# Fractional Delay-Doppler Channel Estimation in OTFS with Sparse Superimposed Pilots using RNNs

Sandesh Rao Mattu and A. Chockalingam

Department of ECE, Indian Institute of Science, Bangalore 560012

*Abstract*—In this paper, we consider the problem of delay-Doppler (DD) channel estimation in orthogonal time frequency space (OTFS) modulation with fractional delays and Dopplers. Exclusive use of DD bins in a frame for pilot symbols causes rate loss. Superimposing pilot symbols over data symbols avoids this rate loss. Our contributions in this paper are two-fold. 1) We propose a sparse superimposed pilot (SSP) scheme where pilot and data symbols are superimposed in a few bins and the remaining bins carry data symbols only. This scheme offers the benefit of better inter-symbol leakage profile in a frame, while retaining full rate. 2) For the SSP scheme, we propose a recurrent neural network based learning architecture (referred to as SSPNet) trained to provide accurate channel estimates overcoming the leakage effects in channels with fractional DD. Simulation results show that the proposed SSP scheme along with fractional DD channel estimation using the proposed SSPNet performs better than a fully superimposed pilot scheme.

*Index Terms*—OTFS, fractional DD channel estimation, super-imposed pilots, deep learning, recurrent neural networks.

## I. Introduction

Orthogonal time frequency space (OTFS) modulation is known to offer robust performance in high-Doppler channels [1], [2]. OTFS multiplexes information symbols in the delay-Doppler (DD) domain. Also, time-varying channels when viewed in the DD domain are almost time-invariant and sparse. Pilot symbols are sent in OTFS frames for the purpose of channel estimation in the DD domain at the receiver [3] - [7]. The number of pilot symbols in an OTFS frame and where they are placed in the frame influence the rate and performance. Placement of pilot and data symbols in an OTFS frame can be done in different ways. Three widely considered pilot placement schemes in the OTFS literature include exclusive pilot scheme [3], embedded pilot scheme [4], and superimposed pilot scheme [5]. In the exclusive pilot scheme, a frame is used exclusively for pilot without any data. This scheme is simple, but it suffers rate loss because the full frame goes as overhead. In the embedded pilot scheme, each OTFS frame consists of a pilot symbol surrounded by guard bins, and the remaining bins carry data symbols. The guard bins are introduced to avoid pilot leakage into data bins and vice versa. The number of guard bins is chosen to accommodate the maximum delay and Doppler spreads of the channel, and these guard bins contribute to some rate loss. In superimposed pilot scheme, all bins carry data symbols and pilot symbols are superimposed on all these data symbols, i.e., there is no rate loss in this scheme and all frames are full rate frames. Since pilots are superimposed in all the

bins, we refer to this scheme in [5] as *fully superimposed pilot* (FSP) scheme. A drawback with this scheme, however, is the inter-symbol leakage/interference, which compromises performance. Iterative cancellation based techniques have been employed to alleviate this issue [5]. Another scheme in [6] uses one pilot symbol superimposed at the beginning of the frame for channel estimation. However the authors consider an ideal pulse at the transmitter, which is not realizable.

In this paper, we consider an alternate full rate achieving OTFS frame structure and devise a learning based channel estimation scheme for the same. Further, we consider the channel to have fractional DD which is practical. Our new contributions in this paper are two-fold. First, we propose a *sparse superimposed pilot* (SSP) scheme, where all the bins in a frame carry data symbols, and, in addition, pilot symbols are superimposed in some of these bins. This is in contrast with the FSP scheme in [5], where all bins carry both pilot and data symbols. The proposed scheme is motivated by the possibility of achieving a better inter-symbol leakage/interference profile while retaining full rate. Towards this, a lattice-type pilot placement pattern is adopted. Also, since the pilot symbols per frame are fewer in the proposed scheme, the energy per pilot symbol is high for the same total pilot energy per frame. Still, the leakage between the pilot and data symbols needs to be handled. Towards this, we propose a recurrent neural network (RNN) [8] based learning architecture trained to provide accurate fractional DD channel estimates overcoming the leakage effects. This forms the second contribution in the paper. We use long short term memory (LSTM) [9], a variant of RNN suited for learning dependencies in sequences. We obtain optimum number of pilot symbols and power allocation among pilot and data symbols through simulations. Our results show that the proposed SSP scheme along with channel estimation using the proposed SSPNet performs better than the FSP scheme in [5]. This is significant given that the FSP scheme in [5] assumes knowledge of the delay and Doppler taps and estimates only the channel gains, whereas the proposed SSPNet estimates all the delay taps, Doppler taps, and channel gains.

## II. System Model

The OTFS system model with fractional DD and rectangular transmit and receive pulses is derived as follows. Information symbols, $\mathbf{A}^{\text{DD}}[n, m]$s, each drawn from a modulation alphabet $\mathbb{A}$ are placed in an $M \times N$ DD grid given by $\left\{ \left( \frac{l}{M\Delta f}, \frac{k}{NT} \right), l = 0, \cdots, M - 1, k = 0, \cdots, N - 1 \right\}$, where $M$ is the number of delay bins, $N$ is the number of Doppler bins, $\Delta f$ is the subcarrier spacing, and $T = 1/\Delta f$. Bin sizes in the delay and Doppler domains are given by $1/M\Delta f$

and $1/NT$, respectively. The $\mathbf{A}^{\mathrm{DD}}[n,m]$s are converted to TF domain symbols $\mathbf{A}^{\mathrm{TF}}[k,l]$s using the ISFFT operation, as

$$\mathbf{A}^{\mathrm{TF}}[k,l] = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \mathbf{A}^{\mathrm{DD}}[n,m] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}, \quad (1)$$

for $l = 0, \cdots, N-1$ and $k = 0, \cdots, M-1$. The TF frame has duration $NT$ and bandwidth $M\Delta f$, where $T$ and $\Delta f$ are the sampling intervals along time and frequency, respectively, satisfying $T\Delta f = 1$. Equation (1) can be written in matrix form as $\mathbf{A}^{\mathrm{TF}} = \mathbf{F}_M \mathbf{A}^{\mathrm{DD}} \mathbf{F}_N^H \in \mathbb{C}^{M \times N}$, where $\mathbf{F}_M[m,n] = (1/\sqrt{M}) \exp(-j2\pi mn/M)$, $\mathbf{F}_N[m,n] = (1/\sqrt{N}) \exp(-j2\pi mn/N)$, and $(\cdot)^H$ represents the Hermitian operation. The TF domain samples, $\mathbf{A}^{\mathrm{TF}}[k,l]$s, are pulse shaped using transmit pulse $p_{\mathrm{tx}}(t)$ to generate a time domain signal $a(t)$. $a(t)$ sampled at rate $f_s = M\Delta f$ can be expressed in matrix form as $\mathbf{A}^t = \mathbf{P}_{\mathrm{tx}} \mathbf{F}_M^H \mathbf{A}^{\mathrm{TF}} = \mathbf{P}_{\mathrm{tx}} \mathbf{A}^{\mathrm{DD}} \mathbf{F}_N^H$, where $\mathbf{A}^t$ contains $MN$ samples of $a(t)$. The sampling interval is set to $T_s = 1/M\Delta f = T/M$ as per symbol spaced sampling [4], which results in $M$ length samples of the transmit and receive pulses. $\mathbf{P}_{\mathrm{tx}} \in \mathbb{C}^{M \times M}$ is a diagonal matrix, whose diagonal entries are obtained by uniformly sampling the transmit pulse $p_{\mathrm{tx}}(t)$ at time instants $mT/M, m = 0, 1, \cdots, M-1$. We consider the transmit pulse $p_{\mathrm{tx}}(t)$ and the receive pulse at the receiver $p_{\mathrm{rx}}(t)$ to be a rectangular pulse (i.e., $\mathbf{P}_{\mathrm{tx}} = \mathbf{P}_{\mathrm{rx}} = \mathbf{I}_M$, where $\mathbf{I}_M$ is $M \times M$ identity matrix).

Using the relation $\mathrm{vec}(\mathbf{XYZ}) = (\mathbf{Z}^T \otimes \mathbf{X})\mathrm{vec}(\mathbf{Y})$, where $\otimes$ is the Kronecker product, the time domain vector $\mathbf{a}^t = \mathrm{vec}(\mathbf{A}^t)$ can be written as

$$\mathbf{a}^t = \mathrm{vec}(\mathbf{A}^t) = \mathrm{vec}(\mathbf{P}_{\mathrm{tx}} \mathbf{A}^{\mathrm{DD}} \mathbf{F}_N^H) = (\mathbf{F}_N^H \otimes \mathbf{P}_{\mathrm{tx}})\mathbf{a}^{\mathrm{DD}}, \quad (2)$$

where $\mathbf{a}^{\mathrm{DD}} = \mathrm{vec}(\mathbf{A}^{\mathrm{DD}})$ and the operation $\mathrm{vec}(\mathbf{Z})$ vectorizes matrix $\mathbf{Z}$. Let $g(\tau, \nu)$ denote the complex baseband channel response in the DD domain. Then,

$$g(\tau, \nu) = \sum_{i=0}^{L-1} g_i \delta(\tau - \tau_i) \delta(\nu - \nu_i), \quad (3)$$

where $L$ is the number of channel paths in the DD domain, $\delta$ is the Kronecker delta function, and $g_i$, $\tau_i$, and $\nu_i$ denote the complex channel gain, delay, and Doppler, respectively, corresponding to the $i$th path. For fractional delays and Dopplers, $\tau_i = \frac{\alpha_i + a_i}{M\Delta f}$ and $\nu_i = \frac{\beta_i + b_i}{NT}$, where $\alpha_i = [\tau_i M\Delta f]^{\odot}$, $\beta_i = [\nu_i NT]^{\odot}$, and $[\cdot]^{\odot}$ denotes the nearest integer rounding operator with $-\frac{1}{2} < a_i, b_i < \frac{1}{2}$. At the OTFS receiver, the time domain signal, $b(t)$, is given by

$$b(t) = \int_\nu \int_\tau g(\tau, \nu) a(t-\tau) e^{j2\pi\nu(t-\tau)} \mathrm{d}\tau \mathrm{d}\nu + w(t), \quad (4)$$

where $w(t)$ represents the additive noise. A forward cyclic shift matrix defined as

$$\mathbf{\Pi} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{MN \times MN}, \quad (5)$$

and $\mathbf{\Delta}_i = \mathrm{diag}\Big\{ \exp\left(-\frac{j2\pi(\alpha_i + a_i)(\beta_i + b_i)}{MN}\right),$ $\exp\left(\frac{j2\pi(1 - (\alpha_i + a_i))(\beta_i + b_i)}{MN}\right), \cdots,$ $\exp\left(\frac{j2\pi(MN - 1 - (\alpha_i + a_i))(\beta_i + b_i)}{MN}\right) \Big\}$ model the delays and Dopplers, respectively, so that the channel matrix $\mathbf{G} \in \mathbb{C}^{MN \times MN}$ can be obtained as

$$\mathbf{G} = \sum_{i=0}^{L-1} g_i \mathbf{\Delta}_i \mathbf{\Pi}^{\lceil \alpha_i + a_i \rceil}. \quad (6)$$

The proof of the above equation is relegated to Appendix A. The discrete baseband vector $\mathbf{b}^t \in \mathbb{C}^{MN \times 1}$ of $b(t)$ can be represented as $\mathbf{b}^t = \mathbf{G}\mathbf{a}^t + \mathbf{w}$. The TF matrix $\mathbf{B}^{\mathrm{TF}} \in \mathbb{C}^{M \times N}$ is derived from $\mathbf{b}^t$ using Wigner transform, i.e., $\mathbf{B}^{\mathrm{TF}} = \mathbf{F}_M \mathbf{P}_{\mathrm{rx}} \mathbf{B}^t$, where $\mathbf{B}^t = \mathrm{vec}^{-1}(\mathbf{b}^t) \in \mathbb{C}^{M \times N}$, and $\mathbf{P}_{\mathrm{rx}} = \mathbf{I}_M$ for the considered rectangular receive pulse $p_{\mathrm{rx}}(t)$. The DD signal matrix, $\mathbf{B}^{\mathrm{DD}}$, is obtained from $\mathbf{B}^{\mathrm{TF}}$ as

$$\mathbf{B}^{\mathrm{DD}} = \mathbf{F}_M^H \mathbf{B}^{\mathrm{TF}} \mathbf{F}_N = \mathbf{P}_{\mathrm{rx}} \mathbf{B}^t \mathbf{F}_N. \quad (7)$$

This can be vectorized to obtain

$$\mathbf{b}^{\mathrm{DD}} = (\mathbf{F}_N \otimes \mathbf{P}_{\mathrm{rx}})\mathbf{b}^t = (\mathbf{F}_N \otimes \mathbf{P}_{\mathrm{rx}})(\mathbf{G}\mathbf{a}^t + \mathbf{w}). \quad (8)$$

Substituting (2) in (8), we get

$$\begin{aligned} \mathbf{b}^{\mathrm{DD}} &= (\mathbf{F}_N \otimes \mathbf{P}_{\mathrm{rx}})\mathbf{G}(\mathbf{F}_N^H \otimes \mathbf{P}_{\mathrm{tx}})\mathbf{a}^{\mathrm{DD}} + \mathbf{w}' \\ &= \mathbf{G}_{\mathrm{eff}}\mathbf{a}^{\mathrm{DD}} + \mathbf{w}', \end{aligned} \quad (9)$$

where $\mathbf{w}' = (\mathbf{F}_N \otimes \mathbf{P}_{\mathrm{rx}})\mathbf{w}$ and $\mathbf{G}_{\mathrm{eff}} \in \mathbb{C}^{MN \times MN} = (\mathbf{F}_N \otimes \mathbf{P}_{\mathrm{rx}})\mathbf{G}(\mathbf{F}_N^H \otimes \mathbf{P}_{\mathrm{tx}})$ is the effective channel matrix.

### A. Proposed sparse superimposed pilot (SSP) scheme

The receiver needs the knowledge of the channel for data detection. Pilot symbols are sent in OTFS frames for the purpose of estimating the channel at the receiver. Pilot and data symbols can be placed in a frame in different ways. As mentioned in Sec. I, there is rate loss in exclusive pilot scheme (where a frame consists of only a pilot symbol and no data symbols [3]) and embedded pilot scheme (where a frame consists of a pilot symbol surrounded by some guard bins and the remaining bins are occupied by data symbols [4]). Superimposed pilot schemes, where all bins are occupied by data symbols and pilot symbols are superimposed on data symbols, offer full rate frames. We consider two superimposed pilot schemes that achieve full rate. The first scheme is the full superimposed pilot (FSP) scheme proposed in [5], where all bins carry both pilot as well as data symbols as shown in Fig. 1(a). The second scheme is the one we consider in this paper, which we call sparse superimposed pilot (SSP) scheme. In the proposed SSP scheme, all bins carry data symbols and pilot symbols are superimposed on only a few bins as shown in Fig. 1(b). We sparsely place the pilots in a lattice-type arrangement where pilot symbols are spaced $S_\tau$ bins apart in the delay axis and $S_\nu$ bins apart in the Doppler axis. The advantages of doing this are that 1) by careful choice of $S_\tau$ and $S_\nu$ inter-symbol leakage/interference among the pilot symbols can be alleviated, and 2) this allows for higher energy per pilot symbol which helps to improve channel estimation accuracy, as will be seen in Sec. IV.

(a) Full superimposed pilot frame

(b) Sparse superimposed pilot frame
(Proposed)
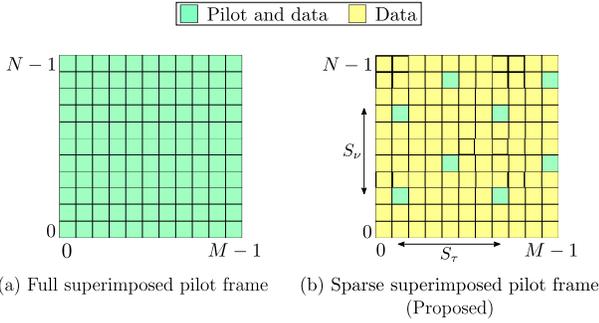
Fig. 1: Pilot and data symbols placement in the proposed SSP scheme and the FSP scheme in [5].

## III. SSPNet - Proposed DD channel estimator

In this section, we present the proposed SSPNet, an RNN based DD channel estimator network for the proposed SSP frame, its architecture, and training methodology. The motivation behind using RNN for channel estimation is that the symbols received corresponding to each pilot symbol transmitted can be viewed as a time sequence and RNNs are typically chosen for learning dependency in time sequences. Given a received SSP frame, the task is to obtain estimates of the channel parameters $(g_i, \tau_i, \nu_i), i = 0, \cdots, L - 1$. Figure 2 shows the architecture of the proposed SSPNet for channel estimation. The vector of received symbols in the DD domain, $\mathbf{b}^{\mathrm{DD}}$ (see (9)), is used to generate the input vector to the SSPNet, $\mathbf{b}'^{\mathrm{DD}}$, as outlined below. Let $n_{p_i}$ and $m_{p_i}$ denote the Doppler and delay indices for the $i$th pilot symbol, respectively (see Fig. 1b), where $i = 1, \cdots, N_p$, and $N_p$ is the number of pilot symbols superimposed in the SSP frame. The channel spreads the pilot symbols into their nearby DD bins. For the $i$th pilot, the spread is contained within the indices $n_{p_i} - n_\nu$ to $n_{p_i} + n_\nu$ on the Doppler axis, and $m_{p_i}$ to $m_{p_i} + m_\tau$ on the delay axis. Here, $m_\tau = \lceil \tau_{\max} M \Delta f \rceil$ and $n_\nu = \lceil \nu_{\max} NT \rceil$ are integers corresponding to maximum delay and Doppler spread, respectively. The received symbols in the bins of the $i$th pilot's spread area in the frame are extracted and vectorized to obtain the vector $\mathbf{b}_i'^{\mathrm{DD}} \in \mathbb{C}^{(2n_\nu+1)(m_\tau+1) \times 1}$. This is done for each pilot. The concatenated vector $\mathbf{b}'^{\mathrm{DD}} \in \mathbb{C}^{N_p(2n_\nu+1)(m_\tau+1) \times 1}$, given by

$$\mathbf{b}'^{\mathrm{DD}} = [\mathbf{b}_1'^{\mathrm{DD}} \ \mathbf{b}_2'^{\mathrm{DD}} \ \cdots \ \mathbf{b}_{N_p}'^{\mathrm{DD}}]^T, \tag{10}$$

is fed as the input to the SSPNet. The architecture of the SSPNet is designed and trained such that the same network, such that once trained, works for different $N_p$ values, SNRs, and DD profiles. The SSPNet obtains an estimate of the channel gain vector $\hat{\mathbf{g}} \in \mathbb{C}^{(2n_\nu+1)(m_\tau+1) \times 1}$. Among the $(2n_\nu+1)(m_\tau+1)$ entries in this vector, only those channel gain estimates whose absolute values are greater than a small threshold $\epsilon$ are picked as valid paths, i.e.,

$$\hat{g}_i = \begin{cases} 0, & \text{if } |\hat{g}_i| \le \epsilon \\ \hat{g}_i, & \text{otherwise.} \end{cases} \tag{11}$$

This is required as the output of SSPNet for an invalid path is close to but not equal to zero. The locations of the valid
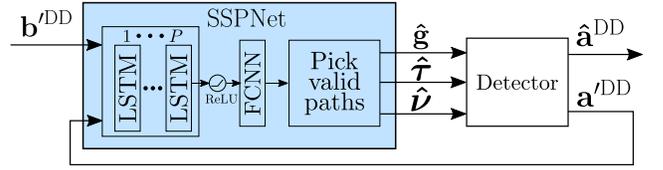


Fig. 2: Proposed RNN based channel estimation scheme.

| Parameter | Value |
|---|---|
| Number of LSTM layers ($P$) | 3 |
| LSTM hidden size ($h$) | 50 |
| LSTM input dimensions | $(c, s, 2)$ |
| LSTM output dimensions | $(c, s, 50)$ |
| FCNN input neurons | 50 |
| FCNN output neurons | $2(2n_\nu+1)(m_\tau+1)$ |

TABLE I: Parameters of the SSPNet architecture.

paths are then used to obtain the estimates for delay ($\hat{\tau}_i$s) and Doppler spreads ($\hat{\nu}_i$s) in the DD grid. Using the estimated vectors $\hat{\mathbf{g}}$, $\hat{\boldsymbol{\tau}}$, and $\hat{\boldsymbol{\nu}}$, the estimated DD domain channel matrix $\hat{\mathbf{G}}$ is obtained, which is used for detection of data symbols.

*Iterative scheme (SSP-I):* To further improve the accuracy of the channel estimates, the output of the detector, $\mathbf{a}'_{\mathrm{DD}}$, is fed back to the SSPNet for iteratively cancelling the effect of data symbols in channel estimation as follows. A new DD frame is constructed as

$$\mathbf{b}''^{\mathrm{DD}} = \mathbf{b}^{\mathrm{DD}} - \hat{\mathbf{G}} \mathbf{a}'^{\mathrm{DD}}. \tag{12}$$

The vector $\mathbf{b}'^{\mathrm{DD}}$ is computed again using (10) with $\mathbf{b}''^{\mathrm{DD}}$ in (12) as the received frame. This updated $\mathbf{b}'^{\mathrm{DD}}$ vector is given as input to the SSPNet and another set of refined channel estimates is obtained. This iterative procedure is repeated $N_{iter}$ times or until a convergence criterion is met. The procedure is stopped at $i$th iteration if $\|\hat{\mathbf{g}}^{(i)} - \hat{\mathbf{g}}^{(i-1)}\|^2 < \zeta$, i.e., the squared norm of the difference between the channel estimate vector at the $i$th iteration and the $(i-1)$th iteration is less than $\zeta$. The output of the detector at the end of the iterations, denoted by $\hat{\mathbf{a}}^{\mathrm{DD}}$, is taken as the final detected output.

### A. Architecture

The SSPNet architecture (see Fig. 2) consists of $P$ layers of long short-term memory (LSTM) [9], a variant of RNN. The output of the LSTM layers is passed through a rectified linear unit (ReLU) activation function, given by $\mathrm{ReLU}(x) = \max(0, x)$, $\forall x \in (-\infty, \infty)$, followed by a fully connected neural network (FCNN) layer. The FCNN is employed to reduce the dimension of the output of the LSTM network to the required dimension. A linear activation function with range $(-\infty, \infty)$ is used at the output of the FCNN. Using (11), valid paths are picked at the output of FCNN and $\hat{\mathbf{g}}$ thus obtained is the estimated channel gain vector. The other parameters of the SSPNet architecture are presented in Table I. The variable $c$ is the batch size and $s = N_p(2n_\nu+1)(m_\tau+1)$ is the sequence length. The output of the FCNN is a vector of dimension $2(2n_\nu+1)(m_\tau+1)$, where the the first $(2n_\nu+1)(m_\tau+1)$ dimensions are treated as real and the remaining as imaginary parts of the channel gain estimates.

| Parameter | Value |
|---|---|
| Epochs | 20000 |
| Optimizer | Adam |
| Learning rate | 0.001, divide by 2 every 4000 epochs |
| Batch size | 1000 |
| Mini-batch size | 64 |
| Refresh training data | Every epoch |

TABLE II: Hyper-parameters used for training the SSPNet.

### B. Training methodology

Data for training the SSPNet is obtained by synthetically generating multiple SSP OTFS frames with varying $N_p$. These frames are converted to time domain and sent through a time-varying fading channel. The received signal is converted back to DD domain. $N_p(2n_\nu+1)(m_\tau+1)$ symbols corresponding to the $N_p$ transmitted pilots are extracted from the received frame as per (10) to obtain the vector $\mathbf{b}'$ whose real and imaginary parts are concatenated. The ground truth data for training the SSPNet is obtained by generating a $(2n_\nu+1)(m_\tau+1)$ length true channel gain vector, $\mathbf{g}$. This vector is constructed such that the entries are channel gains only where there are valid paths and zeros elsewhere. During training, the weights of SSPNet are updated such that the L1 loss between the output of SSPNet, $\hat{\mathbf{g}}$, and ground truth $\mathbf{g}$, given by $L(g_i, \hat{g}_i) = |g_i - \hat{g}_i,|$, is minimized. The other hyper-parameters used in the training of the SSPNet are listed in Table II. Once the training is completed offline, the network weights are frozen. The same trained network can provide channel estimates for different SNRs, $N_p$ values, and DD profiles in the testing phase.

### IV. RESULTS AND DISCUSSIONS

In this section, we present the mean square error (MSE) and bit error rate (BER) performance of the proposed SSPNet for DD channel estimation. A carrier frequency of $f_c = 4$ GHz and a subcarrier spacing of $\Delta f = 15$ kHz are considered. Vehicular A (VehA) channel model [11] with $L = 6$ paths and a maximum speed of 220 km/h, i.e., a maximum Doppler $\nu_{\max}$ of 815 Hz, is considered. Each path has a Doppler generated using $\nu_i = \nu_{\max}\cos\theta_i$, where $\theta_i$ is uniformly distributed between $[-\pi, \pi]$. This implies that each path can have random Dopplers between $-\nu_{\max}$ and $\nu_{\max}$. We present results for the cases of integer DDs as well as fractional DDs. In the integer DDs case, the delay and Doppler taps are rounded off to the nearest integers. In the fractional DDs case, the fractional DD values are retained as such without rounding. The number of Doppler bins ($N$) and delay bins ($M$) are taken to be 12 and 64, respectively. Data symbols are chosen from BPSK alphabet and +1 is chosen as the pilot symbol. To train the network, the batch size ($c$) is chosen to be 1100 of which 1000 OTFS frames are used for training and 100 frames are used for validating the training. Each data symbol and pilot symbol has energy denoted by $\sigma_d^2$ and $\sigma_p^2$, respectively, with $\sigma_d^2 + \sigma_p^2 = 1$. This ensures that the average energy per frame is set to 1 for fair comparison with the FSP scheme in [5]. Data symbol detection is carried out using message passing (MP) detector in [4]. Maximum number of iterations ($N_{iter}$) in SSP-I scheme is 10, $S_\tau = 2m_\tau + 1 = 5$, $S_\nu = 4n_\nu + 1 = 5$, $\epsilon = 10^{-10}$, $\zeta = 10^{-6}$.



(a) FSP Tx frame, $N_p = MN = 768$  (b) SSP Tx frame, $N_p = 12$

(c) FSP Rx frame, $N_p = MN = 768$  (d) SSP Rx frame, $N_p = 12$
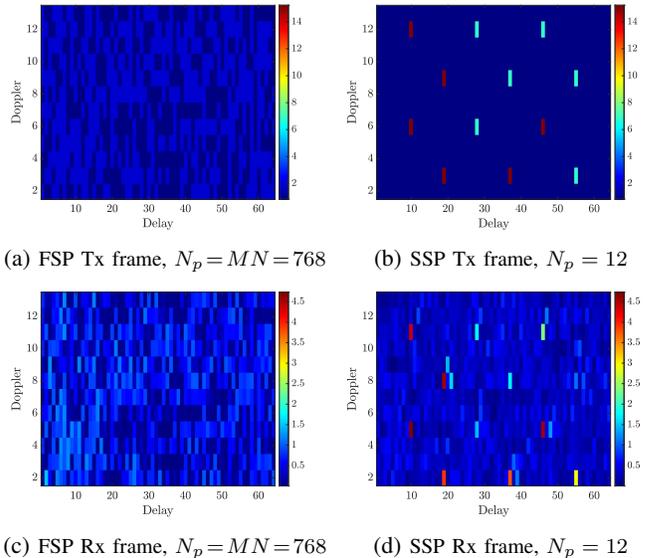
Fig. 3: Energy distribution in FSP and SSP frames at the transmitter and receiver for integer DD.



(a) BER vs $N_p$  (b) BER vs $\sigma_d^2$
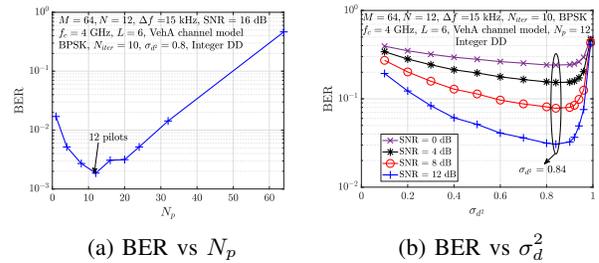
Fig. 4: BER performance of the proposed SSPNet as a function of $N_p$ and $\sigma_d^2$ for integer DD.

### A. Pilot energy spread in FSP and SSP frames

Figure 3 shows an example distribution of energy in various bins in FSP and SSP frames at the OTFS transmitter and receiver for integer DD. At the transmitter (Figs. 3a and 3b), there is no discernible difference between the pilot and data symbols in the FSP scheme, whereas, in the proposed SSP scheme, the pilot symbols have significantly higher energy ($MN\sigma_p^2/N_p$) than data symbols. This is because of fewer pilots in the SSP scheme for the same total pilot energy per frame. A similar trend is observed at the receiver as well (Figs. 3c and 3d), where the pilots and data symbols in the FSP scheme have leaked into one another, while the interference among the pilots is alleviated in the SSP scheme. Due to higher energy per pilot symbol in the SSP scheme, the corresponding received frame also contains high energy DD bins that help to achieve improved channel estimation accuracy.

### B. Choosing optimal $N_p$ and $\sigma_d^2$

Figure 4a shows the BER performance of using the proposed SSPNet for channel estimation as a function of number of pilots, $N_p$, in a frame (see Fig. 1b). When $N_p$ is small, the SSPNet has very few observations ($s$) to work with and the MSE and consequently the BER performance is poor. The BER performance improves as $N_p$ is increased. For

Fig. 5: MSE vs SNR of the proposed SSPNet compared with the FSP scheme in [5] for integer DD.
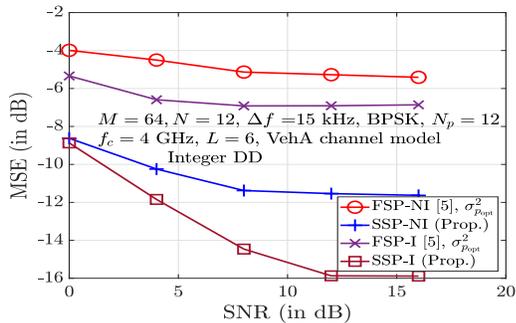


Fig. 6: BER vs SNR of the proposed SSPNet compared with the FSP scheme in [5] for integer DD.

$N_p > 20$, the energy per pilot symbol gets reduced and becomes comparable to the data symbol energy. This degrades the MSE of the channel estimates and therefore the BER increases. The BER attains its minimum value when $N_p = 12$. We choose this value of $N_p = 12$ in all the simulations that follow. Figure 4b shows the BER performance as a function of the data energy, $\sigma_d^2$. It is seen that the BER performance improves as the data energy $\sigma_d^2$ increases. Minimum BER is attained at $\sigma_d^2 = 0.84$, beyond which the BER increases. This is because $\sigma_p^2 < 0.16$ in this regime and the energy per pilot symbol reduces, resulting in poor accuracy of channel estimates. We fix $\sigma_d^2 = 0.84$ for the rest of the simulations. We note that the optimal energy allocation for the FSP scheme in [5] is $\sigma_{p_{\mathrm{opt}}}^2 = 0.3322$ and $\sigma_{d_{\mathrm{opt}}}^2 = 0.6678$. This choice works well for both integer as well as fractional DDs.

### C. MSE and BER comparison between SSP and FSP schemes for integer DD

The MSE performance of the proposed SSPNet without iterative cancellation (SSP-NI) and with iterative cancellation (SSP-I) with integer DD are presented in Fig. 5. The performance of the FSP-NI and FSP-I scheme in [5] are also plotted for comparison. It is seen that the MSE performance of the SSP-NI scheme is better than both the FSP-NI and FSP-I schemes. This is because, the proposed approach is able to cancel the interference between pilot and data symbols better through learning. Also, the SSP-I scheme achieves even better MSE performance. Figure 6 shows the BER performance of the proposed SSPNet as a function of SNR for integer DD. The performance of FSP-NI, FSP-I, and perfect CSI schemes are also plotted in an integer DD channel for comparison. It can be observed that the performance of the SSP-NI scheme is better than the FSP-NI scheme, with a performance gain of about 4 dB. For example, a BER of $4 \times 10^{-2}$ is attained at 8 dB with SSP-NI, while it is attained at 12 dB with FSP-NI scheme. It can also be observed that the SSP-I scheme performs better than the FSP-I scheme.

### D. MSE and BER comparison between SSP and FSP schemes for fractional DD

The MSE performance of the proposed SSPNet without iterative cancellation (SSP-NI) and with iterative cancellation (SSP-I) with fractional DD are presented in Fig. 7. The performance of FSP-NI and FSP-I schemes in [5] with fractional
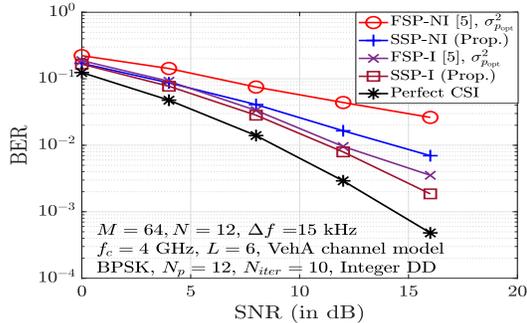
DD are also plotted for comparison. It is seen that the MSE performance of the SSP-NI scheme is superior compared to that of the FSP-NI scheme and FSP-I scheme in low and mid SNR regime, while the SSP-I scheme performs even better. Figure 8 shows the BER performance of the proposed SSPNet as a function of SNR with fractional DD. The performance of FSP-NI, FSP-I, and perfect CSI schemes with fractional DD are also plotted for comparison. The proposed SSP-NI scheme performs much better than the FSP-NI counterpart. For example, a BER of $6 \times 10^{-2}$ is attained at 8 dB with SSP-NI, while it is attained at 16 dB with FSP-NI scheme. It can also be observed that the SSP-I scheme performs better than the FSP-I scheme. These performance improvements are significant given that the FSP schemes in [5] assume that the estimates of $\tau_i$s and $\nu_i$s are perfectly known and only the $g_i$s are estimated, whereas the proposed SSPNet estimated all the three tuples $(\tau_i, \nu_i, g_i)$. This shows that superimposing the pilots sparsely as in the proposed SSP frame, and using a learning based SSPNet for channel estimation can achieve better MSE and BER performance.

*Complexity:* The proposed SSPNet contains 52518 parameters that need to trained offline, only once. Once trained, the SSP-NI scheme requires only 1018 floating point operations (FLOPs). This is in contrast to FSP-NI scheme in [5] which requires about $6.6 \times 10^6$ FLOPs. The high complexity of FSP-NI scheme is due to the computation of multiplications and inverses of matrices of size $MN \times MN$ to obtain the estimates. The low complexity of the SSPNet is due to processing only a few bins around the pilot locations in the frame.

## V. CONCLUSIONS

We proposed a new frame structure for OTFS having full rate with sparsely superimposed pilots for the purpose of channel estimation. For a given total pilot energy per OTFS frame, this scheme allowed higher pilot energy and reduced interference among the pilot symbols compared to fully super-imposed pilot scheme. To handle the leakage between pilot and data symbols, we proposed an LSTM based network called SSPNet. The proposed SSPNet was trained to provide the channel gain, delay, and Doppler estimates for each OTFS SSP frame. The proposed SSPNet was able to provide the channel estimates with good accuracy in integer as well as fractional DD channels. Performance results showed that the proposed
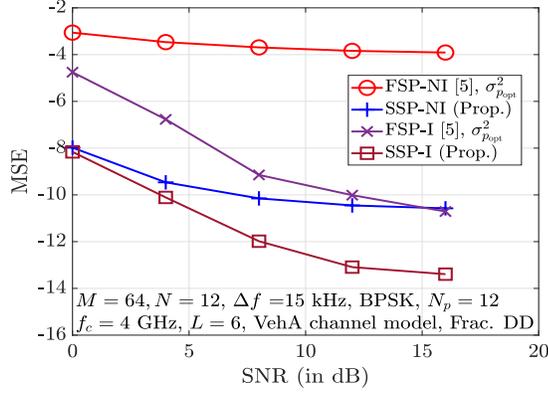
Fig. 7: MSE vs SNR of the proposed SSPNet compared with the FSP scheme in [5] for fractional DD.
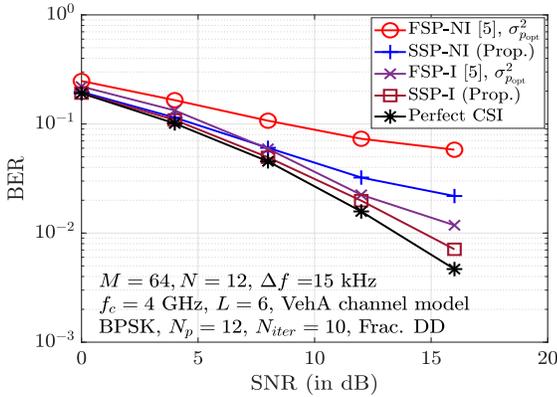


Fig. 8: BER vs SNR of the proposed SSPNet compared with the FSP scheme in [5] for fractional DD.

SSP frame along with the proposed SSPNet is an effective approach for fractional DD channel estimation.

## Appendix A
### Derivation of channel matrix with fractional DD and rectangular pulse

At the transmitter, the discrete signal is given by $\mathbf{a}^t = (\mathbf{F}_N^H \otimes \mathbf{P}_{\text{tx}})\mathbf{a}^{\text{DD}}$ from (2). This is converted to continuous time domain signal as

$$a^t(t) = \sum_{n=0}^{MN-1} \mathbf{a}^t[n] p_{\text{tx}}(t - nT_s), \tag{13}$$

where $T_s$ is the sampling instant with $MT_s = T = \frac{1}{\Delta f}$. For a rectangular pulse, $p_{\text{tx}}(t - nT_s) = \mathbb{1}_{\{nTs \leq t < (n+1)T_s\}}$, where $\mathbb{1}_{\{\cdot\}}$ is the indicator function. Equation (13) can then be simplified as

$$a^t(t) = \sum_{n=0}^{MN-1} \mathbf{a}^t[n] \mathbb{1}_{\{nTs \leq t < (n+1)T_s\}} = \mathbf{a}^t \left[ \left\lfloor \frac{t}{T_s} \right\rfloor \right]_{MN}, \tag{14}$$

where $\lfloor \cdot \rfloor$ denotes that flooring operation and $[\cdot]_{MN}$ denotes the mod $MN$ operation. At the receiver, the received signal, $b^t(t)$, is obtained as

$$b^t(t) = \sum_{i=0}^{L-1} g_i a^t(t - \tau_i) e^{j2\pi\nu_i(t-\tau_i)}, \tag{15}$$

where $L$ is the number of paths in the channel and $g_i, \tau_i = (\alpha_i + a_i)T_s$, and $\nu_i = \frac{\beta_i + b_i}{NT}$ are the channel gains, delay, and Doppler spread of the $i$th path, respectively. Substituting in the above equation, we get

$$b^t(t) = \sum_{i=0}^{L-1} g_i \mathbf{a}^t \left[ \left\lfloor \frac{t - (\alpha_i + a_i)T_s}{T_s} \right\rfloor \right]_{MN}$$
$$\times e^{\frac{j2\pi(\beta_i+b_i)}{NT}(t-(\alpha_i+a_i)T_s)}. \tag{16}$$

Sampling the continuous signal at $t = nT_s, n = 0, 1, \cdots$, and simplifying, we get

$$\mathbf{b}^t[n] = \sum_{i=0}^{L-1} g_i \mathbf{a}^t[n - \lceil \alpha_i + a_i \rceil]_{MN} e^{\frac{j2\pi(\beta_i+b_i)}{MN}(n-(\alpha_i+a_i))}. \tag{17}$$

This can be vectorized to obtain

$$\mathbf{b}^t = \sum_{i=0}^{L-1} g_i \boldsymbol{\Delta}_i \boldsymbol{\Pi}^{\lceil \alpha_i+a_i \rceil} \mathbf{a}^t = \mathbf{G}\mathbf{a}^t, \tag{18}$$

where $\mathbf{G} = \sum_{i=0}^{L-1} g_i \boldsymbol{\Delta}_i \boldsymbol{\Pi}^{\lceil \alpha_i+a_i \rceil}$, $\boldsymbol{\Pi}$ is as given in (5), and $\boldsymbol{\Delta}_i$ is as defined in Sec. II. ∎

### References

[1] R. Hadani, S. Rakib, M. Tsatsanis, A. Monk, A. J. Goldsmith, A. F. Molisch, and R. Calderbank, "Orthogonal time frequency space modulation," *Proc. IEEE WCNC'2017*, pp. 1-6, Mar. 2017.

[2] 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, Aug. 2018.

[3] 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.

[4] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for OTFS in delay–doppler channels," *IEEE Trans. Veh. Tech.*, vol. 68, no. 5, pp. 4906-4917, May 2019.

[5] H. B. Mishra, P. Singh, A. K. Prasad, and R. Budhiraja, "OTFS channel estimation and data detection designs with superimposed pilots," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2258-2274, Apr. 2022.

[6] C. Yang, J. Wang, Z. Pan, and S. Shimamoto, "Delay-Doppler frequency domain-aided superimposing pilot OTFS channel estimation based on deep learning," *Proc. IEEE VTC-Fall'2022*, pp. 1-6, Sept. 2022.

[7] V. K. Singh, M. F. Flanagan, and B. Cardiff, "Maximum likelihood channel path detection and MMSE channel estimation in OTFS systems," *Proc. IEEE VTC'2020-Fall*, pp. 1-5, Nov.-Dec. 2020.

[8] C. Sun, J. Fan, C. Chen, W. Li and W. Chen, "A two-stage neural network for sleep stage classification based on feature learning, sequence learning, and data augmentation, *IEEE Access*, vol. 7, pp. 109386-109397, 2019.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[10] P. Raviteja, Y. Hong, E. Viterbo and E. Biglieri, "Practical pulse-shaping waveforms for reduced-cyclic-prefix OTFS," *IEEE Trans. Veh. Tech.*, vol. 68, no. 1, pp. 957-961, Jan. 2019.

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