Learning based Delay-Doppler Channel Estimation with Interleaved Pilots in OTFS

Sandesh Rao Mattu and A. Chockalingam
Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore

***Abstract*—Traditionally, channel estimation in orthogonal time frequency space (OTFS) is carried out in the delay-Doppler (DD) domain by placing pilot symbols surrounded by guard bins in the DD grid. This results in reduced spectral efficiency as the guard bins do not carry information. In the absence of guard bins, there is leakage from pilot symbols to data symbols and vice versa. Therefore, in this paper, we consider an interleaved pilot (IP) placement scheme with a lattice-type arrangement (which does not have guard bins) and propose a deep learning architecture using recurrent neural networks (referred to as IPNet) for efficient estimation of DD domain channel state information. The proposed IPNet is trained to overcome the effects of leakage from data symbols and provide channel estimates with good accuracy (e.g., the proposed scheme achieves a normalized mean square error of about 0.01 at a pilot SNR of 25 dB). Our simulation results also show that the proposed IPNet architecture achieves good bit error performance while being spectrally efficient. For example, the proposed scheme uses 12 overhead bins (12 pilot bins and no guard bins) for channel estimation in a considered frame while the embedded pilot scheme uses 25 overhead bins (1 pilot bin and 24 guard bins).**

***Index Terms*—OTFS modulation, DD channel estimation, interleaved pilots, deep learning, recurrent neural networks.**

I. Introduction

High-mobility support is one of the critical requirements in next generation wireless systems. However, high-mobility scenarios are associated with high Doppler spreads in the channel which are detrimental to performance of multicarrier systems such as orthogonal frequency division multiplexing (OFDM). Orthogonal time frequency space (OTFS) modulation overcomes this issue by multiplexing information symbols in the delay-Doppler (DD) domain and viewing the channel in the DD domain where the channel varies slowly and is sparse, thereby offering robust performance in high Doppler spreads [1]-[3]. The time-invariance and sparsity of the DD channel simplifies the channel estimation task in OTFS [4],[5].

Several approaches for DD domain channel estimation have been reported in the OTFS literature [6]-[10]. In [6], the authors consider an impulse based channel estimation scheme, wherein an exclusive pilot frame consisting of one pilot symbol and zeros elsewhere in the frame is used for channel estimation. This scheme is simple but it results in poor spectral efficiency. The authors in [7],[8] use an embedded pilot frame, where the pilot symbol is surrounded by some guard bins with zeros and the remaining bins in the frame are occupied by data symbols. In this embedded pilot approach, the number of guard bins are chosen to accommodate the maximum delay spread and Doppler shift of the channel. This ensures that the spread from the pilot symbol does not interfere with the data symbols and vice versa. Although this alleviates the effects of interference between pilot and data symbols, this approach also incurs spectral efficiency loss due to guard bins. Estimation schemes based on compressed sensing/sparse Bayesian learning have been reported in the literature [9],[10].

Our current contribution differs from past works in two ways: 1) we consider an interleaved pilot scheme with no guard bins in the frame (hence offers better spectral efficiency), and 2) for this interleaved pilot setting, we propose a deep learning (DL) architecture using recurrent neural networks (RNN) for efficient channel estimation. The considered interleaved pilot scheme uses a lattice-type pilot structure in the DD grid (similar to the pilot structure in the time-frequency grid in OFDM [11]). Although the interleaved pilot approach is spectrally efficient, the absence of guard bins in the frame leads to leakage of pilot symbols into data symbols and vice versa. It is therefore important to pre-process the received pilot symbols before estimates of the channel can be obtained. RNNs have been used in the deep learning literature for drawing inference from correlated sequences [12]. In the proposed interleaved pilot scheme for channel estimation, the pilot symbols leak into the adjacent data bins resulting in a sequence of correlated received symbols. Therefore, the use of RNNs can be a natural approach for making inference (i.e., estimating the channel) from the correlated received symbols in the proposed scheme. Accordingly, we propose an RNN based DL network, which we refer to as interleaved pilot network (IPNet) for DD channel estimation. The proposed IPNet is trained to overcome the effects of leakage from data symbols and provide channel estimates with good accuracy. The obtained channel estimates are then used to remove the effect of leakage from pilot symbols in the detection of data symbols. To further improve the accuracy of the channel estimates, the detected symbols are used to further reduce the leakage from data symbols and another set of channel estimates is obtained from IPNet. This process is carried out in an iterative fashion. Our simulation results show the proposed IPNet achieves good mean square error and bit error rate performance with better spectral efficiency.

Notations: t_j represents the jth entry in the vector $\mathbf{t}$. $\lfloor x \rfloor$ represents the flooring operation on x. $(x)_M$ denotes the modulo M operation on x. $\mathbb{E}[.]$ denotes the expectation operator and $\|\mathbf{A}\|_F$ denotes the Frobenius norm of matrix $\mathbf{A}$.

The rest of the paper is organized as follows. The OTFS system model and the considered interleaved pilot scheme is presented in Sec. II. The details of the proposed IPNet for channel estimation are presented in Sec. III. Results

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

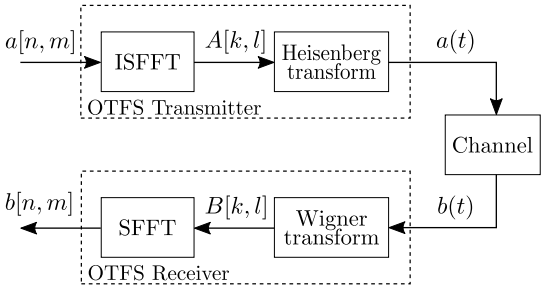
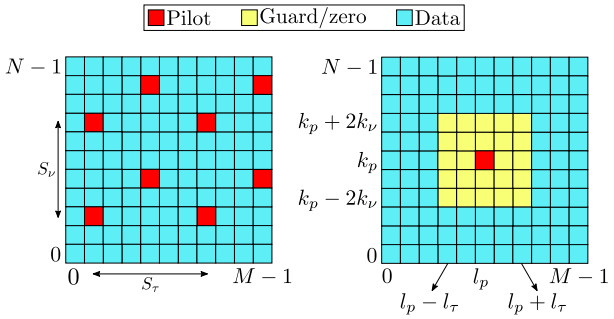


Fig. 1: OTFS modulation scheme.



(a) Interleaved pilot frame (b) Embedded pilot frame

Fig. 2: Pilot, guard, and data symbol placements in interleaved pilot and embedded pilot frames.

and discussions are presented in Sec. IV. Conclusions are presented in Sec. V.

II. OTFS System Model

The block diagram of the OTFS modulation scheme is shown in Fig. 1. The following set of operations are carried out at the OTFS transmitter. Information symbols are placed in the DD domain and they are converted to TF domain using inverse symplectic finite Fourier transform (ISFFT). The TF domain symbols are then mapped to time domain using Heisenberg transform. The resulting time domain signal is transmitted through the channel. At the OTFS receiver, the following inverse operations are carried out. First, the received time domain signal is converted to TF domain using Wigner transform. The TF domain symbols are then converted to DD domain using symplectic finite Fourier transform (SFFT).

Information symbols, $a[n,m]$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, Δf 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 $a[n,m]$s are converted to TF domain symbols $A[k,l]$s using the ISFFT operation, as

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

for $l = 0, \cdots, N-1$ and $k = 0, \cdots, M-1$. The time domain signal $a(t)$ is obtained from the TF symbols $A[n,m]$s using Heisenberg transform. Denoting the transmit pulse as $p_{tx}(t)$, $a(t)$ is obtained as

$$a(t) = \sum_{l=0}^{N-1} \sum_{k=0}^{M-1} A[k,l] p_{tx}(t - nT) e^{j2\pi k\Delta f(t - lT)}, \quad (2)$$

which is transmitted through the time-varying channel. 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, δ is the Kronecker delta function, and g_i, τ_i, and ν_i denote the complex channel gain, delay, and Doppler shift, respectively, corresponding to the ith path. 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. At the OTFS receiver, a matched filtering operation is carried out on the received time domain signal $b(t)$ with a receive pulse denoted by $p_{rx}(t)$, which results in the TF domain cross-ambiguity function, given by

$$F_{p_{rx},b}(t,f) = \int_{t'} p_{rx}^*(t'-t) b(t') e^{-j2\pi f(t'-t)} \mathrm{d}t', \quad (5)$$

where $(\cdot)^*$ represents the complex conjugation operation. The transmit and receive pulse are chosen such that they satisfy the biorthogonality condition, i.e., $F_{h_{rx}h_{tx}}(t,f)|_{t=nT, f=m\Delta f} = \delta(n)\delta(m)$. Sampling (5) at $t = lT$ and $f = k\Delta f$ gives

$$B[k,l] = F_{p_{rx},b}(t,f)|_{t=lT, f=k\Delta f}. \quad (6)$$

The sampled TF domain symbols are converted back to DD domain through SFFT operation to obtain $b[n,m]$ as

$$b[n,m] = \frac{1}{\sqrt{MN}} \sum_{l=0}^{N-1} \sum_{k=0}^{M-1} B[k,l] e^{-j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}. \quad (7)$$

The effective input-output relation in DD domain can be written combining (1)-(7) as [3]

$$b[n,m] = \sum_{i=0}^{L-1} g_i' a[(n-\beta_i)_N, (m-\alpha_i)_M] + w[n,m], \quad (8)$$

where $g_i' = g_i e^{-j2\pi\tau_i\nu_i}$, α_i is the integer corresponding to the index of delay tap and β_i is the integer corresponding to the Doppler frequency associated with τ_i and ν_i, respectively, i.e, $\tau_i = \frac{\alpha_i}{M\Delta f}$ and $\nu_i = \frac{\beta_i}{NT}$. Vectorizing (8), the input-output relation can be compactly written as

$$\mathbf{b} = \mathbf{G}\mathbf{a} + \mathbf{w}, \quad (9)$$

where $\mathbf{b}, \mathbf{a}, \mathbf{w} \in \mathbb{C}^{M\times N}$ and $\mathbf{G} \in \mathbb{C}^{MN\times MN}$ and the $(nM+m)$th entry of $\mathbf{a}$, $a_{nM+m} = a[n,m]$ for $n = 0, \cdots, N-1, m = 0, \cdots M-1$ and $a[n,m] \in \mathbb{A}$. Likewise, $b_{nM+m} = b[n,m]$ and $w_{nM+m} = w[n,m]$ for $n = 0, \cdots, N-1, m = 0, \cdots M-1$. g_is are assumed to be i.i.d. and are distributed as $\mathcal{CN}(0, 1/\sigma_{p_i}^2)$, with $\sum_i \sigma_{p_i}^2 = 1$.

Fig. 3: Proposed RNN based IPNet channel estimation scheme.

A. Pilot placement schemes

To obtain an estimate of the DD domain channel matrix $\mathbf{G}$, pilot symbols are placed in the DD grid and transmitted. These pilot symbols leak into the neighboring DD bins due to delay and Doppler spreads of the channel. At the receiver, the symbols corresponding to the transmitted pilots are used to obtain an estimate of the DD channel. The number of pilot symbols and how they are placed in an OTFS frame influence performance and spectral efficiency. Figure 2 shows two types of pilot placement schemes, namely, interleaved pilot scheme (Fig. 2a) and embedded pilot scheme (Fig. 2b), which are described below.

1) Embedded pilot scheme (Fig. 2b): The embedded pilot scheme is widely used in the OTFS literature [7],[8]. In this scheme, each frame consists of a pilot symbol (marked in red), guard symbols (marked in yellow), and data symbols (marked in blue), which can be represented as

$$\mathbf{a} = \begin{cases} 0, & \text{if } n = n_g, m = m_g \\ a_p, & \text{if } n = n_p, m = m_p \\ a_d, & \text{elsewhere,} \end{cases} \tag{10}$$

for $n = 0, \cdots, N-1$ and $m = 0, \cdots M-1$. In (10), n_gs and m_gs denote the indices of guard bands around the pilot symbol a_p, and the remaining indices are occupied by data symbols $a_d \in \mathbb{A}$. The pilot symbol is surrounded by guard symbols to alleviate interference from data symbols. The number of guard symbols are adjusted to accommodate the maximum delay spread $\tau_{\max}$ and maximum Doppler spread $\nu_{\max}$. While interference between pilot and data symbols are avoided in this scheme, spectral efficiency is compromised because of the presence of guard bins.

2) Interleaved pilot scheme (Fig. 2a) [11]: In this scheme, pilot symbols (marked in red) are placed across each frame in a lattice-type arrangement. The pilots are surrounded by data symbols (marked in blue) without any guard bins in between. The pilots are separated in the delay domain by S_τ bins and in Doppler domain by S_ν bins, which are chosen based on the number of pilots, N_p, in a frame, with the constraint $S_\tau > m_\tau$ and $S_\nu > n_\nu$, where m_τ and n_ν are the delay and Doppler taps corresponding $\tau_{\max}$ and $\nu_{\max}$, respectively. We will consider this interleaved pilot placement scheme, which has not been considered for OTFS before. While guard bins are avoided in this scheme, the receive signal processing must be capable of handling the effect of the leakage between pilot and data symbols. We propose an RNN based network for this very purpose in the next section.

III. IPNet – Proposed RNN based DD channel estimator

In this section, we present the proposed IPNet, an RNN based network for DD channel estimation, its architecture and training methodology. The block diagram of the proposed IPNet is presented in Fig. 3. Information symbols $a[n,m]$s are converted to time domain signal $a(t)$ at the OTFS transmitter and transmitted through a time-varying fading channel. At the OTFS receiver, the received signal, $b(t)$, is converted back to DD domain to obtain symbols $b[n,m]$, $n = 0, \cdots, N-1$, $m = 0, \cdots, M-1$, given by (8). The received DD frame is passed on to the IPNet block. At the IPNet block, the following set of operations are carried out to obtain the first set of channel estimates.

Let n_{p_i} and m_{p_i} denote the Doppler and delay indices for the ith pilot symbol, respectively (see Fig. 2a), where $i = 1, \cdots, N_p$. Due to the channel, the pilot symbol spreads into the nearby DD bins. For the ith 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. From the received OTFS frame, the symbols in these locations are extracted and vectorized to obtain the vector $\mathbf{b}'_i \in \mathbb{C}^{(2n_\nu+1)(m_\tau+1)\times 1}$ for the ith pilot. This is repeated for each pilot to obtain the vector $\mathbf{b}' \in \mathbb{C}^{N_p(2n_\nu+1)(m_\tau+1)\times 1}$, given by

$$\mathbf{b}' = [\mathbf{b}'_1, \mathbf{b}'_2, \cdots, \mathbf{b}'_{N_p}]. \tag{11}$$

Note that the vector $\mathbf{b}'$ contains the effect of both pilot and data symbols. The input to the IPNet is the vector $\mathbf{b}'$. The output of the IPNet is a vector $\hat{\mathbf{g}} \in \mathbb{C}^{(2n_\nu+1)(m_\tau+1)\times 1}$ of channel estimates. Among the $(2n_\nu+1)(m_\tau+1)$ entries in this vector, only those channel estimates are picked as valid paths for which the absolute value is greater than 4% of the maximum absolute value in the vector, i.e., if $\hat{g}_{\max} = \max_i |\hat{g}_i|$, then

$$\hat{g}_i = \begin{cases} 0, & \text{if } |\hat{g}_i| \leq 0.04\hat{g}_{\max} \\ \hat{g}_i, & \text{otherwise.} \end{cases} \tag{12}$$

This operation is carried out because the absolute value of the output of the IPNet for an invalid path is close to zero but not exactly zero. The locations of the valid paths are then used to obtain the estimates for integers corresponding to delay taps ($\hat{\alpha}_i$s) and Doppler frequencies ($\hat{\beta}_i$s) (see (8)) in the DD grid. Using the estimates $\hat{\mathbf{g}}$, $\hat{\boldsymbol{\alpha}}$, and $\hat{\boldsymbol{\beta}}$, the estimated DD domain channel matrix $\hat{\mathbf{G}}$ is obtained. This matrix is used for detection of data symbols. To further improve the accuracy of the channel estimates, the output of the detector, $a'[n,m]$, is fed back to the IPNet block for cancelling the effect of data symbols. A new DD frame is constructed as

$$\mathbf{b}'' = \mathbf{b} - \hat{\mathbf{G}}\mathbf{a}', \tag{13}$$

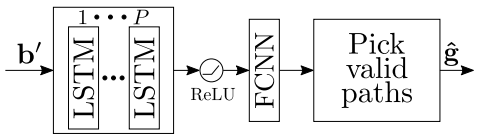


Fig. 4: Proposed RNN based IPNet architecture.

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 IPNet architecture.

where $\mathbf{a}' \in \mathbb{C}^{NM\times 1}$ ($\mathbf{b} \in \mathbb{C}^{NM\times 1}$) is the vectorized version of $a'[n, m]$ ($b[n, m]$). Vector $\mathbf{b}'$ is computed again using (11) and $\mathbf{b}''$ as the received frame. This is provided as input to the IPNet and another set of refined channel estimates are obtained. This iterative procedure is repeated P times and the output of the detector at the end of Pth iteration, $\hat{a}[n, m]$, is used to compute the bit error performance.

A. Architecture

The architecture of the proposed IPNet block is shown in Fig. 4. The architecture consists of P layers of long short-term memory (LSTM) [13],[14], a variant of RNN. The output of the LSTM layers is passed through a ReLU activation function, given by $\text{ReLU}(x) = \max(0, x)$, $\forall x \in (-\infty, \infty)$. The output of the LSTM layers is passed through a fully connected neural network (FCNN) with one layer. The FCNN is employed to reduce the dimension of the output of the LSTM network to the required dimension. Since the output of the FCNN layer needs to be the channel estimate, a linear activation function, with range between $(-\infty, \infty)$, is used at the output of the FCNN. Valid paths are picked from the resulting vector at the output of the FCNN layer using (12). The resulting vector $\hat{\mathbf{g}}$ is then returned as the channel coefficient vector. The other parameters of the IPNet architecture are presented in Table I. The variable c refers to 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 part of the channel estimates.

B. Training methodology

Training data is obtained by generating multiple OTFS frames with varying N_p (number of interleaved pilots). These frames are converted to time domain and transmitted 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 (11) to obtain the vector $\mathbf{b}'$. The real and imaginary parts of $\mathbf{b}'$ are concatenated before being provided as input to IPNet. For training the IPNet, the ground truth data is obtained by generating a $(2n_\nu + 1)(m_\tau + 1)$ length true channel estimate vector, $\mathbf{g}$. This vector is constructed such that the entries are channel estimates only where there are valid paths and zeros elsewhere. During training, the weights of IPNet are updated such that the L1 loss between the output of IPNet, $\hat{\mathbf{g}}$, and $\mathbf{g}$ is minimized. The L1 loss function is

$$L(\mathbf{g}, \hat{\mathbf{g}}) = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{g} - \hat{\mathbf{g}}|, \tag{14}$$

where N is the number of samples in the training set. The other hyper-parameters used while training the IPNet are presented in Table II. Note that this training needs to be carried out offline, only once. Subsequently, the network weights are frozen. New channel estimates are obtained from pilots in each OTFS frame using the same trained network. Further, as will be shown in Sec. IV, the trained network is able to work well for various N_p values, owing to the construction of training data.

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 IPNet.

C. Estimation of delay and Doppler indices

Once the IPNet is trained, the weights are frozen. During the inference (testing) phase, channel estimates, $\hat{\mathbf{g}}$, in the DD domain are obtained after picking the valid paths ((12)). To obtain the estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$, respectively, the following steps are followed. The channel estimate vector $\hat{\mathbf{g}} \in \mathbb{C}^{(2n_\nu+1)(m_\tau+1)\times 1}$ is reshaped into a matrix $\hat{\mathbf{H}} \in \mathbb{C}^{(2n_\nu+1)\times(m_\tau+1)}$. Index sets, $\mathcal{I}$ and $\mathcal{J}$ are defined to store the row and column indices of the non-zero elements in $\hat{\mathbf{H}}$, respectively. That is, $\mathcal{I} = \{i : \hat{\mathbf{H}}[i, j] \neq 0, i = 0, \cdots, 2n_\nu, j = 0, \cdots, m_\tau\}$ and $\mathcal{J} = \{j : \hat{\mathbf{H}}[i, j] \neq 0, i = 0, \cdots, 2n_\nu, j = 0, \cdots, m_\tau\}$. Then, for the pth path index, the estimate of delay and Doppler indices are obtained as

$$\hat{\alpha}_p = \mathcal{J}_p, \tag{15}$$

$$\hat{\beta}_p = \mathcal{I}_p - n_\nu. \tag{16}$$

We have carried out the simulations using PyTorch machine learning library [15],[16] on RTX 3090 GPU platform.

Remark on complexity: For the IPNet, the number of parameters to be learnt for P layers can be computed as $N_P = 4h^2(2P - 1) + 4hi_d + 8Ph + 100 \times 2(2n_\nu + 1)(m_\tau + 1) + 2(2n_\nu + 1)(m_\tau + 1)$, where h is the hidden size (see Table I), $i_d = 2$ is the input dimension, and $100\times 2(2n_\nu+1)(m_\tau+1)+2(2n_\nu+1)(m_\tau+1)$ is the number of parameters in the FCNN layer. For $P = 3$, $n_\nu = 1, m_\tau = 2$, and $h = 50$, $N_P = 52518$. Note that these parameters need to be learnt only once, offline. During the inference (testing) stage, only 1018 floating point operations (FLOPs) are required to compute the channel estimate. In contrast, the approach in [7] does not involve an offline training phase. Further, the number of FLOPs required is $5(2n_\nu+1)(m_\tau+1)$.

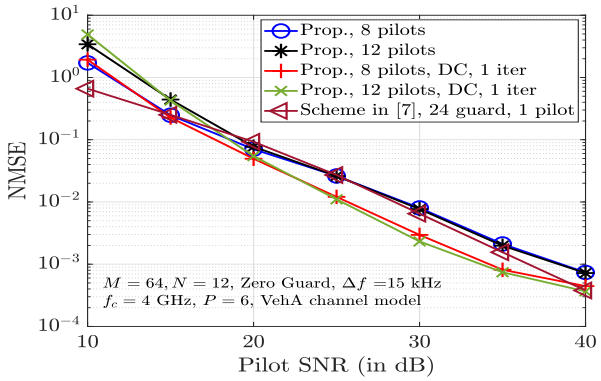


Fig. 5: NMSE performance of the proposed IPNet as a function of pilot SNR for different number of pilots.

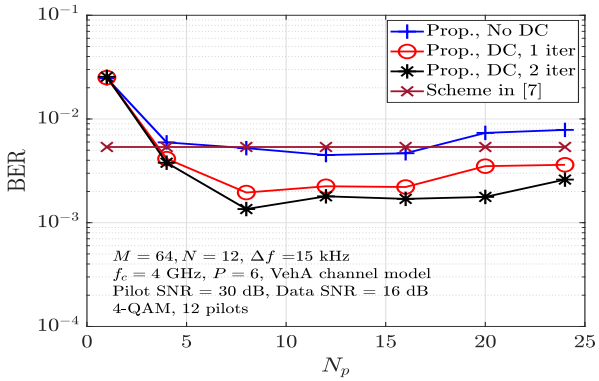


Fig. 6: BER of the proposed IPNet and the scheme in [7] as a function of number of pilots, N_p.

IV. Results and Discussions

In this section, we present the mean square error (MSE) and bit error rate (BER) performance of the proposed IPNet for DD channel estimation in OTFS. A carrier frequency of $f_c = 4$ GHz and a subcarrier spacing of $\Delta f = 15$ kHz are considered. We consider the Vehicular A (VehA) channel model [17],[18] with $L = 6$ paths and a maximum speed of 220 km/h. This speed at 4 GHz carrier frequency corresponds to a maximum Doppler shift, $\nu_{\max}$, of 815 Hz. Each path has a Doppler shift generated using Jakes model $\nu_i = \nu_{\max} \cos\theta_i$, where θ_i is assumed to be uniformly distributed between $[-\pi, \pi]$. We fix the number of Doppler bins (N) and delay bins (M) to be 12 and 64, respectively. A BPSK symbol +1 is used as the pilot symbol and data symbols are chosen from 4-QAM alphabet. 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.

To evaluate the accuracy of the channel estimates provided by the IPNet, we evaluate the normalized mean square error (NSME) for the DD domain channel matrix. The value of NMSE is computed as follows. The estimates of the channel coefficients, delay taps, and Doppler taps are obtained from the IPNet as described in Sec. III-C. Using these values, an estimate for the matrix $\mathbf{G}$ (see (9)), denoted by $\hat{\mathbf{G}}$, is obtained. The NMSE is computed as NMSE $= \mathbb{E}\left[\frac{\|\mathbf{G}-\hat{\mathbf{G}}\|_F^2}{\|\mathbf{G}\|_F^2}\right]$. For evaluating the BER performance, the message passing (MP) detector in [3] is used. Note that with the proposed approach, since the valid paths are chosen based on (12), which in turn depends on the energy of channel estimates, the matrix $\hat{\mathbf{G}}$ may have more non-zero entries than the actual channel matrix $\mathbf{G}$. Therefore, at low pilot SNRs (around 0 dB), where the noise energy is dominant, the NMSE can take values that are greater than 1. In all the simulations presented below, the pilot energy is kept same for all the competing schemes.

A. NMSE performance of IPNet

Figure 5 shows the NMSE performance of the proposed IPNet as a function of pilot SNR for $N_p = 8$ and 12. The pilot power is equally distributed among the N_p pilots. The NMSE performance of the embedded pilot scheme in [7] is also presented for comparison. The NMSE performance of the IPNet without data cancellation (DC) for both the N_p values is observed to be close to that of the scheme in [7] with the performance being slightly inferior at SNRs above 35 dB. However, with 1 iteration of DC (see (13)), the NMSE performance improves beyond the scheme in [7] in the mid and high SNR regime. Note that, for the considered parameters, n_ν is 1 and m_τ is 2. The scheme in [7], therefore, requires $(4n_\nu + 1)(2m_\tau + 1) = 25$ symbols for interference free estimation of channel coefficients, while a better NMSE performance is achieved from the proposed scheme using fewer DD bins (12 and 8 bins for N_p =12 and 8, respectively).

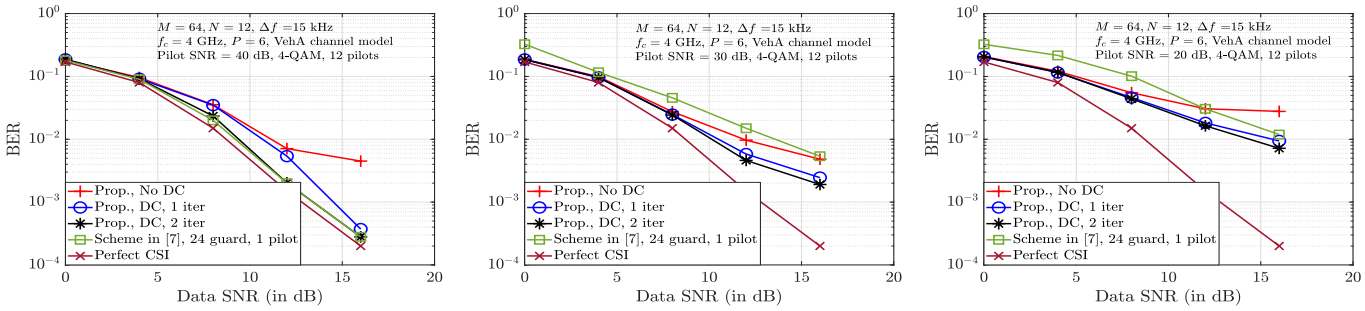
B. BER as a function of number of pilots

Figure 7 shows the BER performance of the proposed IPNet as a function of number of pilots, N_p, for a fixed pilot SNR of 30 dB and a fixed data SNR of 16 dB. The performance of the scheme in [7], with 1 pilot and 24 guard symbols, is also presented for comparison. With the increase in the the number of pilots, the BER performance is observed to improve for the proposed IPNet. This is because with increase in N_p, the sequence length s at the input of the IPNet also increases (see Table I). This allows the IPNet to provide estimates with better accuracy as more information is available at the input. However, with further increase in the number of pilots ($N_p > 16$), the BER performance is observed to increase slightly, owing to decrease in the energy per pilot symbol. This demonstrates that the proposed IPNet is able to work with different pilot densities and is able to perform better than the scheme in [7] when DC is employed.

C. BER vs SNR at different pilot SNRs

Figure 7 shows the BER performance of the proposed IPNet with 12 pilots for pilot SNRs of 40 dB, 30 dB, and 20 dB. In all the figures, the performance of the OTFS system with perfect CSI is presented for comparison. In addition, the BER performance of the embedded pilot scheme in [7] is also presented.

1) Pilot SNR = 40 dB: The BER performance of the proposed scheme with a pilot SNR of 40 dB is presented in Fig. 7a. It is seen that with no DC, the performance is close



(a) Pilot SNR = 40 dB (b) Pilot SNR = 30 dB (c) Pilot SNR = 20 dB

Fig. 7: BER performance comparison between the proposed IPNet with 12 pilots and the scheme in [7] for different pilot SNR values.

to that of the perfect CSI case till about 12 dB, after which the performance floors. This flooring is alleviated when DC is employed. With 1 iteration of DC, the performance improves closer to the perfect CSI performance, while with 2 iterations of DC, the performance matches that of the scheme in [7]. The performance is also very close to that of perfect CSI. For example, for a BER of 10^{-3}, the gap in data SNR is observed to be less than a dB.

2) Pilot SNR = 30 dB: Figure 7b shows the BER performance when the pilot SNR is 30 dB. The BER performance of the proposed IPNet with and without DC is observed to be better than that of the scheme in [7]. With DC, the performance improves with the improvement being larger when two iterations are used. With two iterations of DC and the proposed approach, there is significant gain observed over the scheme in [7]. For example, for a BER of 5×10^{-3} an SNR advantage of about 5 dB is observed.

3) Pilot SNR = 20 dB: The BER performance for a pilot SNR of 20 dB is presented in Fig. 7c. It is observed that the performances of the proposed IPNet and the scheme in [7] floor. In the low and mid SNR regime, the BER performance of the proposed IPNet is observed to better, after which the performance of IPNet without DC is observed to floor. With 1 and 2 iterations of DC, the proposed IPNet is observed to outperform the scheme in [7]. From the results presented above, it is seen that the proposed IPNet with interleaved pilots can achieve similar or better bit error performance compared to the embedded pilot scheme in [7], while being more spectrally efficient.

V. Conclusions

We introduced and investigated interleaved pilot scheme for DD channel estimation in OTFS systems. This pilot scheme does not contain guard bins in OTFS frames. To handle the leakage between the pilot and data symbols and to obtain good DD channel estimates, we proposed a multi-layer LSTM based learning network called IPNet. The proposed IPNet demonstrated good NMSE and BER performance compared to existing techniques in the literature. This performance advantage with the proposed IPNet was achieved while using fewer pilot resources, and therefore being spectrally efficient. Also, a learning based approach for DD channel estimation in OTFS with super-imposed pilots is an interesting topic for future research.

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] K. R. Murali and A. Chockalingam, "On OTFS modulation for high-Doppler fading channels," *Proc. ITA*, 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] S. S. Das and R. Prasad, *OTFS: Orthogonal Time Frequency Space Modulation A Waveform for 6G*, River Publishers, 2021.

[5] Y. Hong, T. Thaj, and E. Viterbo, *Delay-Doppler Communications: Principles and Applications*, Academic Press, 2022.

[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] 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.

[8] V. Kumar Singh, M. F. Flanagan, and B. Cardiff, "Maximum likelihood channel path detection and MMSE channel estimation in OTFS systems," *Proc. IEEE VTC'2020*, pp. 1-5, Feb. 2021.

[9] L. Zhao, W. -J. Gao and W. Guo, "Sparse Bayesian learning of delay-Doppler channel for OTFS system," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2766-2769, Dec. 2020.

[10] S. Srivastava, R. K. Singh, A. K. Jagannatham, and L. Hanzo, "Bayesian learning aided sparse channel estimation for orthogonal time frequency space modulated systems," *IEEE Trans. Veh. Tech.*, vol. 70, no. 8, pp. 8343-8348, Aug. 2021.

[11] S. Coleri, M. Ergen, A. Puri and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. Broadcasting*, vol. 48, no. 3, pp. 223-229, Sep. 2002.

[12] 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.

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

[14] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM – a tutorial into long short-term memory recurrent neural networks," available online: arXiv:1909.09586 [cs.NE] 12 Sep 2019.

[15] A. Paszke *et al.*, "PyTorch: an imperative style high-performance deep learning library," *NeurIPS'2019*, pp. 1-12, Dec. 2019.

[16] J. Han *et al.*, "An empirical study of the dependency networks of deep learning libraries," *Proc. IEEE ICSME'2020*, pp. 868-878, Nov. 2020.

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

[18] R. Jain, "Channel models: a tutorial," *WiMAX forum AATG*, vol. 10, Dept. CSE, Washington univ. St. Louis, 2007.