper, we propose a learning approach for DD channel estimation in DZT-OTFS. Specifically, in the proposed approach, we learn a delay-Doppler matrix (DDM) in the estimation algorithm that captures the effect of channel delays and Dopplers through training instead of analytically computing it explicitly. This results in a significant reduction in the computational complexity without compromising much on the performance. A key novelty in the proposed approach lies in the way we learn this matrix. We observed that learning the DDM directly in the DD domain is not favorable for training because of the large swing in the magnitude of the elements of the matrix. This can be seen in Fig. 1a where the swing is in the range of about -30 dB to 10 dB. Whereas, the magnitude swing is much less if the DDM is considered in the TF domain (see Fig. 1b where the magnitude swing is in the range of about -5 dB to 0 dB), and this is favourable for training. Therefore, we train the DDM in the TF domain and take the symplectic finite Fourier transform (SFFT) to obtain the trained matrix in the DD domain. Our simulation results show that the proposed TF learning based channel estimation achieves almost the same performance as that of a state-of-the-art algorithm in the literature [8], but at a significantly lesser complexity. This shows that a judicious adoption of learning approach for the problem at hand can yield efficient solutions. Once trained, the proposed DNN architecture is found to achieve robust performance under various channel

This work was supported in part by the J. C. Bose National Fellowship, Department of Science and Technology, Government of India.



(a) DD domain

(b) TF domain

Fig. 1: Magnitude of elements of DDM in log scale in DD and TF domain.

conditions. The achieved complexity reduction and robustness make the proposed approach practically appealing.

The rest of the paper is organized as follows. the DZT-OTFS system model is presented in Sec. II. The proposed TF learning approach for DD channel estimation is presented in Sec. III. Results and discussions are presented in Sec. IV. Conclusions are presented in Sec. V.

II. DZT-OTFS SYSTEM MODEL

Figure 2 shows the block diagram of DZT-OTFS system. $\mathbf{Z}_x \in \mathbb{A}^{M\times N}$ is the DD domain frame of information symbols to be transmitted, where M and N are the number of delay and Doppler bins, respectively, and $\mathbb{A}$ is the modulation alphabet. The MN symbols are mounted on the DD grid in locations given by $(\frac{mT}{M}, \frac{n\Delta f}{N})$, $m = 0, \cdots, M-1$, $n = 0, \cdots, N-1$, where $\Delta f T = 1$, $M\Delta f = B$, and B is the bandwidth available for communication. $\mathbf{Z}_x$ is converted to time domain (TD) using inverse DZT (IDZT) before transmission. The TD signal vector $\mathbf{x} \in \mathbb{C}^{MN\times 1}$ is obtained from $\mathbf{Z}_x$ using IDZT as [14]

$$\mathbf{x}[m+nM] = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} \mathbf{Z}_x[m,s] e^{j2\pi\frac{ns}{N}}, \quad (1)$$

which can be written as

$$\mathbf{x} = \text{vec}(\mathbf{Z}_x \mathbf{F}_N^H), \quad (2)$$

where $\mathbf{F}_N$ is the N-point unitary discrete Fourier transform (DFT) matrix and $\text{vec}(\cdot)$ is the vectorization operation. Cyclic prefix (CP) of length L_{CP} is added to $\mathbf{x}$ to obtain the vector $\mathbf{s}$ as

$$\mathbf{s}[u] = \begin{cases} \mathbf{x}[(u)_{MN}], & -L_{CP} \le u \le MN-1 \\ 0, & \text{otherwise}, \end{cases}$$

which is converted to a continuous time signal as

$$s(t) = \sum_{u=-L_{CP}}^{MN-1} \mathbf{s}[u] g(t - uT_s), \quad (3)$$

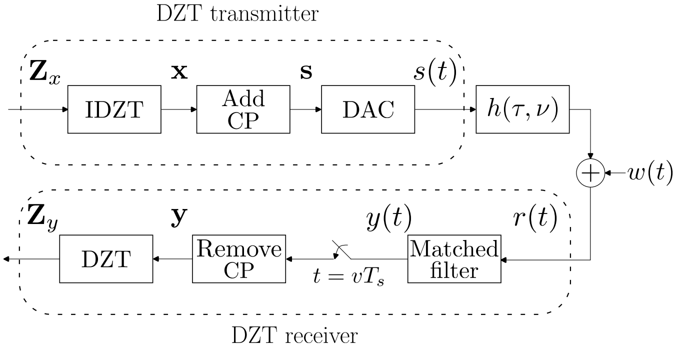


Fig. 2: Block diagram of the DZT-OTFS system.

where $g(t)$ is the transmit pulse and $T_s = 1/B$. The signal $s(t)$ is transmitted through a time-varying channel with impulse response $h(\tau,\nu) = \sum_{i=1}^{I} \alpha_i \delta(\tau - \tau_i)\delta(\nu - \nu_i)$, where I is the number of paths, and α_i, τ_i, and ν_i are the channel coefficient, delay, and Doppler of the ith path, respectively. The received signal, $r(t)$, at the receiver is

$$r(t) = \sum_{i=1}^{I} \alpha_i s(t-\tau_i) e^{j2\pi\nu_i t} + w(t), \quad (4)$$

where $w(t)$ is the additive noise. The received signal is passed through the matched filter, whose output is given by

$$y(t) = \int_{-\infty}^{\infty} r(\tau) g^*(\tau - t) d\tau. \quad (5)$$

Using (3) and (4) in (5), we get

$$y(t) = \sum_{i=1}^{I} \alpha_i \sum_{u=-L_{CP}}^{MN-1} \mathbf{s}[u] \int_{-\infty}^{\infty} g(\tau - uT_s - \tau_i) g^*(\tau - t) e^{j2\pi\nu_i\tau} d\tau + \tilde{w}(t), \quad (6)$$

where $\tilde{w}(t)$ is the match filtered noise. Assuming that the maximum Doppler, $\max_i\{\nu_i\}$, is much less than the bandwidth of the pulse, and denoting $f(t) = \int g(\tau) g^*(\tau - t) d\tau$, $y(t)$ can be approximated as [14]

$$y(t) \approx \sum_{i=1}^{I} \alpha_i e^{j2\pi\tau_i\nu_i} \sum_{u=-L_{CP}}^{MN-1} \mathbf{s}[u] e^{j2\pi\nu_i u T_s} f(t - uT_s - \tau_i) + \tilde{w}(t). \quad (7)$$

For the considered $g(t)$, $f(t)$ can be approximately bounded to finite duration in time [14]. The signal $y(t)$ is sampled at rate $1/T_s$ to obtain the discrete signal

$$\mathbf{y}[v] = \sum_{i=1}^{I} \alpha_i e^{j2\pi\tau_i\nu_i} \sum_{u=-L_{CP}}^{MN-1} \mathbf{s}[u] e^{j2\pi u \nu_i T_s} f_i[v-u] + \tilde{\mathbf{w}}[v], \quad (8)$$

where $f_i(u) = f(uT_s - \tau_i)$ is assumed to have a finite support satisfying the condition that the range of the support is much less than MN. Removing the CP, (8) can be approximated as

$$\mathbf{y}[v] \approx \sum_{i=1}^{I} \alpha_i e^{j2\pi\frac{l_i k_i}{MN}} \sum_{u=0}^{MN-1} \mathbf{s}[u] e^{j2\pi u \frac{k_i}{MN}} \tilde{f}_i[v-u] + \tilde{\mathbf{w}}[v], \quad (9)$$

where $\tilde{f}_i[u]$ is the periodic version of $f_i[u]$ with period MN, $k_i = \nu_i MNT_s \in \mathbb{R}$, and $l_i = \frac{\tau_i}{T_s} \in \mathbb{R}^+$. Equation (9) can be written in a vectorized form as

$$\mathbf{y} = \sum_{i=1}^{I} \alpha_i e^{j2\pi \frac{l_i k_i}{MN}} [(\mathbf{x} \cdot \mathbf{v}_i) \circledast \tilde{\mathbf{f}}_i] + \tilde{\mathbf{w}}, \tag{10}$$

where $\mathbf{v}_i[u] = e^{j2\pi u \frac{k_i}{MN}}$, $\mathbf{x} \cdot \mathbf{v}_i$ denotes the element-wise product of $\mathbf{x}$ and $\mathbf{v}_i$, and $\circledast$ is the circular convolution operator. The vector $\mathbf{y}$ is transformed to DD domain using DZT to obtain $\mathbf{Z}_y$ as [14]

$$\mathbf{Z}_y[m,n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \mathbf{y}[m + kM] e^{\frac{-j2\pi nk}{N}}. \tag{11}$$

Substituting $\mathbf{y}$ in (11) and using modulation and circular convolution properties of DZT [13], $\mathbf{Z}_y$ can be written as

$$\mathbf{Z}_y = \sum_{i=1}^{I} \alpha_i e^{j2\pi\tau_i\nu_i} \mathbf{Z}_{y_i} + \mathbf{w}, \tag{12}$$

where

$$\mathbf{Z}_{y_i}[m,n] = \sum_{l=0}^{M-1} \left(\sum_{k=0}^{N-1} \mathbf{Z}_x[l,k] \mathbf{Z}_{v_i}[l, n-k] \right) \mathbf{Z}_{\tilde{f}_i}[m-l, n], \tag{13}$$

and $\mathbf{Z}_{v_i}$ and $\mathbf{Z}_{\tilde{f}_i}$ are Zak transforms of $\mathbf{v}_i$ and $\tilde{\mathbf{f}}_i$, respectively.

A. Vectorization of input-output relation

Let $\mathbf{z}_y, \mathbf{z}_{y_i}, \mathbf{z}_x$ denote the vectorized forms of $\mathbf{Z}_y, \mathbf{Z}_{y_i}, \mathbf{Z}_x$, respectively, i.e., the $(nM + m)$th element in the vector is the $[m,n]$th entry in the corresponding matrix. The vectorized form of input-output relation between $\mathbf{z}_{y_i}$ and $\mathbf{z}_x$ is derived as follows.

Let $\mathbf{A} \in \mathbb{C}^{M \times N}$ and $\mathbf{B} \in \mathbb{C}^{2M-1 \times N}$ be two matrices with entries $\mathbf{A}[m,n] = \mathbf{Z}_{v_i}[m,n]$ and $\mathbf{B}[m,n] = \mathbf{Z}_{\tilde{f}_i}[m-(M-1), n]$, $m = 0, \cdots, M-1, \;\; n = 0, \cdots, N-1$. Also, let $\mathbf{R}_N \in \mathbb{C}^{N \times N}$ be a reversal matrix and $\mathbf{P}_N$ be a basic circulant permutation matrix of size N [20]. Define a matrix $\mathbf{H}_q^{(i)'} \in \mathbb{C}^{M \times N}$ as

$$\mathbf{H}_q^{(i)'}[m,n] = \begin{cases} \mathbf{A}[m,n], & \text{if } m = [q]_M \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

for $q = 0, 1, \cdots, MN-1$. Here, $[\cdot]_M$ denotes the modulo-M operation. Let $\mathbf{H}_1^{(i)} \in \mathbb{C}^{MN \times MN}$ be a matrix whose qth row is filled with vec$(\mathbf{H}_q^{(i)'} \mathbf{R}_N \mathbf{P}_N^{\lfloor \frac{q}{M} \rfloor + 1})$, where $\lfloor \cdot \rfloor$ denotes the floor operator. Define $\mathbf{H}_q^{(i)''} \in \mathbb{C}^{M \times N}$ as

$$\mathbf{H}_q^{(i)''}[m,n] = \begin{cases} \mathbf{B}[m + [q]_M, n], & \text{if } n = \lfloor \frac{q}{M} \rfloor \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Also, define $\mathbf{H}_2^{(i)} \in \mathbb{C}^{MN \times MN}$ whose qth row is filled with vec$(\mathbf{R}_M \mathbf{H}_q^{(i)''})$. Finally, (13) and (12) can be vectorized as

$$\mathbf{z}_{y_i} = \mathbf{H}_2^{(i)} \mathbf{H}_1^{(i)} \mathbf{z}_x \tag{16}$$

and

$$\mathbf{z}_y = \sum_{i=1}^{I} \alpha_i e^{j2\pi \frac{l_i k_i}{MN}} \mathbf{z}_{y_i}, \tag{17}$$

respectively. Here, the matrix $\mathbf{H}_1^{(i)}$ effectively carries out element-wise multiplication with $\mathbf{v}_i$ and $\mathbf{H}_2^{(i)}$ carries out the circular convolution with $\tilde{\mathbf{f}}_i$ in (10).

III. Proposed TF learning based DD channel estimation

In this section, we present the pilot frame architecture, describe the iterative channel estimation algorithm in [8] by adapting it for DZT-OTFS, and present the proposed TF learning approach that offers significant complexity reduction.

To estimate the DD domain channel at the receiver, a known pilot frame is transmitted. We consider a pilot frame consisting of a pilot symbol at the center and zeros elsewhere, i.e.,

$$\mathbf{Z}_x[m,n] = \begin{cases} \sqrt{MNE_p}, & \text{if } m = \frac{M}{2}, n = \frac{N}{2} \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

Channel estimation algorithm: Equation (17) can be written in an alternate form as

$$\mathbf{z}_y = \sum_{i=1}^{I} \mathbf{g}_i \alpha_i + \mathbf{w} = \mathbf{G}\boldsymbol{\alpha} + \mathbf{w}, \tag{19}$$

where $\mathbf{g}_i = e^{j2\pi \frac{l_i k_i}{MN}} \mathbf{H}_2^{(i)} \mathbf{H}_1^{(i)} \mathbf{z}_x \in \mathbb{C}^{MN \times 1}$, $\mathbf{G} = [\mathbf{g}_1(l_1, k_1), \mathbf{g}_2(l_2, k_2), \cdots, \mathbf{g}_I(l_I, k_I)] \in \mathbb{C}^{MN \times I}$, and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_I]^T \in \mathbb{C}^{I \times 1}$. The matrix $\mathbf{G}$ is referred to as the delay-Doppler matrix (DDM) as it captures the effect of the channel delay and Doppler on the transmitted symbols. The maximum likelihood (ML) solution for the three tuple estimation is then given by

$$(\hat{\mathbf{l}}, \hat{\mathbf{k}}, \hat{\boldsymbol{\alpha}}) = \underset{\mathbf{l}, \mathbf{k}, \boldsymbol{\alpha}}{\text{argmin}} \| \mathbf{z}_y - \mathbf{G}(\mathbf{l}, \mathbf{k}) \boldsymbol{\alpha} \|_2^2, \tag{20}$$

where $\| \cdot \|_2$ denotes 2-norm. This is an estimation problem in three variables. To reduce the complexity, we first solve for $\boldsymbol{\alpha}$ given $(\mathbf{l}, \mathbf{k})$ as

$$\boldsymbol{\alpha} = \left[\mathbf{G}^H(\mathbf{l}, \mathbf{k}) \mathbf{G}(\mathbf{l}, \mathbf{k}) \right]^{-1} \mathbf{G}^H(\mathbf{l}, \mathbf{k}) \mathbf{z}_y. \tag{21}$$

Now, to estimate $\mathbf{k}$ and $\mathbf{l}$, given $\boldsymbol{\alpha}$, (20) can be solved to obtain

$$\hat{\mathbf{l}}, \hat{\mathbf{k}} = \underset{\mathbf{l}, \mathbf{k}}{\text{arg max}} \left[\boldsymbol{\Theta}(\mathbf{G}) \right], \tag{22}$$

where $\mathbf{G} = \mathbf{z}_y^H \mathbf{G}(\mathbf{l}, \mathbf{k}) \left(\mathbf{G}^H(\mathbf{l}, \mathbf{k}) \mathbf{G}(\mathbf{l}, \mathbf{k}) \right)^{-1} \mathbf{G}^H(\mathbf{l}, \mathbf{k}) \mathbf{z}_y$. Substituting $\mathbf{l} = \hat{\mathbf{l}}$ and $\mathbf{k} = \hat{\mathbf{k}}$ in (21), we obtain the estimate of the channel coefficient vector $\hat{\boldsymbol{\alpha}}$.

The channel estimation algorithm proceeds in a path-wise fashion, i.e., the delay and Doppler values of pth path ($1 \leq p \leq P_{\text{max}}$) are estimated before the values of $(p+1)$th path values are estimated. Since the knowledge of the number of paths is not assumed to be known, a maximum of P_{max} paths are estimated. The estimates of l_p and k_p for the pth path is carried out in two steps. First, a coarse estimation (integer estimation) is carried out to obtain $\tilde{l}_p$, $\tilde{k}_p$. This is followed by an iterative fine estimation step where the fractional estimation of the delay and Doppler is carried out to obtain $\hat{l}_p$, $\hat{k}_p$. In each of the steps, the cost function in (22) is maximized over different search ranges as described below. The algorithm begins by initializing $\mathbf{G}(\mathbf{l}, \mathbf{k}) = [\mathbf{g}_1(l_1, k_1) \; \mathbf{g}_2(l_2, k_2) \; \cdots \; \mathbf{g}_{P_{\text{max}}}(l_{P_{\text{max}}}, k_{P_{\text{max}}})] = \mathbf{0}_{MN \times P_{\text{max}}}$.

Coarse estimation: The search range in this step is defined as $\mathcal{G} = \mathcal{L} \otimes \mathcal{K}$, where $\mathcal{L} = \{0, 1, \cdots, \lceil l_{\max} \rceil\}, \mathcal{K} = \{-\lceil k_{\max} \rceil, \cdots, 0, \cdots, \lceil k_{\max} \rceil\}$, $l_{\max} = \max\limits_i \{l_i\}$, $k_{\max} = \max\limits_i \{k_i\}$, and $\otimes$ denotes the Cartesian product of two sets. For estimating the parameters of the pth path, $\mathbf{g}_p(l_p, k_p)$ is computed for all (l_p, k_p) in $\mathcal{G}$ and the coarse estimates are obtained using (22) by maximizing the cost function over the search range.

Iterative fine estimation: Following the coarse estimation step, the search area is now defined around the optimal coarse value (for $s = 1$) or the fine estimate obtained in the previous iteration of the fine estimation step (for $s > 1$), given by

$$\mathcal{I}(s) = \left\{ \left\{ l_p^{(s-1)} - \frac{5}{10^s}, l_p^{(s-1)} - \frac{4}{10^s}, \cdots, l_p^{(s-1)} + \frac{5}{10^s} \right\} \otimes \left\{ k_p^{(s-1)} - \frac{5}{10^s}, k_p^{(s-1)} - \frac{4}{10^s}, \cdots, k_p^{(s-1)} + \frac{5}{10^s} \right\} \right\}, \quad (23)$$

with s denoting the iteration number in the fine estimation step. To begin the iterations, $s = 1$, $l_p^{(0)} = \tilde{l}_p$, and $k_p^{(0)} = \tilde{k}_p$. A similar procedure as in coarse estimation step is followed using $\mathcal{I}(s)$ as the search range for obtaining the first fine estimate $(l_p^{(1)}, k_p^{(1)})$, following which s is incremented by 1. Note that the search resolution becomes finer as s increases. Next, for $s > 1$, the search area is centered over the newly obtained fine estimate with finer resolution. This iterative procedure is stopped when a predefined value for s is achieved, i.e., $s = s_{\max}$, and $(\hat{l}_p, \hat{k}_p) = (l_p^{(s_{\max})}, k_p^{(s_{\max})})$.

Stopping criterion: The algorithm stops once $P_{\max}$ paths have been estimated or $\|\mathbf{z}_c^{(p)} - \mathbf{z}_c^{(p-1)}\|_2^2 < \epsilon$, where $\mathbf{z}_c^{(p)} = \mathbf{G}(\hat{\mathbf{l}}, \hat{\mathbf{k}})\hat{\boldsymbol{\alpha}}(\hat{\mathbf{l}}, \hat{\mathbf{k}})$.

A. Proposed TF based learning approach using DNN

In the channel estimation algorithm described above, the coarse estimation step and the iterative fine estimation step require multiple estimations using cost function in (22), which requires the computation of the DDM, $\mathbf{G}$. Computing the columns of $\mathbf{G}$, $\mathbf{g}_i(l_i, k_i)$, for each path involves high complexity. Therefore, in order to reduce the complexity, we propose to design and train a network to learn the columns of $\mathbf{G}$. Specifically, we note that the DDM is a function of delay and Doppler (i.e., each column of DDM, $\mathbf{g}(l, k)$, has a one-to-one relation to (l, k)-tuple), and we use DNNs to learn this one-to-one relation. It turns out that the proposed learning/training architecture is able to effectively learn this relation accurately, offering complexity benefit in the process. The architecture and training methodology are presented in the following sections.

1) Architecture: Figure 3 shows the block diagram of the proposed TF learning approach. The architecture consists of two architecturally identical neural networks, DNN1 and DNN2, which receive the delay and Doppler indices $(\breve{\mathbf{l}}, \breve{\mathbf{k}})$ as input. The input is a matrix of size $S \times 2$, where S is the cardinality of $\mathcal{G}$ for coarse estimation step or the cardinality of $\mathcal{I}(s)$ for the sth iteration of the fine estimation step and $\breve{\mathbf{l}} = \zeta \mathbf{l}/l_{\max}, \breve{\mathbf{k}} = \zeta \mathbf{k}/k_{\max}$. The division by $l_{\max}$ ($k_{\max}$) is carried out to normalize the values of delay (Doppler) indices between 0 and 1 (-1 and 1)[1]. Further, the multiplication by ζ is carried out to magnify small changes in the delay and Doppler indices in the training and test data. The vectors $\mathbf{l}$ and $\mathbf{k}$ are obtained from the search area $\mathcal{G}$ or $\mathcal{I}(s)$. The input matrix is passed through DNN1 and DNN2. DNN1 (DNN2) is trained to output the real (imaginary) part of the column, $\Re\{\mathbf{G}_{\text{col}}^{\text{TF}}\} \in \mathbb{R}^{S \times MN}$ ($\Im\{\mathbf{G}_{\text{col}}^{\text{TF}}\} \in \mathbb{R}^{S \times MN}$), of the DDM in TF domain. The real and imaginary parts are combined and reshaped to obtain $\mathbf{G}_{\text{col}}^{\text{TF}} \in \mathbb{R}^{S \times M \times N}$. Each $M \times N$ matrix in $\mathbf{G}_{\text{col}}^{\text{TF}}$ is then converted to DD domain from TF domain using SFFT and vectorized column-wise to obtain an MN-length vector. These vectors form the rows of $\mathbf{G}_{\text{col}} \in \mathbb{C}^{S \times MN}$. The DNN1 and DNN2 are trained so as to provide $\mathbf{g}(\mathbf{l}[t], \mathbf{k}[t]) \in \mathbb{C}^{1 \times MN}$ as the tth row of $\mathbf{G}_{\text{col}}$ as output for $(\breve{\mathbf{l}}[t], \breve{\mathbf{k}}[t]) \in \mathbb{R}^{1 \times 2}$ as the tth row in the input matrix.

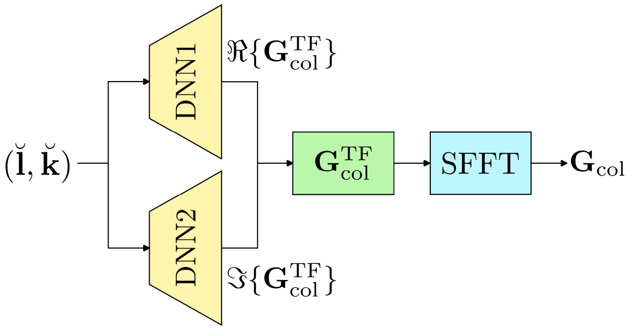

$(\breve{\mathbf{l}}, \breve{\mathbf{k}})$
DNN1
DNN2
$\Re\{\mathbf{G}_{\text{col}}^{\text{TF}}\}$
$\Im\{\mathbf{G}_{\text{col}}^{\text{TF}}\}$
$\mathbf{G}_{\text{col}}^{\text{TF}}$
SFFT
$\mathbf{G}_{\text{col}}$

Fig. 3: Proposed TF learning architecture for learning $\mathbf{G}$.

Architectures of DNN1 and DNN2 are comprised of fully connected layers. For each layer, the output dimension is twice the input dimension, i.e., the ith layer ($i = 1, 2, \cdots$) of DNN1 and DNN2 has input and output dimensions 2^i and 2^{i+1}, respectively. Number of layers, N_L, in DNN1 and DNN2 are determined by the choices of M and N, such that the last layer has input dimension 2^{N_L} and output dimension $\min(2^{N_L+1}, MN)$, with $2^{N_L} < MN$ and $2^{N_L+1} \geq MN$. For each fully connected layer except the last layer, a rectified linear unit (ReLU) activation function is used and a linear activation function is used for the last layer to allow the output of DNN1 and DNN2 to span $\mathbb{R}$.

2) Training methodology: Training data is obtained by generating (l, k) tuples and the corresponding $\mathbf{g}(l, k)$ vectors using $\mathbf{g}(l, k) = e^{j2\pi \frac{lk}{MN}} \mathbf{H}_2 \mathbf{H}_1 \mathbf{z}_x$ (see (19)). The vectors $\mathbf{g}(l, k) \in \mathbb{C}^{MN \times 1}$ are reshaped into matrices of size $M \times N$ and converted to TF domain using ISFFT, following which they are vectorized to obtain $\mathbf{g}^{\text{TF}}(l, k) \in \mathbb{C}^{1 \times MN}$. To train the network, the tuples (l, k) are fed as input to the DNN1 and DNN2 to generate the output. Training is carried out using an Adam optimizer to minimize the mean square error loss evaluated between the output of the DNN1 (DNN2) and $\Re\{\mathbf{g}^{\text{TF}}(l, k)\}$ ($\Im\{\mathbf{g}^{\text{TF}}(l, k)\}$). The other hyper parameters used while training are presented in Table I. We note that this

[1] The normalization of values between 0 and 1, and -1 and 1 for delay and Doppler indices, respectively, are done so that ranges are similar, which aids training. Without this normalization, delay indices would span 0 to $l_{\max}$ and Doppler indices would span $-k_{\max}$ and $k_{\max}$. Note that $l_{\max}$ and $k_{\max}$ need not be equal.

Hyper parameter	Value
Batch size	40000
Mini batch size	8000
Number of epochs	40000
Learning rate	0.001, multiply by 0.9 every 4000 epochs
Number of training samples	325000

TABLE I: Hyper parameters used while training.

training has to be carried out *offline, only once*. Subsequently, the network weights are stored. During test time, the same trained weights are used for both coarse and fine estimation steps of the channel estimation algorithm.

IV. Results and Discussions

This section presents the performance of the proposed TF learning based channel estimation algorithm. A DZT-OTFS system with $M = 64, N = 32$ is considered. Square-root raised cosine pulse with roll-off factor 0.5 is used as the transmit and receive pulse. Two parameter sets are considered for the simulation: For the first set, $\Delta f = 3.75$ kHz, $I = 4$ with uniform power delay profile (PDP), delays are uniformly distributed in $(0, \tau_{\max}]$, $\tau_{\max} = 0.133$ ms, and $\nu_{\max} = 937$ Hz. The second set, a more practical scenario, considers Vehicular A (VehA) channel model [21] with $\Delta f = 156.25$ kHz, and $\nu_{\max} = 1700$ Hz. For both the cases, Dopplers are generated using Jakes' Doppler spectrum, $\nu_i = \nu_{\max}\cos(\theta_i)$, where θ_i is uniformly distributed in $(0, 2\pi]$, and carrier frequency $f_c = 4$ GHz. Further, the following algorithm parameters are chosen: $P_{\max} = 15$, $n_{\max} = 2$, and $\epsilon = 20\sigma^2$, where σ^2 is the variance of noise. For the networks DNN1 and DNN2, $N_L = 10$ and $\zeta = 10^3$. A single training is carried out and the same trained network is used during the testing phase in both the scenarios which shows the network's generalizability. Pilot signal-to-noise ratio (SNR) is taken to be the same as data SNR. Normalized mean square error (NMSE) is computed as $\frac{\|\hat{\mathbf{H}}-\mathbf{H}\|_F^2}{\|\mathbf{H}\|_F^2}$, where $\hat{\mathbf{H}}$ is the channel matrix obtained using the estimated $(\hat{\boldsymbol{\alpha}}, \hat{\mathbf{l}}, \hat{\mathbf{k}})$ and $\|\cdot\|_F$ denotes the Frobenius norm.

DD domain vs TF domain training: The NMSE and BER performances of the channel estimation algorithm using learning in DD domain and TF domain are presented in Figs. 4 and 5, respectively. It is seen that with DD domain learning, both NMSE and BER performances are poor. This is because the large swing in the absolute values of the $\mathbf{G}$ matrix entries in the DD domain (see Fig. 1a) results in ineffective training. Since the same network needs to cater to both the high and low values, the training accuracy and thereby the NMSE and BER performances is poor when trained in the DD domain. Whereas, the performances are seen to significantly improve with the proposed TF domain learning, which is a consequence of the effective training achieved in the TF domain due to a smaller swing in the absolute values in the TF domain (see Fig. 1b). Further, it is also seen that the learning in the TF domain achieves close to perfect channel state information (CSI) performance.

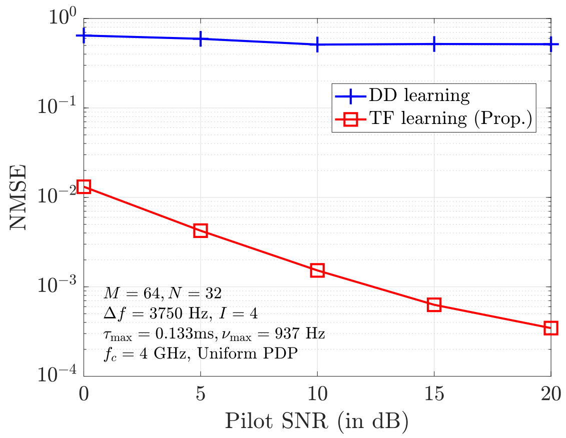


Fig. 4: NMSE performance comparison between DD domain learning and TF domain learning.

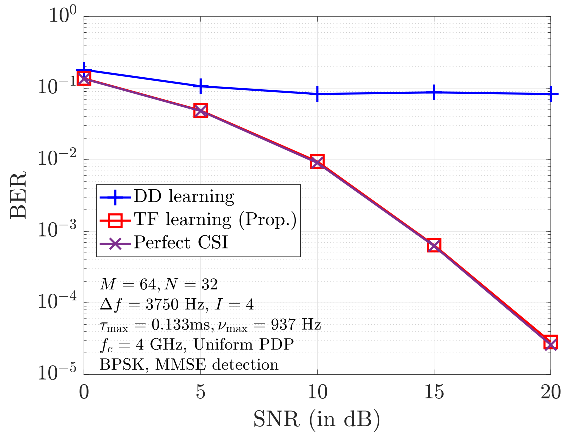


Fig. 5: BER performance comparison between DD domain learning and TF domain learning.

NMSE and BER performance: The NMSE performance of the proposed TF learning based channel estimation algorithm is plotted as a function of pilot SNR in Fig. 6. The NMSE performance obtained for uniform PDP using the estimation algorithm in [8] and a modified maximum likelihood estimation (M-MLE) algorithm in [4] is also added for comparison. It is seen that for uniform PDP, the performance of the proposed approach is quite close[2] to that in [8]. Further, the performance of M-MLE is observed to be worse than the proposed approach. The NMSE performance of the proposed approach with VehA PDP is also seen to perform closely to that with uniform PDP. It is noted that the same trained network works effectively for uniform PDP and VehA PDP channel models, highlighting its generalizability. Next, Fig. 7 shows the BER performance of the algorithm in [8], M-MLE algorithm in [4], and the proposed approach as a function of SNR. The performance attained using perfect CSI is also plotted for comparison. It is seen that BER performance close to that with perfect CSI is achieved by the proposed method

[2] The performance of the proposed approach is slightly inferior compared to that in [8] in Fig. 6. This is because [8] uses exact computation of the DDM, whereas the proposed approach learns the DDM, which is a close approximation of the exact DDM. This difference between the exact and learnt DDM translates to a slight performance degradation.

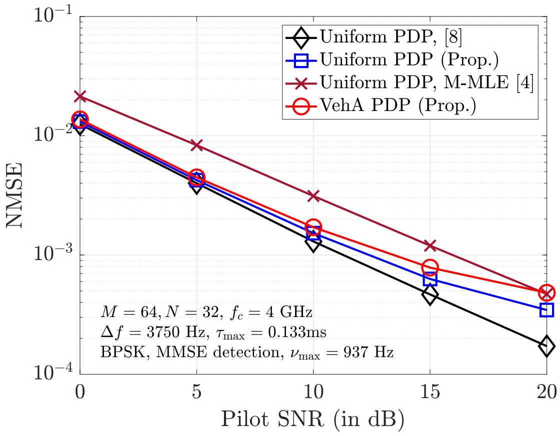


Fig. 6: NMSE performance of the algorithm in [8], M-MLE [4], and the proposed algorithm for different PDPs.

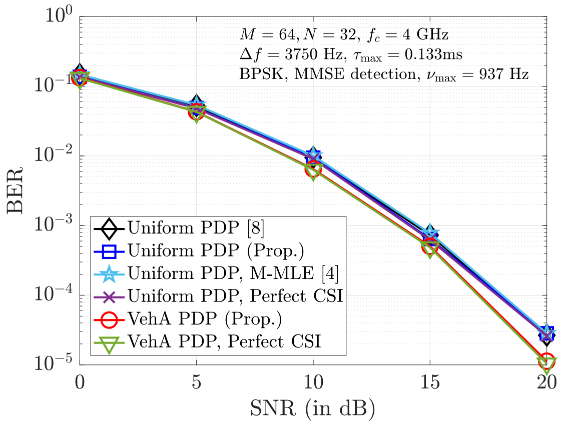


Fig. 7: BER performance of the algorithm in [8], M-MLE [4], and the proposed algorithm for different PDPs.

at a much reduced complexity, which is detailed below.

Complexity: The run time complexities required for the generation of $\mathbf{g}(l,k)$ using 1) the proposed TF learning approach and 2) the exact analytical computation, i.e., computing $\mathbf{g}(l,k) = e^{j2\pi\frac{lk}{MN}}\mathbf{H}_2\mathbf{H}_1\mathbf{z}_x$, are presented here. We obtained the run time complexities using both the approaches on the same machine for fair comparison. The proposed approach takes about 0.04 seconds (including the conversion to DD domain) to generate $\mathbf{g}(l,k)$, while the exact analytical computation takes about 0.4 seconds. This is a significant reduction in complexity. We note that while the analytical computation gives exact values, the proposed method gives the values through learning which need not be exact. Yet, the performance achieved by the proposed learning is quite close to those obtained using the exact analytical computation. The one order complexity reduction achieved by the proposed learning without compromising much on performance is substantial and is quite attractive for practical implementation.

V. Conclusions

We proposed a novel TF learning based channel estimation algorithm for DZT-OTFS systems with fractional DDs. The training of the network was carried out in the TF domain instead of DD domain, which yielded better training accuracy and learning performance. The trained network was shown to generalize well across different channel models. Numerical results demonstrated that the proposed TF learning approach achieved good NMSE and BER performance while being computationally efficient. The proposed learning approach for embedded pilot frames and general pulse shapes can be carried out as future work.

References

[1] R. Hadani et al., “Orthogonal time frequency space modulation,” *Proc. IEEE WCNC’2017*, pp. 1-6, Mar. 2017.

[2] Y. Hong, T. Thaj, and E. Viterbo, *Delay-Doppler Communications: Principles and Applications*, London UK: Elsevier, 2022.

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

[4] I. A. Khan and S. K. Mohammed, “Low complexity channel estimation for OTFS modulation with fractional delay and Doppler,” arXiv:2111.06009 [cs.IT], Nov. 2021.

[5] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, “Interference cancellation and iterative detection for orthogonal time frequency space modulation,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6501-6515, Oct. 2018.

[6] M. K. Ramachandran and A. Chockalingam, “MIMO-OTFS in high-Doppler fading channels: signal detection and channel estimation,” *Proc. IEEE GLOBECOM’2018*, pp. 206-212, Dec. 2018.

[7] 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.

[8] S. P. Muppaneni, S. R. Mattu, and A. Chockalingam, “Channel and radar parameter estimation with fractional delay-Doppler using OTFS,” *IEEE Commun. Letters*, vol. 27, no. 5, pp. 1392-1396, May 2023.

[9] A. J. E. M. Janssen, “The Zak transform: a signal transform for sampled time-continuous signals,” Philips J. Res., 43, pp. 23-69, 1988.

[10] S. K. Mohammed, “Derivation of OTFS modulation from first principles,” *IEEE Trans. Veh. Tech.*, vol. 70, no. 8, pp. 7619-7636, Aug. 2021.

[11] S. K. Mohammed, R. Hadani, A. Chockalingam and R. Calderbank, “OTFS - a mathematical foundation for communication and radar sensing in the delay-Doppler domain,” *IEEE BITS the Information Theory Magazine*, vol. 2, no. 2, pp. 36-55, Nov. 2022.

[12] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, “OTFS - predictability in the delay-Doppler domain and its value to communication and radar sensing,” *IEEE BITS the Information Theory Magazine*, IEEE Xplore Early Access doi: 10.1109/MBITS.2023.3319595.

[13] H. Bolcskei and F. Hlawatsch, “Discrete Zak transforms, polyphase transforms, and applications,” *IEEE Trans. Signal Processing*, vol. 45, no. 4, pp. 851-866, Apr. 1997.

[14] F. Lampel, A. Avarado, F. M. J. Willems, “On OTFS using the discrete Zak transform,” *IEEE ICC’2022 Workshops*, pp. 729-734, May 2022.

[15] V. Yogesh, V. Bhat, S. R. Mattu, and A. Chockalingam, “On the bit error performance of OTFS modulation using discrete Zak transform,” *Proc. IEEE ICC’2023*, May-Jun. 2023.

[16] T. Thaj, E. Viterbo, and Y. Hong, “General I/O relations and low-complexity universal MRC detection for all OTFS variants,” *IEEE Access*, vol. 10, pp. 96026-96037, 2022.

[17] N. Van Huynh and G. Y. Li, “Transfer learning for signal detection in wireless networks,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 11, pp. 2325-2329, Nov. 2022.

[18] H. Park, J. Kang, S. Lee, J. W. Choi, and S. Kim, “Deep Q-network based beam tracking for mobile millimeter-wave communications,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 961-971, Feb. 2023.

[19] I. Helmy, P. Tarafder, and W. Choi, “LSTM-GRU model-based channel prediction for one-bit massive MIMO system,” *IEEE Trans. Veh. Tech.*, vol. 72, no. 8, pp. 11053-11057, Aug. 2023.

[20] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge Univ. Press, 2013.

[21] ITU-R M.1225, “Guidelines for the evaluation of radio transmission technologies for IMT-2000,” International Telecommunication Union Radio communication, 1997.