DNN-Based Joint I/O Relation Estimation and Detection in Zak-OTFS

Naveed Bin Nazir and Ananthanarayanan Chockalingam
Department of ECE, Indian Institute of Science, Bangalore

***Abstract*—Zak transform-based orthogonal time frequency space (Zak-OTFS), a delay-Doppler (DD) domain modulation, is well-suited for general wireless channels, including channels which are highly doubly-dispersive in nature. Accurate input-output (I/O) relation estimation and signal detection at the receiver are crucial for reliable communication. In this paper, we propose a deep neural network (DNN) based Zak-OTFS receiver that jointly performs I/O relation estimation and signal detection, considering an exclusive pilot frame. We focus on two commonly considered DD pulse shaping filters, namely, sinc and Gaussian filters, to demonstrate the effectiveness of the DNN approach. A fully-connected DNN is trained to jointly learn the DD channel within a spatial coherence period and optimize the detection performance. Our simulation results demonstrate that the proposed DNN approach outperforms the conventional approach where model-free I/O relation estimation and minimum mean square error detection are performed separately.**

***Index Terms*—Zak-OTFS modulation, delay-Doppler domain, I/O relation estimation, signal detection, deep neural network, deep learning.**

I. Introduction

Next generation wireless communication systems are envisioned to witness highly time-varying channels where the Doppler spreads are in the kHz range, due to increased mobile speeds and use of high carrier frequencies. Orthogonal time frequency space (OTFS) modulation [1] is a promising modulation scheme suitable for doubly-selective channels. OTFS modulation multiplexes information symbols in the delay-Doppler (DD) domain, followed by conversion from DD domain to time domain for transmission. The received time-domain signal is converted back to DD domain where symbol detection is carried out. Transformation between domains can be performed in different ways. A well known way is the one used in multicarrier OTFS (MC-OTFS) [1]-[4], where the transformation from DD domain to time domain is carried out in two steps, viz., inverse symplectic finite Fourier transform for DD domain to time-frequency (TF) domain conversion, followed by Heisenberg transform for TF-domain to time domain conversion. Corresponding inverse transforms are carried out at the receiver. An alternate way is to achieve direct transformation from DD domain to time domain in a single step using Zak-transform approach [5],[6],[7].

The Zak transform-based OTFS (Zak-OTFS) approach has the advantage of achieving robust performance over a larger range of delay and Doppler spreads compared to the MC-OTFS approach [6]-[8]. The basic Zak-OTFS waveform is a quasi-periodic DD domain pulse, on which an information symbol is multiplexed. In order to limit the transmit signal to finite bandwidth and time duration, a DD domain pulse shaping filter is used at the transmitter. Commonly considered DD filters in the Zak-OTFS literature include sinc, root raised cosine, and Gaussian filters [7]-[14]. The filtered DD domain signal is converted to time domain using inverse Zak transform for transmission. Input-output (I/O) relation estimation and signal detection are crucial tasks at the Zak-OTFS receiver. In the Zak-OTFS literature, these tasks are carried out separately. For example, the signal detection task has been carried out using minimum mean square error (MMSE) detection [7] and local neighborhood search based detection [14]. Likewise, the I/O relation estimation task has been carried out using model-dependent and model-free approaches [7]. In this paper, we jointly carry out the I/O relation estimation and detection tasks in Zak-OTFS using a deep learning framework.

Over the past decade, deep learning has significantly transformed various fields, enabling the resolution of complex, non-trivial problems that were once considered challenging. Applying deep learning techniques to wireless communication problems is on the rise, with advancements in channel estimation, signal detection, and compensation of impairments making wireless transceiver designs more efficient [15]-[22]. In this paper, we present a Zak-OTFS receiver using a deep learning framework, where a fully-connected deep neural network (DNN) is trained for joint I/O relation estimation and signal detection, considering an exclusive pilot frame. Unlike conventional methods that handle these tasks separately, our DNN approach performs these tasks jointly. We focus on two commonly considered DD pulse shaping filters, namely, sinc and Gaussian filters, to demonstrate the effectiveness of the DNN approach. A fully-connected DNN is designed and trained to jointly learn the DD channel within a spatial coherence interval and optimize the detection performance. Our simulation results demonstrate that the proposed DNN approach outperforms the conventional approach where model-free I/O relation estimation and MMSE detection are performed separately.

II. Zak-OTFS system model

The basic information carrier in Zak-OTFS is a quasi-periodic DD domain pulse localized in a fundamental DD period, defined by a delay period τ_{p} and a Doppler period

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

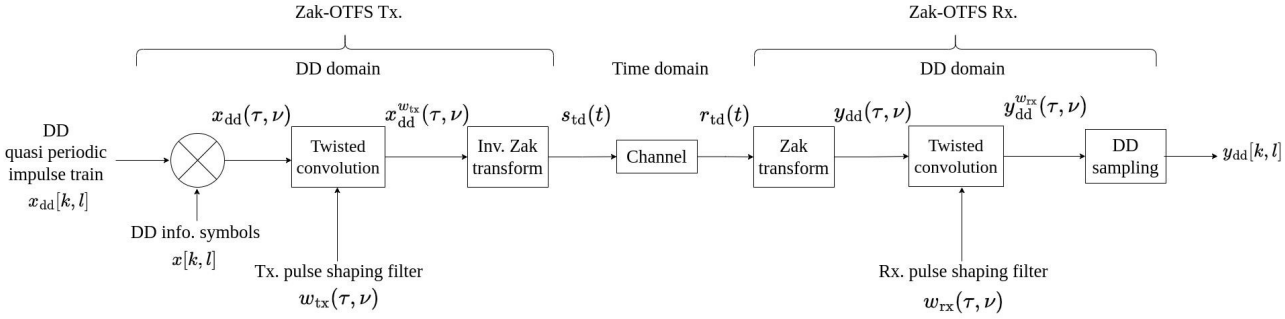

Fig. 1: Zak-OTFS transceiver signal flow diagram.

$\nu_{\rm p}$ such that $\tau_{\rm p}\nu_{\rm p} = 1$. The fundamental DD period is defined as $\mathcal{D}_0 = \{(\tau,\nu) \mid 0 \le \tau < \tau_p, 0 \le \nu < \nu_p\}$, where τ and ν represent the delay and Doppler variables, respectively, and $\tau_{\rm p}\nu_{\rm p} = 1$. A quasi-periodic DD domain pulse, when viewed in the time domain, is a pulsone which is a time domain pulse train modulated by a frequency tone. The delay period $\tau_{\rm p}$ is divided into M delay bins and the Doppler period $\nu_{\rm p}$ is divided into N Doppler bins, and MN information symbols are multiplexed on MN DD pulses located at these MN DD bins. M and N are chosen such that $MN = BT$, where B and T are the bandwidth of transmission and time duration of a Zak-OTFS frame. That is, the resolution along the delay axis is $\Delta\tau = \frac{1}{B} = \frac{\tau_{\rm p}}{M}$ and the resolution along the Doppler axis is $\Delta\nu = \frac{1}{T} = \frac{\nu_{\rm p}}{N}$. In order to limit the bandwidth and time duration to B and T, respectively, a DD domain pulse shaping filter is used at the transmitter. Fig. 1 depicts the block diagram of the Zak-OTFS transceiver.

The information symbols $x[k,l]$s, $k = 0,\cdots,M-1$, $l = 0,\cdots,N-1$ are drawn from a modulation alphabet $\mathbb{A}$. The DD grid on which these MN information symbols are multiplexed is given by $\Lambda_{\rm dd} \triangleq \left\{\left(k\frac{\tau_p}{M}, l\frac{\nu_p}{N}\right) \mid k = 0,\cdots,M-1, l = 0,\cdots,N-1\right\}$. The $x[k,l]$s on $\Lambda_{\rm dd}$ are encoded as quasi-periodic discrete DD information as $x_{\rm dd}[k+nM, l+mN] = x[k,l]e^{j2\pi n\frac{l}{N}}, n,m \in \mathbb{Z}$, which is converted into a continuous DD signal by mounting on a continuous DD domain quasi-periodic impulse train, resulting in $x_{\rm dd}(\tau,\nu) = \sum_{k,l\in\mathbb{Z}} x_{\rm dd}[k,l]\delta(\tau - k\Delta\tau)\,\delta(\nu - l\Delta\nu)$, where $\delta(\cdot)$ denotes Kronecker delta function. Note that $x_{\rm dd}(\tau,\nu)$ is also quasi-periodic with period τ_p and ν_p along the delay and Doppler axes, respectively, i.e.,

$$x_{\rm dd}(\tau+n\tau_p, \nu+m\nu_p) = e^{j2\pi n\nu\tau_p}\, x_{\rm dd}(\tau,\nu),\ \forall\, n,m \in \mathbb{Z}, \quad (1)$$

The signal $x_{\rm dd}(\tau,\nu)$ is then time and bandwidth limited by filtering through the Tx DD domain filter $w_{\rm tx}(\tau,\nu)$, to obtain

$$x_{\rm dd}^{\rm w_{tx}}(\tau,\nu) = w_{\rm tx}(\tau,\nu) *_\sigma x_{\rm dd}(\tau,\nu), \quad (2)$$

where $*_\sigma$ denotes twisted convolution[1]. The time domain signal for transmission is obtained using inverse Zak transform as

$$s_{\rm td}(t) = \mathcal{Z}_t^{-1}\left(x_{\rm dd}^{w_{\rm tx}}(\tau,\nu)\right) = \sqrt{\tau_p}\int_0^{\nu_p} x_{\rm dd}^{w_{\rm tx}}(t,\nu)d\nu. \quad (3)$$

The transmitted signal passes through the channel having P paths whose impulse response in the DD domain is given by $h(\tau,\nu) = \sum_{i=1}^{P} h_i\delta(\tau-\tau_i)\delta(\nu-\nu_i)$, where h_i, τ_i, and ν_i are the channel gain, delay, and Doppler of the ith path, respectively. The received time domain signal is given by

$$r_{\rm td}(t) = \int\int h(\tau,\nu)s_{\rm td}(t-\tau)e^{j2\pi\nu(t-\tau)}d\tau d\nu + n(t), \quad (4)$$

where $n(t)$ is the additive white Gaussian noise (AWGN). Zak transform is used to convert the received time domain signal to DD domain as $y_{\rm dd}(\tau,\nu) = \mathcal{Z}_t(r_{\rm td}(t))$, i.e.,

$$y_{\rm dd}(\tau,\nu) = \sqrt{\tau_p}\sum_{k\in\mathbb{Z}} r_{\rm td}(\tau+k\tau_p)e^{-j2\pi\nu k\tau_p} + n_{\rm dd}(\tau,\nu), \quad (5)$$

where $n_{\rm dd}(\tau,\nu) = \mathcal{Z}_t(n(t))$ is the noise in DD domain. Next, $y_{\rm dd}(\tau,\nu)$ is filtered through the Rx DD domain filter $w_{\rm rx}(\tau,\nu)$, which is matched to the Tx DD filter, i.e., $w_{\rm rx}(\tau,\nu) = w_{\rm tx}^*(-\tau,-\nu)e^{j2\pi\tau\nu}$. The output of the Rx DD filter is given by $y_{\rm dd}^{w_{\rm rx}}(\tau,\nu) = w_{\rm rx}(\tau,\nu) *_\sigma y_{\rm dd}(\tau,\nu)$, i.e.,

$$y_{\rm dd}^{w_{\rm rx}}(\tau,\nu) = h_{\rm eff}(\tau,\nu) *_\sigma x_{\rm dd}(\tau,\nu) + n_{\rm dd}^{w_{\rm rx}}(\tau,\nu), \quad (6)$$

where $h_{\rm eff}(\tau,\nu)$ is the effective continuous DD channel (consisting of the cascade of the Tx DD filter, physical DD channel, and Rx DD filter), given by

$$h_{\rm eff}(\tau,\nu) = w_{\rm rx}(\tau,\nu) *_\sigma h(\tau,\nu) *_\sigma w_{\rm tx}(\tau,\nu), \quad (7)$$

and $n_{\rm dd}^{w_{\rm rx}}(\tau,\nu) = w_{\rm rx}(\tau,\nu) *_\sigma n_{\rm dd}(\tau,\nu)$ is the filtered AWGN in DD domain. The DD domain signal $y_{\rm dd}^{w_{\rm rx}}(\tau,\nu)$ is sampled on the information lattice, resulting in the discrete quasi-periodic DD domain received signal $y_{\rm dd}[k,l]$ as

$$y_{\rm dd}[k,l] = y_{\rm dd}^{w_{\rm rx}}\Big(\tau = \frac{k\tau_{\rm p}}{M}, \nu = \frac{l\nu_{\rm p}}{N}\Big), \quad k,l \in \mathbb{Z}, \quad (8)$$

which is given by

$$y_{\rm dd}[k,l] = h_{\rm eff}[k,l] *_{\sigma{\rm d}} x_{\rm dd}[k,l] + n_{\rm dd}[k,l], \quad (9)$$

where the $*_{\sigma{\rm d}}$ in (9) is twisted convolution in discrete DD domain, i.e.,

$$h_{\rm eff}[k,l] *_{\sigma{\rm d}} x_{\rm dd}[k,l] = \sum_{k',l'\in\mathbb{Z}} h_{\rm eff}[k-k', l-l']x_{\rm dd}[k',l']\, e^{j2\pi\frac{k'(l-l')}{MN}}, \quad (10)$$

[1]Twisted convolution operation between two DD functions $a(\tau,\nu)$ and $b(\tau,\nu)$ is defined as $a(\tau,\nu) *_\sigma b(\tau,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} a(\tau',\nu')b(\tau-\tau', \nu-\nu')e^{j2\pi\nu'(\tau-\tau')}d\tau' d\nu'$. Twisted convolution operation preserves quasi-periodicity.

where the effective channel filter $h_{\text{eff}}[k,l]$ and filtered noise samples $n_{\text{dd}}[k,l]$ are given by

$$h_{\text{eff}}[k,l] = h_{\text{eff}}\Big(\tau = \frac{k\tau_p}{M}, \nu = \frac{l\nu_p}{N}\Big), \tag{11}$$

$$n_{\text{dd}}[k,l] = n_{dd}^{w_{\text{rx}}}\Big(\tau = \frac{k\tau_p}{M}, \nu = \frac{l\nu_p}{N}\Big). \tag{12}$$

Due to the quasi-periodicity in the DD domain, it is sufficient to consider the received samples $y_{\text{dd}}[k,l]$ within $\mathcal{D}_0$. We write the $y_{\text{dd}}[k,l]$ samples as a vector and the end-to-end DD domain I/O relation in matrix-vector form as

$$\mathbf{y} = \mathbf{H}_{\text{eff}}\mathbf{x} + \mathbf{n}, \tag{13}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{n} \in \mathbb{C}^{MN\times 1}$, such that their $(kN+l+1)^{\text{th}}$ entries are given by $x_{kN+l+1} = x_{\text{dd}}[k,l]$, $y_{kN+l+1} = y_{\text{dd}}[k,l]$, $n_{kN+l+1} = n_{\text{dd}}[k,l]$, and $\mathbf{H}_{\text{eff}} \in \mathbb{C}^{MN\times MN}$ is the effective channel matrix such that

$$\mathbf{H}_{\text{eff}}[k'N+l'+1, kN+l+1] = \sum_{m,n\in\mathbb{Z}} h_{\text{eff}}[k'-k-nM, l'-l-mN] e^{j2\pi nl/N} e^{j2\pi \frac{(l'-l-mN)(k+nM)}{MN}}, \tag{14}$$

where $k', k = 0, \ldots, M-1$, $l', l = 0, \ldots, N-1$.

III. Proposed DNN receiver architecture

The system model in (13) can be used for the purpose of I/O relation estimation and signal detection at the receiver. I/O relation estimation is the task of estimating the effective channel matrix $\mathbf{H}_{\text{eff}}$. Note that the $\mathbf{H}_{\text{eff}}$ matrix depends on the delay and Doppler spreads due to the Tx filter, the physical channel, and the Rx filter. Signal detection is the task of recovering the information symbols in the $\mathbf{x}$ vector, given the knowledge of the estimated $\mathbf{H}_{\text{eff}}$ matrix. Conventional approaches in the Zak-OTFS literature carry out these two tasks separately. For example, I/O relation estimation using model-dependent approach or model-free approach and signal detection using MMSE detection have been considered [7]. In the model-dependent approach of I/O relation estimation, the parameters of the physical channel $h(\tau,\nu)$, i.e., $\{\tau_i, \nu_i, h_i\}$s, are estimated using a channel estimation scheme and these estimated parameters are then used to construct the $\mathbf{H}_{\text{eff}}$ matrix. That is, use the estimated $\{\tau_i, \nu_i, h_i\}$s to compute $h_{\text{eff}}(\tau,\nu)$ defined in (7) and sample it to obtain $h_{\text{eff}}[k,l]$ as in (11), which when substituted in (14) gives the estimated I/O relation $\hat{\mathbf{H}}_{\text{eff}}$. On the other hand, model-free approach of I/O relation estimation does not require explicit estimation of the physical channel parameters $\{\tau_i, \nu_i, h_i\}$s. Instead, the I/O relation can be obtained by sending a pilot symbol in a frame and directly reading out the corresponding DD domain output samples in $\mathcal{D}_0$ at the receiver [7].

In this paper, instead of carrying out the I/O relation estimation and signal detection tasks separately, we perform these two tasks jointly. The approach we adopt for this purpose is deep learning approach. Specifically, we propose a DNN architecture in which we train the network using synthetically generated pairs of pilot and data frames as training data, and directly minimize the root mean square error (RMSE) between the actual and estimated data symbols, i.e., RMSE between actual and estimated data symbols is used as the loss function. So, the network learns the I/O relation internally and optimizes the detection performance without an explicit estimation of the I/O relation. The proposed DNN architecture and training methodology are described below.

Fig. 2 shows the proposed DNN-based Zak-OTFS receiver architecture for BPSK modulation. For higher-order modulation, two such DNNs are used (one for real part and another for imaginary part of the complex symbol). The proposed network is a fully-connected DNN that carries out joint estimation of the I/O relation and detection of the transmitted symbols. The number of input neurons in the DNN is $4MN$ and the number of output neurons is MN. For training the DNN, we generate two consecutive Zak-OTFS frames:

- $\mathbf{x}_{\text{P}}$: a pilot frame which contains a pilot symbol.
- $\mathbf{x}_{\text{D}}$: a data frame which carries the information symbols.

The DD domain channel is assumed to remain constant over these frames. The pilot frame consists of a single pilot symbol placed at the center of the pilot frame, i.e., $(k_{\text{p}}, l_{\text{p}}) = \big(\frac{M}{2}, \frac{N}{2}\big)$, and zeros at other locations. The data frame consists of information symbols. From (13), the received signal vector for the pilot frame is given by

$$\mathbf{y}_{\text{P}} = \mathbf{H}_{\text{eff}}\mathbf{x}_{\text{P}} + \mathbf{n}_{\text{P}}, \tag{15}$$

where $\mathbf{y}_{\text{P}} \in \mathbb{C}^{MN\times 1}$ and $\mathbf{x}_{\text{P}}$ is the vectorized representation of the pilot frame. The $(k_pN + l_p + 1)$th entry of $\mathbf{x}_{\text{P}}$ corresponds to the pilot location in the DD domain. Similarly, the received signal vector for the data frame is given by

$$\mathbf{y}_{\text{D}} = \mathbf{H}_{\text{eff}}\mathbf{x}_{\text{D}} + \mathbf{n}_{\text{D}}, \tag{16}$$

where $\mathbf{x}_{\text{D}}$ is the vectorized representation of the data frame. The received pilot vector $\mathbf{y}_{\text{P}}$ and received data vector $\mathbf{y}_{\text{D}}$ are stacked to form the composite vector $\mathbf{y}' \in \mathbb{C}^{2MN\times 1}$ as

$$\mathbf{y}' = \begin{bmatrix}\mathbf{y}_{\text{P}} \\ \mathbf{y}_{\text{D}}\end{bmatrix}. \tag{17}$$

The $\mathbf{y}'$ is converted to a real vector $\mathbf{y}'' \in \mathbb{R}^{4MN\times 1}$ by stacking the real and imaginary parts as

$$\mathbf{y}'' = \begin{bmatrix}\Re(\mathbf{y}') \\ \Im(\mathbf{y}')\end{bmatrix} = \begin{bmatrix}\Re(\mathbf{y}_{\text{P}}) \\ \Re(\mathbf{y}_{\text{D}}) \\ \Im(\mathbf{y}_{\text{P}}) \\ \Im(\mathbf{y}_{\text{D}})\end{bmatrix}, \tag{18}$$

where $\Re(.)$ and $\Im(.)$ denote real and imaginary parts, respectively. The $\mathbf{y}''$ vector is fed as the input to the DNN, making the number of input neurons to the DNN as $4MN$. The DNN processes the input vector $\mathbf{y}''$ and directly outputs the estimated information symbols in the corresponding data frame, making the number of output neurons to be MN, i.e., one neuron per information symbol in the data frame.

Training of the DNN is carried out using a dataset consisting of multiple realizations of $(\mathbf{y}'', \mathbf{x}_{\text{D}})$ pairs, where $\mathbf{y}''$ represents the input data and $\mathbf{x}_{\text{D}}$ represents the corresponding labels. The

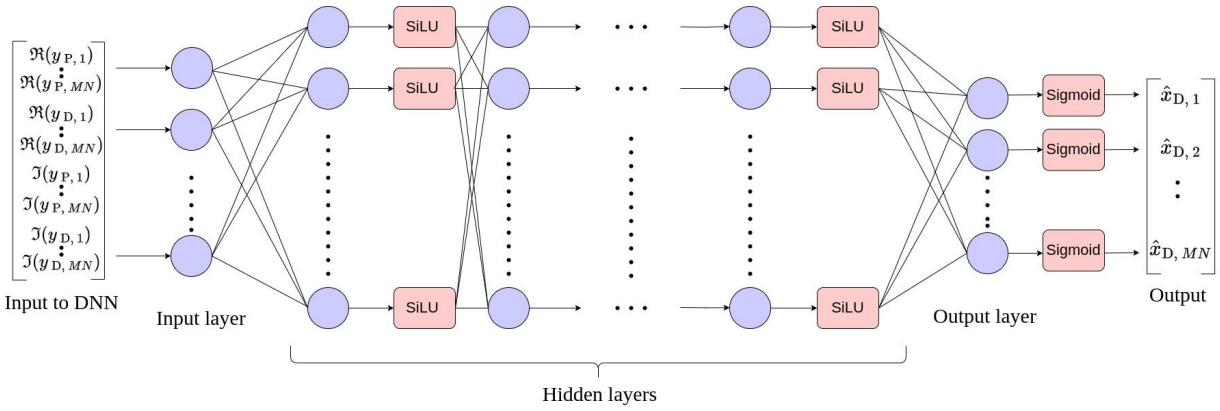

Fig. 2: Proposed DNN-based Zak-OTFS receiver architecture.

network aims to minimize the RMSE loss function $\mathcal{L}(\hat{\mathbf{x}}_{\mathrm{D}}, \mathbf{x}_{\mathrm{D}})$ defined as

$$\mathcal{L}(\hat{\mathbf{x}}_{\mathrm{D}}, \mathbf{x}_{\mathrm{D}}) = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(x_{\mathrm{D},i} - \hat{x}_{\mathrm{D},i})^2}, \tag{19}$$

where K is the total number of training samples, and $x_{\mathrm{D},i}$ and $\hat{x}_{\mathrm{D},i}$ are the true and predicted values of the ith sample. Optimization is performed using the Adam optimizer to enhance convergence efficiency and stability with a learning rate of 10^{-3} and follows a piecewise schedule, decreasing by a factor of 0.8 every 2 epochs[2]. The training dataset is generated for different pulse shaping filters. We consider sinc and Gaussian pulse shaping filters. The sinc filter is given by

$$w_{\mathrm{tx}}(\tau, \nu) = \sqrt{BT}\,\mathrm{sinc}(B\tau)\,\mathrm{sinc}(T\nu), \tag{20}$$

and the Gaussian filter is given by

$$w_{\mathrm{tx}}(\tau, \nu) = \left(\frac{2\alpha_\tau B^2}{\pi}\right)^{\frac{1}{4}} e^{-\alpha_\tau B^2 \tau^2} \left(\frac{2\alpha_\nu T^2}{\pi}\right)^{\frac{1}{4}} e^{-\alpha_\nu T^2 \nu^2}. \tag{21}$$

In order to ensure no bandwidth and time expansion beyond B and T, respectively, $\alpha_\tau = \alpha_\nu = 1.584$ is used for Gaussian filter. The receive filter at the receiver matched to the transmit filter is given by $w_{\mathrm{rx}}(\tau, \nu) = e^{j2\pi\tau\nu}\, w^*_{\mathrm{tx}}(-\tau, -\nu)$.

IV. Results and Discussions

In this section, we evaluate the performance of the proposed DNN receiver for Zak-OTFS. A comparison is made between our proposed DNN approach of joint I/O relation estimation and signal detection and the conventional approach where I/O relation estimation and signal detection are performed separately. For the conventional approach, we consider model-free I/O relation estimation [7] and MMSE detection. Simulations are performed for Zak-OTFS with the following parameters: Doppler period $\nu_{\mathrm{p}} = 15$ kHz, delay period $\tau_{\mathrm{p}} = \frac{1}{\nu_{\mathrm{p}}} = 66.66$ μs, $M = 12$, $N = 14$, total frame duration $T = N\tau_{\mathrm{p}} = 0.93$ ms, bandwidth $B = M\nu_{\mathrm{p}} = 180$ kHz,

[2]A higher initial learning rate speeds up the training and helps the network to prevent local minima. Gradually decreasing the learning rate reduces fluctuations around the optimal solution.

Path index (i)	1	2	3	4	5	6
Delay τ_i (μs)	0	0.31	0.71	1.09	1.73	2.51
Relative power (dB)	0	-1	-9	-10	-15	-20

TABLE I: Power delay profile of Vehicular-A channel model.

and BPSK. We consider the Vehicular-A channel model [26] with $P = 6$ paths, maximum Doppler shift $\nu_{\max} = 815$ Hz, maximum delay spread of $\tau_{\max} = 2.51\mu$s, and power delay profile given in Table I. The Doppler associated with the ith path is modeled as $\nu_i = \nu_{\max}\cos(\theta_i)$ where θ_is are independent and uniformly distributed in $[0, 2\pi)$. A fully-connected DNN architecture as shown below is used.

Proposed DNN: *Input $\rightarrow$ 5000 $\rightarrow$ SiLU $\rightarrow$ LayerNorm $\rightarrow$ 3000 $\rightarrow$ SiLU $\rightarrow$ LayerNorm $\rightarrow$ 2000 $\rightarrow$ SiLU $\rightarrow$ LayerNorm $\rightarrow$ 500 $\rightarrow$ Sigmoid $\rightarrow$ Output.*

The DNN consists of four hidden layers with 5000, 3000, 2000, and 500 neurons, respectively. The hidden layers use the sigmoid linear unit [23] (SiLU[3]) activation function, followed by Layer Normalization [24][4] in each layer to stabilize training and improve convergence. The SiLU activation function is used due to its non-monotonic behavior, which allows it to retain information from negative input values rather than discarding them entirely, as in the case of ReLU. This characteristic helps prevent information loss and improves gradient flow, particularly in deep networks. The output layer uses the Sigmoid activation function, which constrains the output within the range $[0, 1]$. A threshold of 0.5 is used to determine the transmitted bits. The parameters used in the proposed DNN are summarized in Table II.

In Fig. 3, the BER performance of the proposed DNN receiver is compared with the conventional receiver which employs model-free I/O relation estimation followed by MMSE detection. The comparison is made for sinc and Gaussian

[3]The SiLU activation function, also known as the Swish function, is defined as $\mathrm{SiLU}(x) = x \cdot \frac{1}{1+e^{-x}}$, where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. SiLU has been shown to outperform traditional activation functions like rectified linear unit (ReLU) in certain deep learning models (e.g., [25]). In our system also, we observed that ReLU did not perform well whereas SiLU performed quite well. Hence, we chose the SiLU activation function for the proposed architecture.

[4]Layer normalization (LayerNorm) normalizes the activations of each layer independently, which leads to smoother gradients and faster training.

Parameters	Prop. DNN (Sinc)	Prop. DNN (Gaussian)
No. of input neurons	$4MN = 672$	$4MN = 672$
No. of output neurons	$MN = 168$	$MN = 168$
No. of hidden layers	4	4
Hidden layer activation	SiLU	SiLU
Output layer activation	Sigmoid	Sigmoid
No. of training examples	3,500,000	5,000,000
No. of epochs	50	50
Optimization	Adam	Adam
Loss function	RMSE	RMSE
Pilot SNR	30 dB	30 dB
Data SNR	21 dB	5 dB, 24 dB
Batch size	2048	2048

TABLE II: Proposed DNN and training parameters for sinc and Gaussian filters.

filters. Additionally, as a performance benchmark, the performance of MMSE detection with perfect channel state information (CSI), i.e., perfect knowledge of $\mathbf{H}_{\text{eff}}$, is also shown. The training dataset for the sinc filter is generated at a pilot SNR of 30 dB and a data SNR of 21 dB. For the Gaussian filter, the training dataset is generated at a pilot SNR of 30 dB and data SNR values of 5 dB and 24 dB to account for variations across different SNR conditions. Sinc filter is found to perform better than Gaussian filter. This is because sinc filter has nulls at the information grid sampling points, whereas, at these sampling points, Gaussian filter has non-zero values close to the main lobe peak which causes interference. From Fig. 3, we can see that, as the pilot SNR increases, the BER performance of both the proposed DNN approach and the conventional approach improves and moves close to the perfect CSI performance. However, the proposed DNN approach achieves significantly better BER performance compared to that of the conventional approach. For example, at a pilot SNR of 20 dB, with Gaussian filter, the proposed DNN approach achieves a BER of 1.4×10^{-3} which is quite close to the perfect CSI BER of 1.2×10^{-3}, whereas the conventional approach achieves a BER of 7×10^{-3}. Likewise, for sinc filter, the DNN approach BER is 3×10^{-4} and the conventional approach BER is 1.5×10^{-3}. This demonstrates the effectiveness of the proposed DNN approach.

In Fig. 4, we present the BER performance of both sinc and Gaussian filters as a function of data SNR, while keeping the pilot SNR fixed at 15 dB. From Fig. 4, it is seen that, for both the filters, the proposed DNN approach significantly outperforms the conventional approach. For example, at a data SNR of 24 dB, the proposed DNN approach achieves more than an order of better BER performance compared to the conventional approach, demonstrating the ability of the proposed DNN to learn the I/O relation and optimize the detection performance.

In order to demonstrate the effectiveness of the proposed DNN-based approach for higher-order QAM, we use two DNNs as shown in Fig. 5. The first DNN is trained using the dataset $(\mathbf{y}'', \Re(\mathbf{x}_{\text{D}}))$, mapping $\mathbf{y}''$ to $\Re(\mathbf{x}_{\text{D}})$, while the second DNN is trained with $(\mathbf{y}'', \Im(\mathbf{x}_{\text{D}}))$, mapping $\mathbf{y}''$ to $\Im(\mathbf{x}_{\text{D}})$, where $\mathbf{x}_{\text{D}}$ denotes the transmitted vector consisting

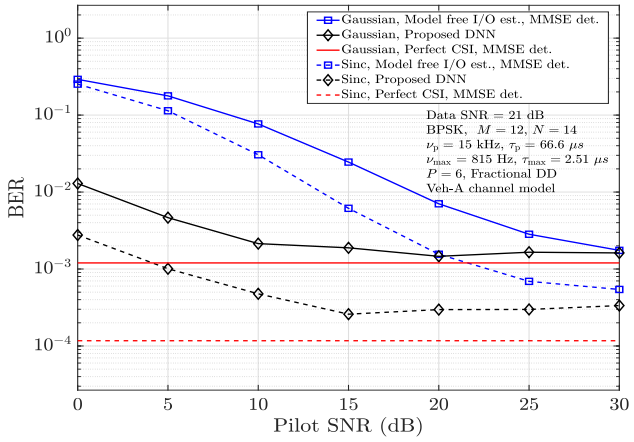

Fig. 3: BER performance of the proposed DNN receiver for sinc and Gaussian filters as function of pilot SNR.

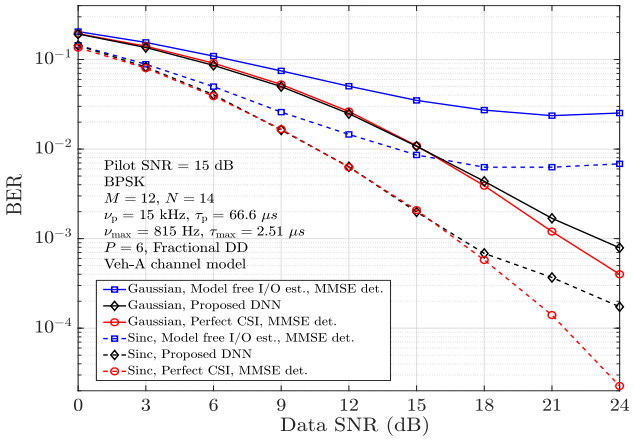

Fig. 4: BER performance of the proposed DNN receiver for sinc and Gaussian filters as function of data SNR.

of QAM symbols. At the receiver, the predicted outputs from the two DNNs, $\hat{\mathbf{x}}_{\text{D}_1}$ and $\hat{\mathbf{x}}_{\text{D}_2}$, corresponding to the real and imaginary parts, respectively, are combined to reconstruct the detected complex symbol vector as $\hat{\mathbf{x}}_{\text{D}} = \hat{\mathbf{x}}_{\text{D}_1} + j\hat{\mathbf{x}}_{\text{D}_2}$. which is then appropriately decoded to recover the transmitted bits. In Fig. 6, we present the BER performance of sinc and Gaussian filters for 8-QAM for a pilot SNR of 15 dB. The results indicate that the proposed DNN-based approach achieves better performance compared to the conventional approach. We also notice that the sinc filter performs better than the Gaussian filter in 8-QAM as well.

V. Conclusions

We proposed a DNN-based solution to the problem of joint I/O relation estimation and signal detection in Zak-OTFS receivers. A fully-connected network was proposed for this purpose, where the network was trained using synthetically generated pairs of pilot and data frames to directly minimize the RMSE between the actual and estimated data symbols. The network was able to learn the I/O relation internally and optimize the detection performance without explicitly

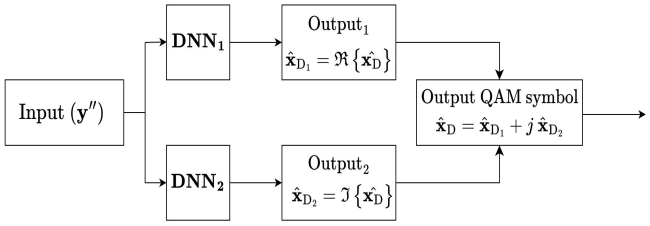

Fig. 5: Block diagram of the proposed DNN architecture for QAM-modulated Zak-OTFS signals.

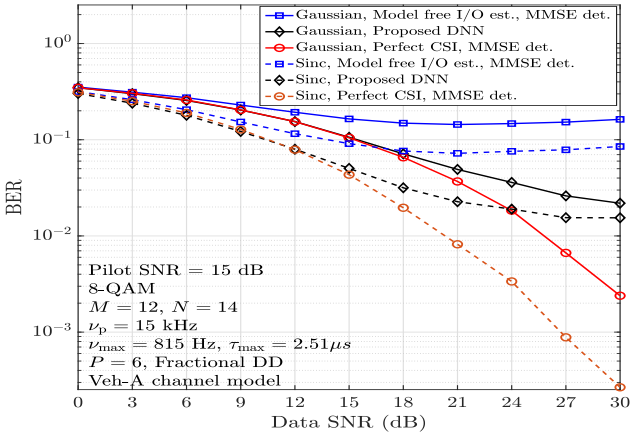

Fig. 6: BER performance of the proposed DNN-based receiver for 8-QAM as a function of data SNR.

estimating the I/O relation. Our simulation results for sinc and Gaussian filters using an exclusive pilot frame showed improved BER performance compared to the conventional approach where I/O relation estimation and signal detection are carried out separately. Future work can consider DNN-based solutions for joint I/O relation estimation and signal detection with embedded and superimposed pilot frames.

REFERENCES

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

[2] K. R. Murali and A. Chockalingam, "On OTFS modulation for high-Doppler fading channels," *Proc. ITA'2018*, pp. 1-10, Feb. 2018.

[3] 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.

[4] Best Readings in Orthogonal Time Frequency Space (OTFS) and Delay Doppler Signal Processing, https://www.comsoc.org/publications/best-readings/orthogonal-time-frequency-space-otfs-and-delay-doppler-signal-processing.

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

[6] 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 Inform. Theory Mag.*, vol. 2, no. 2, pp. 36-55, 1 Nov. 2022.

[7] 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 Inform. Theory Mag.*, vol. 3, no. 2, pp. 7-31, Jun. 2023.

[8] S. K. Mohammed, R. Hadani, and A. Chockalingam, *OTFS Modulation: Theory and Applications*, IEEE press-Wiley, Nov. 2024.

[9] S. Gopalam, I. B. Collings, S. V. Hanly, H. Inaltekin, S. R. B. Pillai, and P. Whiting, "Zak-OTFS implementation via time and frequency windowing," *IEEE Trans. Commun.*, vol. 72, no. 7, pp. 3873-3889, Jul. 2024.

[10] S. Gopalam, H. Inaltekin, I. B. Collings, and S. V. Hanly, "Optimal Zak-OTFS receiver and its relation to the radar matched filter," *IEEE Open J. of the Commun. Soc.*, vol. 5, pp. 4462-4482, 2024.

[11] M. Ubadah, S. K. Mohammed, R. Hadani, S. Kons, A. Chockalingam, and R. Calderbank, "Zak-OTFS for integration of sensing and communication," online arxiv.org/abs/2404.04182, 5 Apr 2024.

[12] J. Jayachandran, R. K. Jaiswal, S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "Zak-OTFS: pulse shaping and the tradeoff between time/bandwidth expansion and predictability," online arxiv.org/abs/2405.02718, 4 May 2024.

[13] B. Dabak, V. Khammammetti, S. K. Mohammed, and R. Calderbank, "Zak-OTFS and LDPC codes," *Proc. IEEE ICC'2024*, pp. 3785-3790, Jun. 2024.

[14] F. Jesbin and A. Chockalingam, "Near-optimal detection of Zak-OTFS signals," *Proc. IEEE ICC'2024*, pp. 4476-4481, Jun. 2024.

[15] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognitive Commun. and Netw.*, vol. 3, pp. 563-575, Dec. 2017.

[16] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114-117, Feb. 2018.

[17] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Comm. Lett.,* vol. 23, no. 4, pp. 652-655, Apr. 2019.

[18] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: a deep learning based Viterbi algorithm for symbol detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3319-3331, May 2020.

[19] L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo, "Deep learning for wireless communications: an emerging interdisciplinary paradigm," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 133-139, Aug. 2020.

[20] S. R. Mattu and A. Chockalingam, "Learning-based channel estimation and phase noise compensation in doubly-selective channels," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1052-1056, May 2022.

[21] S. R. Mattu, L. N. Theagarajan, and A. Chockalingam, "Deep channel prediction: a DNN framework for receiver design in time-varying fading channels," *IEEE Trans. Veh. Tech.*, vol. 71, no. 6, pp. 6439-6453, Jun. 2022.

[22] S. R. Mattu and A. Chockalingam, "Learning in time-frequency domain for fractional delay-Doppler channel estimation in OTFS," *IEEE Wireless Commun. Lett.*, vol. 13, no. 5, pp. 1245-1249, May 2024.

[23] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.

[24] J. Ba, J. Kiros, and G. Hinton, "Layer normalization," available online: arXiv:1607.06450v1 [stat.ML] 21 July 2016.

[25] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3-11, 2018.

[26] ITU-R M.1225, "Guidelines for evaluation of radio transmission technologies for IMT-2000," *International Telecommunication Union Radio communication*, 1997.