# A Delay-Doppler Dictionary Based I/O Relation Estimation in Zak-OTFS

Sai Pradeep Muppaneni, Sandesh Rao Mattu, and Ananthanarayanan Chockalingam

***Abstract*—In this letter, we propose a low-complexity delay–Doppler (DD) dictionary-based input–output relation (IOR) estimation scheme for Zak transform based orthogonal time frequency space (Zak-OTFS) modulation. The proposed approach approximates the IOR matrix as a linear combination of component matrices corresponding to a pre-constructed set of DD tuples, thereby avoiding explicit estimation of the underlying physical DD parameters. The path-gain coefficients associated with the DD tuples are obtained through a closed-form linear solution. By avoiding explicit DD parameter estimation and obtaining the path gains in a single linear step, the resulting estimation procedure has significantly lower computational complexity compared to existing model-dependent estimation approaches (e.g., sparse Bayesian learning (SBL)). Simulation results with fractional DDs and superimposed spread-pilot transmission demonstrate that the proposed scheme achieves good normalized mean square error performance and a bit error rate performance within 0.5 dB of the perfect channel state information benchmark, while requiring low complexity to mitigate pilot–data interference.**

***Index Terms*—Delay-Doppler domain, Zak transform, Zak-OTFS, input-output relation estimation, DD dictionary.**

## I. Introduction

Zak transform based orthogonal time frequency space (Zak-OTFS) modulation has attracted considerable interest due to its robustness under high Doppler conditions and its structural advantages over multicarrier (MC) OTFS [1], [2]. In Zak-OTFS, the input-output relation (IOR) admits a particularly simple form, expressed as a cascade of twisted convolutions. This structure allows the Zak-OTFS IOR matrix to be estimated in a model-free manner, without explicitly estimating the channel delays, Dopplers, or path gains under crystalline condition (where the maximum delay and Doppler spreads of the effective channel are less than the delay and Doppler periods, respectively) [2]. Owing to its simplicity, this model-free estimation approach has been widely adopted in the literature [2], [3]. However, the performance of the model-free approach is strongly influenced by the choice of transmit pulse shaping filter and pilot structure [3].

This raises the possibility of exploring model-dependent IOR estimation approaches (where the delay, Doppler and path gains are estimated explicitly to construct the IOR matrix) to improve estimation performance (e.g., as in MC-OTFS [4]–[6], [7], [8]). A direct extension of existing model-dependent approaches of IOR estimation to Zak-OTFS, however, is computationally expensive, particularly in fractional DD channels. One way to address this complexity issue is to approximate the delays and Dopplers using a predefined set of DD tuples (i.e., a DD dictionary) and then estimate the corresponding path gains. Although dictionary-based representations have previously been explored in the context of compressed sensing based IOR estimation [6], [9], the proposed method in this paper differs fundamentally in how the dictionary is utilized and how the associated path gains are estimated. In the proposed scheme, the DD dictionary is used only to approximate the DD parameters representation, whereas the corresponding path gains are obtained through *a closed-form linear solution*. In contrast, existing dictionary-based approaches in the literature typically rely on iterative sparse recovery algorithms (e.g., sparse Bayesian learning (SBL) [6]), which require repeated updates of the path-gain coefficients. The key novelty in our current work therefore lies in avoiding the iterative path gain updates in previous works and instead estimating the path gains in a single linear step. This leads to a much simpler IOR estimation procedure for Zak-OTFS that significantly reduces the computational complexity compared to other dictionary-based sparse recovery methods (this will be illustrated through numerical results in Sec. IV). The key contributions of this work can be summarized as follows.

- A DD dictionary-based framework is proposed to estimate the IOR in Zak-OTFS without explicitly estimating the underlying DD parameters.
- The path gains corresponding to the DD dictionary tuples are obtained using a closed-form linear solution which significantly reduces the computational complexity. For example, simulation results show that, for the same bit error rate (BER) performance, the SBL algorithm in [6] requires an average simulation run time of 12.76 s whereas the proposed algorithm requires only 3.83 s, demonstrating the complexity advantage.
- Simulation results for the Veh-A, EVA, and TDL channel models with fractional DDs demonstrate that the proposed scheme achieves very good normalized mean square error (NMSE) performance and a BER performance within 0.5 dB of the perfect channel state information (CSI) benchmark. This is an attractive performance for a superimposed pilot frame which is affected by pilot-data interference over the entire DD frame.

## II. System Model

Consider a set of $MN$ information symbols $\mathbf{X} \in \mathbb{A}^{M\times N}$, where $\mathbb{A}$ denotes the modulation alphabet, and $M$ and $N$ denote the number of bins along the delay and Doppler domains, respectively. In Zak-OTFS, the information symbols are embedded in the DD domain. To obtain a quasi-periodic[1] continuous DD signal corresponding to $\mathbf{X}$, a discrete quasi-periodic extension of $\mathbf{X}$ is formed as $x_{\rm dd}[r,s] = \mathbf{X}[r \bmod$

S. P. Muppaneni and A. Chockalingam are with the Department of ECE, Indian Institute of Science, Bangalore 560012, India. E-mail: {saimuppaneni; achockal}@iisc.ac.in. S. R. Mattu is currently with Cohere Technologies Inc., San Jose, USA. Email: sandesh.mattu@cohere-tech.com.

[1]Only those two-dimensional DD signals that are quasi-periodic in nature admit a valid time-domain (TD) representation through the Zak transform. A two-dimensional signal $a(\tau,\nu)$ is said to be quasi-periodic if and only if $a(\tau+n\tau_{\rm p},\nu+m\nu_{\rm p}) = a(\tau,\nu)e^{j2\pi n\frac{\nu}{\nu_{\rm p}}}$.

$M,\, s \bmod N]\, e^{j2\pi \lfloor \frac{r}{M} \rfloor \frac{s}{N}}$, where $r, s \in \mathbb{Z}$. A continuous DD signal $x_{\rm dd}(\tau,\nu)$ is constructed as

$$x_{\rm dd}(\tau,\nu) = \sum_{r,s\in\mathbb{Z}} x_{\rm dd}[r,s]\,\delta\big(\tau - r\tfrac{\tau_{\rm p}}{M}\big)\,\delta\big(\nu - s\tfrac{\nu_{\rm p}}{N}\big)$$
$$= \sum_{m,n\in\mathbb{Z}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \mathbf{X}[k,l] e^{j2\pi\frac{nl}{N}} \delta\left(\tau - (k+nM)\tfrac{\tau_{\rm p}}{M}\right)\delta\left(\nu - (l+mN)\tfrac{\nu_{\rm p}}{N}\right), \quad (1)$$

where $\delta(\cdot)$, $\tau_{\rm p}$ and $\nu_{\rm p}$ denote the Dirac delta function, delay period and Doppler period, respectively, with $\tau_{\rm p}\nu_{\rm p} = 1$. The bandwidth and time duration of the signal in (1) are limited to $B = M\nu_{\rm p}$ and $T = N\tau_{\rm p}$, respectively, by filtering it through a DD domain transmit pulse shaping filter $w_{\rm tx}(\tau,\nu)$ as $x_{\rm dd}^{w_{\rm tx}}(\tau,\nu) = w_{\rm tx}(\tau,\nu) *_\sigma x_{\rm dd}(\tau,\nu)$, where $*_\sigma$ denotes the twisted convolution operator[2]. The DD signal $x_{\rm dd}^{w_{\rm tx}}(\tau,\nu)$ is transformed to TD using inverse Zak transform as [2]

$$x(t) = \mathcal{Z}_t^{-1}(x_{\rm dd}^{w_{\rm tx}}(\tau,\nu)) = \sqrt{\tau_{\rm p}}\int_0^{\nu_{\rm p}} x_{\rm dd}^{w_{\rm tx}}(t,\nu)\,d\nu. \quad (2)$$

The transmitted signal $x(t)$ passes through a doubly-selective channel characterized by the DD impulse response $h(\tau,\nu) = \sum_{i=1}^{P} h_i\,\delta(\tau-\tau_i)\,\delta(\nu-\nu_i)$, where $P$ is the number of paths, and $h_i$, $\tau_i$, and $\nu_i$ denote the path gain, delay, and Doppler shift of the $i$th path, respectively. The received signal at the receiver is given by[3] [1], [2]

$$y(t) = \sum_{i=1}^{P} h_i\, x(t-\tau_i)\, e^{j2\pi\nu_i(t-\tau_i)} + n(t), \quad (3)$$

where $n(t)$ denotes additive white Gaussian noise (AWGN). The received signal $y(t)$ is converted into the DD domain using Zak transform as [2],

$$y_{\rm dd}(\tau,\nu) = \mathcal{Z}_t(y(t)) = \sqrt{\tau_{\rm p}}\sum_{k\in\mathbb{Z}} y(\tau + k\tau_{\rm p})\, e^{-j2\pi k\nu\tau_{\rm p}}. \quad (4)$$

Using the associative property of twisted convolution, the received DD domain signal $y_{\rm dd}(\tau,\nu)$ can be expressed as [3]

$$y_{\rm dd}(\tau,\nu) = (h(\tau,\nu) *_\sigma w_{\rm tx}(\tau,\nu)) *_\sigma x_{\rm dd}(\tau,\nu) + n_{\rm dd}(\tau,\nu), \quad (5)$$

where $n_{\rm dd}(\tau,\nu)$ is the Zak transform of $n(t)$. The DD signal $y_{\rm dd}(\tau,\nu)$ is filtered using a receive filter matched to the transmit filter, as $w_{\rm rx}(\tau,\nu) = w_{\rm tx}^*(-\tau,-\nu)\, e^{j2\pi\tau\nu}$ to obtain $y_{\rm dd}^{w_{\rm rx}}(\tau,\nu) = w_{\rm rx}(\tau,\nu) *_\sigma y_{\rm dd}(\tau,\nu)$, which can be written as

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

where $h_{\rm eff}(\tau,\nu)$ denotes the effective DD domain channel, and $n_{\rm dd}^{w_{\rm rx}}(\tau,\nu)$ is the filtered noise in the DD domain. Sampling $y_{\rm dd}^{w_{\rm rx}}(\tau,\nu)$ at $\tau = k'\frac{\tau_{\rm p}}{M}$ and $\nu = l'\frac{\nu_{\rm p}}{N}$, for $k' = 0,1,\ldots,M-1$ and $l' = 0,1,\ldots,N-1$, yields the discrete DD domain received samples $\mathbf{Y} \in \mathbb{C}^{M\times N}$ as $\mathbf{Y}[k',l'] = y_{\rm dd}^{w_{\rm rx}}(\tau,\nu)\big|_{(\tau,\nu)=(k'\frac{\tau_{\rm p}}{M}, l'\frac{\nu_{\rm p}}{N})}$. Substituting (1) and (6) into the sampled received signal, we obtain

$$\mathbf{Y}[k',l'] = \sum_{r',s'\in\mathbb{Z}} h_{\rm eff}[k'-r', l'-s'] x_{\rm dd}[r',s'] e^{j2\pi\frac{r'(l'-s')}{MN}}$$
$$= \sum_{m,n\in\mathbb{Z}} \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} h_{\rm eff}[k'-k-nM, l'-l-mN]\, \mathbf{X}[k,l] e^{j2\pi\frac{nl}{N}} e^{j2\pi\frac{(l'-l-mN)(k+nM)}{MN}} + n[k',l'], \quad (7)$$

where $h_{\rm eff}[r,s] = h_{\rm eff}(r\frac{\tau_{\rm p}}{M}, s\frac{\nu_{\rm p}}{N})\ \forall\ r,s\in\mathbb{Z}$, and $n[k',l'] = n_{\rm dd}^{w_{\rm rx}}(k'\frac{\tau_{\rm p}}{M}, l'\frac{\nu_{\rm p}}{N})$. Finally, by vectorizing $\mathbf{X}$ and $\mathbf{Y}$, the DD domain input–output relation can be compactly written as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (8)$$

where $\mathbf{x} = \text{vec}(\mathbf{X}) \in \mathbb{A}^{MN\times 1}, \mathbf{y} = \text{vec}(\mathbf{Y}) \in \mathbb{C}^{MN\times 1}$, $\text{vec}(\cdot)$ denotes the column vectorization of a matrix, and $\mathbf{H}\in\mathbb{C}^{MN\times MN}$ is the IOR matrix, whose entries are given by

$$\mathbf{H}[l'M+k', lM+k] = \sum_{m,n\in\mathbb{Z}} h_{\rm eff}[k'-k-nM, l'-l-mN]\, e^{j2\pi\frac{nl}{N}} e^{j2\pi\frac{(l'-l-mN)(k+nM)}{MN}}, \quad (9)$$

where $k',k \in \{0,\ldots,M-1\}$ and $l',l \in \{0,\ldots,N-1\}$. In (9), the indices $(k,l)$ correspond to the transmitted signal located at $(k\frac{\tau_{\rm p}}{M}, l\frac{\nu_{\rm p}}{N})$ in the DD domain, while $(k',l')$ denotes the received signal sampled at $(k'\frac{\tau_{\rm p}}{M}, l'\frac{\nu_{\rm p}}{N})$. The summations over $m$ and $n$ account for the quasi-periodic replicas in the transmitted DD signal. Consequently, arguments $k'-k-nM$ and $l'-l-mN$ in $h_{\rm eff}$ represent the contribution of $(n,m)$th quasi-periodic replica of the transmitted symbol at $(k,l)$ to the received sample at $(k',l')$. In computing the entries of $\mathbf{H}$ as per (9), we limit the summation to $|m| \le m_{\max}$, $|n| \le n_{\max}$, and $m_{\max} = n_{\max} = 2$ is found to give adequate accuracy [3].

*1) Alternate form of system model:* To facilitate IOR estimation, we rewrite (8) in an equivalent form as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} = \underbrace{\big[\,\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ \cdots\ \boldsymbol{\phi}_P\,\big]}_{\triangleq\,\boldsymbol{\Phi}}\mathbf{h} + \mathbf{n}, \quad (10)$$

where the matrix $\boldsymbol{\Phi} \in \mathbb{C}^{MN\times P}$ has as its columns $\boldsymbol{\phi}_i$s, which are obtained as $\boldsymbol{\phi}_i = \mathbf{G}_i\,\mathbf{x} \in \mathbb{C}^{MN\times 1}$, where $\mathbf{G}_i = \mathbf{H}(1,\tau_i,\nu_i)$ is the *ith path component matrix* defined as the $\mathbf{H}$ matrix corresponding to a channel that contains only the $i$th path with unit gain, delay $\tau_i$, and Doppler $\nu_i$. The vector $\mathbf{h} = [h_1\ h_2\ \cdots\ h_P]^T \in \mathbb{C}^{P\times 1}$ denotes the path gain vector. We note that the alternate representation in (10) is a key component that is exploited in the proposed dictionary based scheme for reconstructing the IOR matrix at the receiver.

*2) Superimposed spread-pilot frame:* We consider a superimposed spread-pilot frame, where a frame consists of $MN$ data symbols on which a pilot symbol spread in the DD domain (for IOR estimation) is superimposed over the entire frame [10]. It has the advantage of all the $MN$ bins in a frame to be used for data. However, this requires the receiver signal processing to deal with pilot-data interference. Let $\mathbf{X}_{\rm p,spr} \in \mathbb{C}^{M\times N}$ denote the pilot component of the superimposed spread-pilot frame. The spread-pilot is generated by filtering the quasi-periodic extension of an exclusive pilot frame, containing a pilot symbol $\alpha \in \mathbb{C}$ at its center and zeros elsewhere, using a chirp-based spreading filter implemented

[2]Twisted convolution is defined as $a(\tau,\nu) *_\sigma b(\tau,\nu) = \iint a(\tau',\nu')b(\tau-\tau',\nu-\nu')e^{j2\pi\nu'(\tau-\tau')}d\tau'd\nu'$.

[3]The path gains, delays, and Doppler shifts are assumed to remain approximately constant over one Zak-OTFS frame, as commonly adopted in OTFS system models. For example, at a mobile speed of 220 km/h and a frame duration of 3 ms, the terminal travels only about 18.3 cm. Over such a small distance, the physical propagation paths do not change appreciably, and hence the corresponding DD-domain path parameters can be reasonably assumed to remain approximately unchanged over the frame duration.

through $MN$-periodic twisted convolution[4] [10]. The chirp spreading filter is given by $w[r,s] = e^{j2\pi q\frac{r^2+s^2}{MN}}$, where $q$ is the chirp slope parameter. Let $x_{\mathrm{dd,exc}}$ denote the quasi-periodic extension of the exclusive pilot frame. The spread-pilot component is then obtained as [10]

$$\begin{aligned}\mathbf{X}_{\mathrm{p,spr}}[k,l] &= w[k,l] \circledast_{\sigma} x_{\mathrm{dd,exc}}[k,l] \\ &= \alpha \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} e^{j2\pi n\frac{(l_p+mN)}{N}} e^{j2\pi\frac{(l-l_p-mN)(k_p+nM)}{MN}} \\ &\quad \times w[k-k_p-nM,\, l-l_p-mN]. \end{aligned} \tag{11}$$

For the case when $M$ and $N$ are odd primes, and when the chirp slope parameter $q$ is relatively prime to both $M$ and $N$, the magnitude of $\mathbf{X}_{\mathrm{p,spr}}[k,l]$ becomes constant across all DD bins [10]. This ensures that the pilot energy is evenly spread over the entire frame. The data component $\mathbf{X}_{\mathrm{d,spr}} \in \mathbb{A}^{M\times N}$ is then superimposed onto the pilot component to obtain the overall frame $\mathbf{X}_{\mathrm{spr}} \in \mathbb{C}^{M\times N}$ as $\mathbf{X}_{\mathrm{spr}} = \mathbf{X}_{\mathrm{p,spr}} + \mathbf{X}_{\mathrm{d,spr}}$. Upon reception, the pilot contribution is used to reconstruct the IOR matrix, which is then used to detect the data symbols.

## III. Proposed I/O Relation Estimation

Using the alternate system model in (10), the received signal $\mathbf{y}_{\mathrm{spr}}$ can be written as

$$\mathbf{y}_{\mathrm{spr}} = \underbrace{\left(\mathbf{\Phi}_{\mathrm{p,spr}} + \mathbf{\Phi}_{\mathrm{d,spr}}\right)}_{\triangleq \mathbf{\Phi}_{\mathrm{spr}}} \mathbf{h} + \mathbf{n}, \tag{12}$$

where $\mathbf{\Phi}_{\mathrm{p,spr}} = \left[\,\mathbf{G}_1\mathbf{x}_{\mathrm{p,spr}} \;\cdots\; \mathbf{G}_P\mathbf{x}_{\mathrm{p,spr}}\,\right]$ and $\mathbf{\Phi}_{\mathrm{d,spr}} = \left[\,\mathbf{G}_1\mathbf{x}_{\mathrm{d,spr}} \;\cdots\; \mathbf{G}_P\mathbf{x}_{\mathrm{d,spr}}\,\right]$, where $\mathbf{x}_{\mathrm{p,spr}}$ and $\mathbf{x}_{\mathrm{d,spr}}$ denote the vectorized pilot and data components of the transmitted frame, respectively. Instead of estimating the true DD tuples explicitly, we approximate the physical channel by a DD dictionary covering the channel support. Let $k_{\max} = \lceil \tau_{\max} M/\tau_p \rceil$ and $l_{\max} = \lceil \nu_{\max} N/\nu_p \rceil$ denote the maximum delay and Doppler shift indices, respectively, where $\lceil\cdot\rceil$ denotes ceiling operator. The DD dictionary is constructed as $\mathcal{D} = \tilde{\boldsymbol{\tau}} \times \tilde{\boldsymbol{\nu}}$, with $\tilde{\boldsymbol{\tau}} = \left\{ \frac{u\tau_p}{Ms_\tau} \;\middle|\; u = 0,1,\ldots,k_{\max}s_\tau \right\}$ and $\tilde{\boldsymbol{\nu}} = \left\{ \frac{v\nu_p}{Ns_\nu} \;\middle|\; v = -l_{\max}s_\nu,\ldots,l_{\max}s_\nu \right\}$, where $s_\tau, s_\nu \in \mathbb{R}^+\backslash\{0\}$ are the delay and Doppler discretization parameters, respectively. The corresponding delay and Doppler grid spacings are $\tau_p/(Ms_\tau)$ and $\nu_p/(Ns_\nu)$, respectively. Hence, $s_\tau$ and $s_\nu$ determine the dictionary resolution, whereas $\tau_{\max}$ and $\nu_{\max}$ determine its support. Using these DD tuples, the corresponding path component matrices $\{\tilde{\mathbf{G}}_i\}_{i=1}^{|\mathcal{D}|}$ are generated as $\tilde{\mathbf{G}}_i = \mathbf{H}(1,\tilde{\tau}_i,\tilde{\nu}_i)$. The $\tilde{\mathbf{\Phi}}_{\mathrm{spr}}$ matrix is then given by

$$\begin{aligned}\tilde{\mathbf{\Phi}}_{\mathrm{spr}} &= \underbrace{\left[\tilde{\mathbf{G}}_1\mathbf{x}_{\mathrm{p,spr}} \;\cdots\; \tilde{\mathbf{G}}_{|\mathcal{D}|}\mathbf{x}_{\mathrm{p,spr}}\right]}_{\triangleq \tilde{\mathbf{\Phi}}_{\mathrm{p,spr}}} + \underbrace{\left[\tilde{\mathbf{G}}_1\mathbf{x}_{\mathrm{d,spr}} \;\cdots\; \tilde{\mathbf{G}}_{|\mathcal{D}|}\mathbf{x}_{\mathrm{d,spr}}\right]}_{\triangleq \tilde{\mathbf{\Phi}}_{\mathrm{d,spr}}} \\ &= \tilde{\mathbf{\Phi}}_{\mathrm{p,spr}} + \tilde{\mathbf{\Phi}}_{\mathrm{d,spr}}. \end{aligned} \tag{13}$$

From (12), the maximum-likelihood (ML) estimate of the path gain vector is

$$\hat{\mathbf{h}}_{\mathrm{spr}} = \left(\tilde{\mathbf{\Phi}}_{\mathrm{spr}}^H \mathbf{R}^{-1} \tilde{\mathbf{\Phi}}_{\mathrm{spr}}\right)^{-1} \tilde{\mathbf{\Phi}}_{\mathrm{spr}}^H \mathbf{R}^{-1} \mathbf{y}_{\mathrm{spr}}, \tag{14}$$

where $\mathbf{R}$ denotes the noise covariance matrix. Since $\tilde{\mathbf{\Phi}}_{\mathrm{d,spr}}$ depends on the (unknown) data symbols, a direct computation of the path gain vector is not possible. We therefore adopt an iterative approach in which channel reconstruction and data detection aid each other. This is summarized in **Algorithm 1**.

**Algorithm 1** Proposed IOR estimation algorithm

1: **Input:** $\{\tilde{\mathbf{G}}_i\}_{i=1}^{|\mathcal{D}|}$, $\tilde{\mathbf{\Phi}}_{\mathrm{p,spr}}$, $\mathbf{y}_{\mathrm{spr}}$, $\mathbf{R}^{-1}$, $I_{\max}$
2: **Initialize:** $\hat{\mathbf{x}}_{\mathrm{d,spr}}^{(0)} = \mathbf{0}_{MN\times 1}$, $t \leftarrow 1$
3: **repeat**
4: $\quad \tilde{\mathbf{\Phi}}_{\mathrm{d,spr}}^{(t)} = \left[\tilde{\mathbf{G}}_1\hat{\mathbf{x}}_{\mathrm{d,spr}}^{(t-1)} \;\cdots\; \tilde{\mathbf{G}}_{|\mathcal{D}|}\hat{\mathbf{x}}_{\mathrm{d,spr}}^{(t-1)}\right]$
5: $\quad \tilde{\mathbf{\Phi}}_{\mathrm{spr}}^{(t)} \leftarrow \tilde{\mathbf{\Phi}}_{\mathrm{p,spr}} + \tilde{\mathbf{\Phi}}_{\mathrm{d,spr}}^{(t)}$
6: $\quad \hat{\mathbf{h}}_{\mathrm{spr}}^{(t)} = (\tilde{\mathbf{\Phi}}_{\mathrm{spr}}^{(t)H} \mathbf{R}^{-1} \tilde{\mathbf{\Phi}}_{\mathrm{spr}}^{(t)})^{-1} \tilde{\mathbf{\Phi}}_{\mathrm{spr}}^{(t)H} \mathbf{R}^{-1} \mathbf{y}_{\mathrm{spr}}$
7: $\quad \hat{\mathbf{H}}_{\mathrm{spr}}^{(t)} = \sum_{i=1}^{|\mathcal{D}|} \hat{h}_{\mathrm{spr}}^{(t)}[i]\, \tilde{\mathbf{G}}_i$
8: $\quad$ Detect $\hat{\mathbf{x}}_{\mathrm{d,spr}}^{(t)}$ using $\hat{\mathbf{H}}_{\mathrm{spr}}^{(t)}$
9: $\quad t \leftarrow t+1$
10: **until** $t > I_{\max}$ or $\|\hat{h}_{\mathrm{spr}}^{(t)} - \hat{h}_{\mathrm{spr}}^{(t-1)}\| < \eta$
11: **Output:** $\hat{\mathbf{H}}_{\mathrm{spr}} \leftarrow \hat{\mathbf{H}}_{\mathrm{spr}}^{(t)}$

The algorithm begins by initializing the unknown data vector to zeros. Consider the algorithm in its $t$th iteration. Using the data detected in the $(t-1)$th iteration, we first construct the data contribution to the $\tilde{\mathbf{\Phi}}_{\mathrm{spr}}$ matrix, i.e., $\tilde{\mathbf{\Phi}}_{\mathrm{d,spr}}$ (step 4, **Algorithm 1**). This updated contribution is then combined with the pilot contribution part to obtain the updated $\tilde{\mathbf{\Phi}}_{\mathrm{spr}}$ matrix (step 5). With this updated matrix, the path gain vector is re-estimated using the expression in (14). The estimated coefficients are then used to form the IOR matrix for the current iteration (step 7), from which an updated detection[5] of the data symbols is obtained. The next iteration proceeds with this updated data decision. This process continues until a maximum number of iterations $I_{\max}$ is reached.

*Remark:* In this work, the DD dictionary is constructed using uniformly spaced DD tuples with constant discretization parameters $s_\tau$ and $s_\nu$, so that the estimator remains independent of channel statistics. If channel statistics are available, $s_\tau$ and $s_\nu$ need not be constant and can be chosen adaptively, with larger values in regions of high channel energy and smaller values in low-energy regions. Such a choice can reduce approximation error and improve convergence rate.

*1) Computational complexity:* A key feature of the proposed approach is that the DD dictionary tuples are created a priori. Because of this, the component matrices $\tilde{\mathbf{G}}_i$ can be precomputed and stored offline, thereby avoiding their regeneration during real-time IOR estimation. This results in a substantial reduction in real-time computational load, with the trade-off being an increased memory requirement[6]. With the precomputation in footnote 6 in place, the per-iteration complexity in terms of complex multiplications

[4]Let $a[r,s]$ be $MN$-periodic and $b[r,s]$ be quasi-periodic. Their $MN$-periodic twisted convolution is defined as $a[r,s] \circledast_{\sigma} b[r,s] = \sum_{r'=0}^{M-1}\sum_{s'=0}^{N-1} a[r',s']\, b[r-r', s-s']\, e^{j2\pi s'\frac{(r-r')}{MN}}$.

[5]Data detection is carried out on the residual signal obtained by subtracting the estimated pilot component in the received signal, $\hat{\mathbf{H}}^{(t)}\mathbf{x}_p$, from $\mathbf{y}_{\mathrm{spr}}$.

[6]For a frame of size $M \times N$, the total memory needed for storing all $\tilde{\mathbf{G}}_i$ component matrices is $M_{\mathrm{c}}|\mathcal{D}|M^2N^2$ bytes, where $M_{\mathrm{c}}$ is memory in bytes per complex number. Storing all component matrices may become prohibitive for large $M, N$. In such cases, instead of storing all $\{\tilde{\mathbf{G}}_i\}_{i=1}^{|\mathcal{D}|}$, one may store only the corresponding effective channel samples $\tilde{h}_{\mathrm{eff}}^{(i)}[k,l]$, and generate the $\tilde{\mathbf{G}}_i$ matrix as needed for some paths in dictionary. This significantly reduces memory usage. For example, for a $64 \times 24$ system, storing a $\tilde{\mathbf{G}}_i$ matrix requires approximately 37.75 MB of memory, whereas storing the corresponding $\tilde{h}_{\mathrm{eff}}^{(i)}$ values requires only about 876.3 KB.

TABLE I: Per-iteration computational complexity of the proposed and SBL-based estimation schemes.

| Operation/scheme | No. of complex-valued multiplications |
|---|---|
| Algorithm 1 Step 4 | $M^2N^2\,\lvert\mathcal{D}\rvert$ |
| Algorithm 1 Step 6 | $M^2N^2\,\lvert\mathcal{D}\rvert + MN\left(\lvert\mathcal{D}\rvert^2 + \lvert\mathcal{D}\rvert\right) + \mathcal{O}\left(\lvert\mathcal{D}\rvert^3\right) + \lvert\mathcal{D}\rvert^2$ |
| Algorithm 1 Step 7 | $M^2N^2\,\lvert\mathcal{D}\rvert$ |
| Proposed scheme: total | $3M^2N^2\,\lvert\mathcal{D}\rvert + MN\left(\lvert\mathcal{D}\rvert^2 + \lvert\mathcal{D}\rvert\right) + \mathcal{O}\left(\lvert\mathcal{D}\rvert^3\right) + \lvert\mathcal{D}\rvert^2$ |
| SBL-based scheme | $2M^2N^2\,\lvert\mathcal{D}\rvert + I_{\mathrm{SBL}}\left(3MN\,\lvert\mathcal{D}\rvert + 3\,\lvert\mathcal{D}\rvert + 2\,\lvert\mathcal{D}\rvert^2 + \mathcal{O}\left(\lvert\mathcal{D}\rvert^3\right)\right)$ |

(see Table I) is dominated by $\mathcal{O}(|\mathcal{D}|M^2N^2)$ operations for $\tilde{\mathbf{\Phi}}_{\mathrm{d,spr}}$ matrix construction and IOR reconstruction, and $\mathcal{O}\big(M^2N^2|\mathcal{D}| + MN|\mathcal{D}|^2 + |\mathcal{D}|^3\big)$ for path gain vector estimation. Table I also shows the overall computational complexity of the SBL-based benchmark, where $I_{\mathrm{SBL}}$ denotes the number of inner SBL iterations for channel coefficient estimation within each iteration of the overall estimation procedure.

## IV. Results and Discussions

In this section, we evaluate the performance of the proposed IOR estimation scheme. Sinc and Gaussian pulse shaping filters are considered for their contrasting characteristics. The sinc pulse, while superior in detection due to nulls at DD sampling points, is less suitable for IOR estimation because of significant sidelobe energy. In contrast, the Gaussian pulse, being compact in the DD domain, enables improved IOR estimation but suffers from detection loss due to the absence of such nulls [3]. The simulation parameters are as follows: $M = 31$, $N = 37$, $\nu_p = 15$ kHz, $\tau_p = 1/\nu_p = 66.67$ $\mu$s. The channel has $P = 6$ paths with power delay profile (PDP) following the Veh-A model [2], with the total power of path coefficients normalized to one. The Doppler shifts are chosen as $\nu_i = \nu_{\max}\cos(\theta_i)$ for $1 \le i \le P$, where $\nu_{\max} = 815$ Hz and $\theta_i$ are i.i.d. uniform over $[0, 2\pi)$. For the spread-pilot chirp, the slope parameter is $q = 5$, and $\eta$ in **Algorithm 1** is taken to be $10^{-3}$. Let $E_{\mathrm{p}}$ and $E_{\mathrm{d}}$ denote the pilot and data energies in a frame, respectively. The pilot-to-data power ratio (PDR) is defined as $E_{\mathrm{p}}/E_{\mathrm{d}}$. The data signal-to-noise power ratio (DSNR) is defined as $\mathrm{DSNR} = E_{\mathrm{d}}/(MN\sigma^2)$, where $\sigma^2$ is the noise variance. The NMSE is defined as $\mathbb{E}(\|\mathbf{H}-\hat{\mathbf{H}}\|_F^2/\|\mathbf{H}\|_F^2)$, where $\|\cdot\|_F$ is the Frobenius norm. Detection is carried out using the least-square minimum residual based interference cancellation (LSMR-IC) algorithm [11]. The LSMR-IC detector has a computational complexity of $\mathcal{O}(M^2N^2)$. In comparison, the dominant complexity of the proposed IOR estimation is $\mathcal{O}(M^2N^2|\mathcal{D}|)$, which scales cubically with the frame size $MN$ since $|\mathcal{D}|$ scales with $MN$. All simulations are performed in MATLAB R2023b on a 12th Gen Intel(R) Core(TM) i7-12700 CPU.

Figure 1(a) illustrates the NMSE performance of the proposed estimation scheme as a function of DSNR for different values of $s_\tau = s_\nu$. The case $s_\tau = s_\nu = 1$ yields the poorest performance, since the DD dictionary collapses to integer DDs, which is inadequate for accurately approximating fractional DD components. This degradation is more pronounced for the sinc pulse due to its non-compact support in the DD domain compared to the Gaussian pulse. When $s_\tau = s_\nu = 2$, the NMSE improves significantly for both pulse shapes. The improvement is particularly pronounced for the sinc pulse, as the DD dictionary now contains fractional grid points that provide a much better approximation of the underlying physical channel parameters. Further increasing $s_\tau$ yields only marginal gains. Thus, in practice, $s_\tau$ and $s_\nu$ can be selected to trade off the approximation accuracy against the resulting dictionary size and complexity. The results in Fig. 1(a) show that $s_\tau = s_\nu = 2$ provides a good balance for the considered system. It is also observed that the NMSE with sinc filter is better than that with Gaussian filter. This is because Gaussian filter, being poorer in detection, affects the cancellation of data interference and degrades the IOR estimation accuracy.

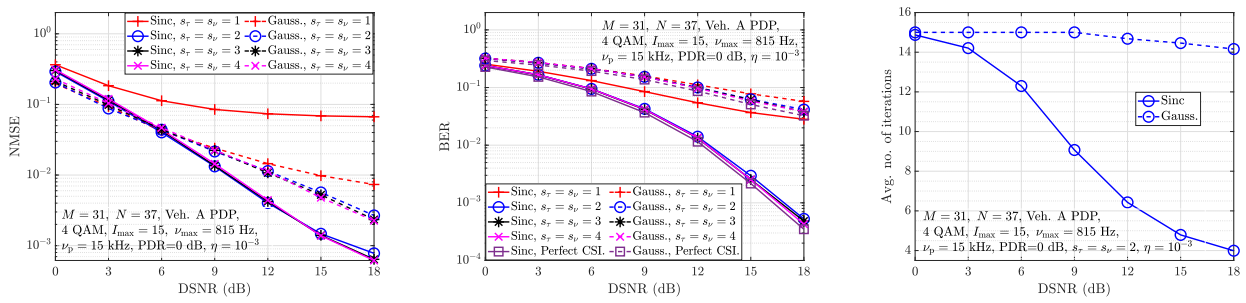
Figure 1(b) presents the BER performance of the proposed scheme for different values of $s_\tau = s_\nu$. When $s_\tau = s_\nu = 1$, the detection performs the poorest, which is consistent with the corresponding NMSE behavior. This is a direct consequence of the integer DD dictionary being inadequate for representing fractional DD components. With $s_\tau = s_\nu = 2$, the BER performance improves remarkably and performs within $0.5$ dB of the perfect CSI benchmark. Further increasing $s_\tau$ does not yield any noticeable improvement in BER, as the minor variations observed in the NMSE curves in Fig. 1(a) do not translate into meaningful detection performance gains.

Figure 1(c) shows the convergence behavior of the proposed estimator in terms of average number of iterations required by **Algorithm 1** for convergence. At low DSNRs, the algorithm requires more iterations to converge due to high noise power and poor detection performance. As DSNR increases, convergence becomes faster for the sinc filter owing to improved detection and therefore effective pilot-data interference cancellation over iterations. For the Gaussian filter, however, convergence remains slower even at higher DSNRs because of the relatively poor detection performance with Gaussian filter, making the pilot-data interference cancellation not so effective.

In Fig. 2(a), we present a BER performance comparison between the proposed estimation scheme, the on-grid SBL scheme[7] [6], and the model-free estimation scheme [10]. The performance is evaluated for two channel models, namely EVA and TDL-A, with $\nu_{\max} = 2500$ Hz. It can be observed that the model-free estimation scheme exhibits a high error floor relative to the perfect CSI benchmark for both channel models. In contrast, the proposed scheme and the SBL-based scheme achieve similar BER performance, remaining within approximately $0.5$ dB of the perfect CSI performance.

Figure 2(b) compares the computational complexities of the proposed scheme and the SBL-based scheme with $I_{\mathrm{SBL}} = 200$ in terms of the number of complex-valued multiplications, given in Table I, as a function of the frame size $MN$ with $M = N$. It can be observed that the proposed scheme requires fewer complex-valued multiplications than the SBL-based scheme over the considered frame sizes. This reduction is primarily due to the closed-form estimation of the channel coefficients in the proposed scheme, whereas SBL requires

[7]The following parameters are used for the SBL simulations: $c = d = 10^{-4}$, $\rho = 10^{-2}$, $\epsilon = 10^{-4}$, $\beta_0 = \frac{100MN}{\|\mathbf{y}_{\mathrm{spr}}\|^2}$, $I_{\mathrm{SBL}} = 200$, with $\mathcal{D}$ chosen as the set of on-grid points. The path gains are initially estimated by assuming that the received signal contains only the pilot component. The detected data based on this estimate is then used to compensate the received signal, followed by re-estimation of the channel. This procedure is repeated until a maximum number of iterations $I_{\max}$ is reached or the change in successive path-gain estimates falls below a threshold $\eta$.




(a) NMSE vs DSNR across various $s_\tau = s_\nu$. (b) BER vs DSNR across various $s_\tau = s_\nu$. (c) Avg. no. of iterations required for convergence of **Algorithm 1** as a function of DSNR.

Fig. 1: NMSE and BER performances and convergence analysis of the proposed IOR estimation scheme.

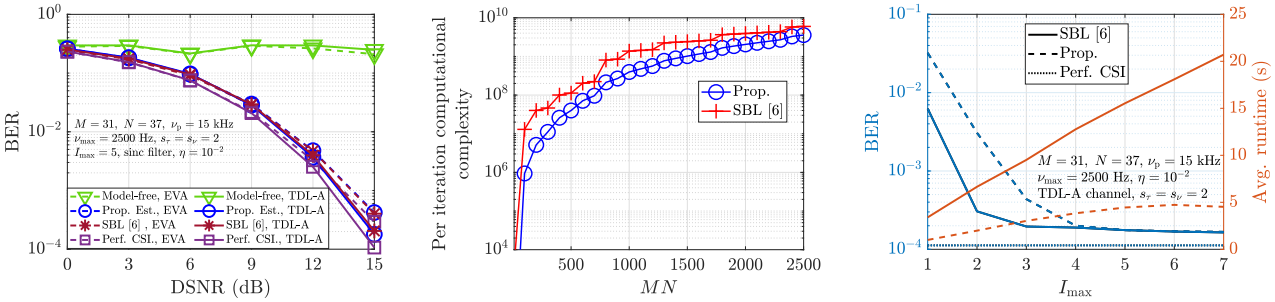



Fig. 2: (a) BER vs DSNR for various estimation algorithms. (b) Complexity comparison. (c) BER & average runtime vs $I_{\max}$.

iterative path gain estimation. In Fig. 2(c), we plot both BER and average simulation runtime[8] (on the left and right y-axes, respectively) as a function of maximum number of iterations $I_{\max}$. In terms of BER performance, at smaller values of $I_{\max}$, the SBL scheme performs better owing to its iterative path-gain estimation, which enables it to adapt to pilot-data interference. In contrast, the proposed scheme relies on a closed-form estimate that does not explicitly account for data interference initially, leading to a performance gap for small $I_{\max}$. However, as $I_{\max}$ increases, both schemes converge to similar performance, with near-identical BER observed for $I_{\max} \geq 4$. It can be observed that the proposed scheme achieves this convergence at a significantly lower runtime compared to the SBL-based scheme. In addition to its higher computational complexity, the sequential nature of the iterative path-gain estimation in SBL further increases its runtime. In contrast, the proposed scheme estimates the path gains using the closed-form expression in (14), which requires only a single matrix inversion. Consequently, the runtime of the SBL-based scheme increases with $I_{\max}$, whereas that of the proposed scheme increases more slowly and eventually floors as **Algorithm 1** satisfies the stopping criterion before reaching $I_{\max}$ iterations.

## V. Conclusions

We proposed a DD dictionary based IOR estimation scheme for Zak-OTFS. The proposed approach approximated the physical channel using a predefined DD dictionary and estimated the corresponding path gains through a closed-form linear solution, thereby achieving a significant complexity advantage compared to SBL based baseline scheme. The proposed scheme performed close to perfect CSI benchmark within 0.5 dB for different channel models. Statistically optimized dictionary designs can be explored as future work.

[8]The average runtime is obtained over 2500 independent simulation realizations, excluding the preprocessing and offline dictionary construction times.

## References


[1] S. K. Mohammed et al., "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, Nov. 2022.

[2] S. K. Mohammed et al., "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.

[3] A. Das, F. Jesbin, and A. Chockalingam, "A Gaussian-sinc pulse shaping filter for Zak-OTFS," *IEEE Trans. Veh. Tech.*, vol. 75, no. 3, pp. 4307-4322, Mar. 2026.

[4] W. Yuan et al., "Data-aided channel estimation for OTFS systems with a superimposed pilot and data transmission Scheme," *IEEE Wireless Commun. Lett.*, vol. 10, no. 9, pp. 1954-1958, Sep. 2021.

[5] I. A. Khan and S. K. Mohammed, "A low-complexity OTFS channel estimation method for fractional delay-Doppler scenarios," *IEEE Wireless Commun. Lett.*, vol. 12, no. 9, pp. 1484-1488, Sep. 2023.

[6] Z. Wei et al., "Off-grid channel estimation with sparse Bayesian learning for OTFS systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7407-7426, Sep. 2022.

[7] A. Abdelbar, H. Hegazy, I. Al-Nahhal, and O. A. Dobre, "Efficient and reliable index modulation OTFS framework for THz-ISAC systems," *IEEE Trans. Wireless Commun.*, vol. 25, pp. 18276-18292, 2026.

[8] H. Hegazy, A. Abdelbar, I. Al-Nahhal, and O. A. Dobre, "Low complexity super-resolution OTFS-assisted ISAC framework for THz communication," *IEEE Trans. Veh. Tech.*, vol. 75, no. 1, pp. 564-578, Jan. 2026.

[9] Y. Ding and B. D. Rao, "Dictionary learning-based sparse channel representation and estimation for FDD massive MIMO systems," *IEEE Trans Wireless Commun.*, vol. 17, no. 8, pp. 5437-5451, Aug. 2018.

[10] M. Ubadah, S. K. Mohammed, R. Hadani, S. Kons, A. Chockalingam, and R. Calderbank, "Zak-OTFS to integrate sensing the I/O relation and data communication," online: arXiv:2404.04182, Jun. 2025.

[11] H. Qu, G. Liu, L. Zhang, S. Wen, and M. A. Imran, "Low-complexity symbol detection and interference cancellation for OTFS system," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1524-1537, Mar. 2021.