Zak-OTFS Implementation via Time and Frequency Windowing

Swaroop Gopalam*, Iain B. Collings*, Stephen V. Hanly*, Hazer Inaltekin*, Sibi Raj B. Pillai†, Philip Whiting*,
*School of Engineering, Macquarie University, Sydney, Australia.
{swaroop.gopalam, iain.collings, stephen.hanly, hazer.inaltekin, philip.whiting}@mq.edu.au
†Department of Electrical Engineering, IIT Bombay, Mumbai, India.
bsraj@ee.iitb.ac.in

***Abstract*—This paper presents an efficient practical Zak-OTFS modulation implementation using time and frequency windowing methods. We present two general classes of delay-Doppler (DD) twisted convolution (TC) filters (Type-1 and Type-2), and show that they can be realized by time and frequency windowing functions. We then propose practical methods to generate time domain Zak-OTFS signals, for actual transmission, using the windowing functions. For Type-1, the signals are generated using an interpolation filter. For Type-2, they are generated using a form of precoded OFDM. We show that this allows a wide variety of pulse shapes to be implemented in practice for Zak-OTFS modulation. This was not previously possible. We also show that the Type-2 signals are more spectrally efficient than their Type-1 counterparts. Finally, we compare the channel predictability of the two implementations.**

***Index Terms*—OTFS, delay-Doppler domain, Zak-OTFS, Zak Transfrom, Twisted Convolution Filters, Time-Frequency Windowing, Channel Predictability.**

I. Introduction

Orthogonal Time Frequency Space (OTFS) modulation was proposed for doubly dispersive channels to overcome the inter-carrier interference (ICI) problem suffered by OFDM in high Doppler scenarios [1], [2]. The core idea of OTFS is to modulate information symbols in the delay-Doppler (DD) domain, by using pulses which are nearly localized in the DD domain [1]–[3]. The sparsity of the DD domain channel is exploited to perform equalization efficiently [3]–[6]. However, channel estimation and equalization are challenging due to the complicated input-output relationship of OTFS [3], [4], [7].

The original proposal of OTFS, referred to as multi-carrier OTFS (MC-OTFS), was a two stage implementation [2], where the OTFS symbol was created via a sequence of coded OFDM symbols. This provided a practical implementation, however it resulted in a complicated input-output relationship in which each information symbol undergoes a different transformation due to the channel, as was shown in [7], [8]. An alternative waveform proposal was presented in [9], called Orthogonal Delay Doppler Modulation (ODDM). It was based on a Nyquist interpolation filter to design pulses for multi-carrier modulation, however in the presence of fractional delays and Doppler shifts, the ODDM pulses lose their orthogonality.

Recently in [7], [8], a new OTFS framework was presented, called Zak-OTFS, based on the Zak Transform. It introduced the concept of a twisted convolution (TC) filter to create an input-output relationship where each symbol undergoes the same transformation through the channel, which leads to more accurate channel estimation and an ability to operate in non-sparse channels. It was shown that the Zak-OTFS framework has the potential to offer superior bit error rate (BER) performance compared to MC-OTFS in high Doppler spread scenarios. The Zak-OTFS transmission method was presented entirely in the DD domain, and no method was provided for implementation in the time domain. In particular, it was not shown how to practically realise the time domain Zak-OTFS signals, given by the TC filtering process.

A practical approach for designing pulses for DD communication, also based on the Zak transform, was considered in [10]. It was different to Zak-OTFS modulation, as it extended the approach in [9] where the focus was on a matched filtering based receiver and on the design of orthogonal pulses. Time domain windowing was introduced to pulse shape the ODDM waveform, however the waveform derivation relied on approximations, to arrive at a practically realizable signal.

In this paper, we present methods for implementing Zak-OTFS modulation in practice. We use complex valued time and frequency windowing methods to derive time domain implementations, which were not possible with the previous DD domain formulation. We first present two general classes of TC filters (Type-1 and Type-2), and show that they can be realized by time and frequency windowing functions. We then propose practical methods to generate the time domain Zak-OTFS signals for each class, for actual transmission, using the windowing functions. For Type-1, the signals are generated using an interpolation filter. For Type-2, they are generated using a generalization of Delay Doppler OFDM (DD-OFDM) [11], with different precoding, and new frequency and time windowing steps.

We point out that the TC filters used in [8] to demonstrate BER superiority of Zak-OTFS, belong to the Type-1 class. We show that those specific Type-1 filters cannot be realised in practice, due to the resulting Zak-OTFS symbol having an infinite time response. We show that an equivalent Type-2 Zak-OTFS signal (which has the same pulse shape in the DD domain) is time-limited and can be practically realized using our method. We propose a modification to the Type-1 implementation by allowing roll-off of frequency window, to make it practical.

This work was supported by the Australian Research Council's Discovery Project Funding Scheme (Project number DP230102252).

We compare the performance of the two implementations in the time and frequency domain, using rectangular and root raised cosine (RRC) windows. We show that the DD-OFDM based Type-2 implementation has a better spectral efficiency, *i.e.* requires less time and bandwidth resources, compared to the Type-1 implementation. It also has significantly better roll-off of side-lobes in the frequency domain with RRC windows.

The rest of the paper is organized as follows. Section II presents the system model. It introduces the Zak-OTFS framework of [7], [8], and highlights the challenges in implementing Zak-OTFS modulation in practice, which includes implementation of TC filters. In Section III, we present our method for realizing Type-1 and Type-2 TC filters using time and frequency windowing. In Section IV, we present the time domain formulation of the Zak-OTFS basis functions, corresponding to Type-1 and Type-2 transmit TC filters. In Section V, we present our proposed implementations of Zak-OTFS modulation. In Section V-A, we present the Zak-OTFS implementation corresponding to the Type-1 TC filters, using an interpolation filter. In Section V-B, we present the Zak-OTFS implementation corresponding to the Type-2 TC filters, using the DD-OFDM framework mentioned earlier. In Section VI, we compare the two implementations in terms of DD domain pulse shape and effective DD domain channel spread. We show that the two implementations have identical performance in the DD domain. In Section VII, we compare the two implementations in terms of time and frequency domain resources. We show that Type-2 Zak-OTFS (based on DD-OFDM) has practical advantages over Type-1 for windows with finite support. Type-2 Zak-OTFS also has better spectral efficiency for practical implementation. In Section VIII, we compare the channel predictability of the two approaches. In Section IX, we present our conclusions.

II. System Model

This section provides an overview of the Zak-OTFS modulation framework introduced in [7], [8] and also the challenges in practical implementation of Zak-OTFS.

A. Discrete DD Domain Zak-OTFS Signal

Zak-OTFS modulates data in the delay-Doppler domain, where both delay and Doppler are discretized on a $M \times N$ grid. The data symbols $\hat{x}[l,k]$ are placed on the $M \times N$ DD data grid, where $0 \leq l \leq M-1$ is the delay index and $0 \leq k \leq N-1$ is the Doppler index. The conversion to DD domain from the time domain is done through the Zak transform as defined below.

Definition 1. *For $T > 0$, the Zak transform of a continuous time signal $s(t)$ is defined as*

$$\mathcal{Z}_s(\tau,\nu) := \sum_{n\in\mathbb{Z}} s(\tau+nT)e^{-j2\pi\nu nT}. \tag{1}$$

$\mathcal{Z}_s(\tau,\nu)$ is the delay-Doppler domain representation of the time domain signal $s(t)$.

A DD domain signal is quasi-periodic due to the quasi-periodicity of the Zak transform, *i.e.*

$$\mathcal{Z}_s(\tau+n'T,\nu+m'\Delta f) = e^{j2\pi\nu n'T}\mathcal{Z}_s(\tau,\nu) \tag{2}$$

for each $m', n' \in \mathbb{Z}$ and $\Delta f := \frac{1}{T}$. Note that the DD domain signal is periodic for shifts of Δf along the Doppler axis, whereas there is an extra multiplicative phase term associated with shifts of T along the delay axis.

The full Zak-OTFS discrete DD domain signal $x_{\text{dd}}[l',k']$ is therefore constructed by performing a quasi periodic encoding of the data symbols onto the infinite DD grid, by extending the $M \times N$ data grid as follows:

$$x_{\text{dd}}[l+nM,k+mN] := \hat{x}[l,k]e^{j2\pi\frac{nk}{N}} \tag{3}$$

for $(m,n) \in \mathbb{Z}\times\mathbb{Z}$ and $(l,k) \in \{0,\ldots,M-1\}\times\{0,\ldots,N-1\}$.

B. Twisted Convolution Filtering and the Transmitter

The discrete signal $x_{\text{dd}}[l',k']$ is converted to an analog DD domain signal $x_{\text{dd}}(\tau,\nu)$ on the grid, by modulating the discrete values using DD domain impulses, as follows:

$$x_{\text{dd}}(\tau,\nu) := \sum_{(l',k')\in\mathbb{Z}\times\mathbb{Z}} x_{\text{dd}}[l',k']\,\delta\left(\tau-\frac{l'T}{M}\right)\delta\left(\nu-\frac{k'\Delta f}{N}\right). \tag{4}$$

Hence, the impulse at location $(l'\frac{T}{M}, k'\frac{\Delta f}{N})$ modulates the symbol $x_{\text{dd}}[l',k']$ for each $(l',k') \in \mathbb{Z}\times\mathbb{Z}$.

The time domain representation of this analog DD domain signal $x_{\text{dd}}(\tau,\nu)$ is neither band-limited nor time-limited and hence is not practical. The crucial step in Zak-OTFS is to generate a practically realizable time-limited and almost band-limited signal from $x_{\text{dd}}(\tau,\nu)$, by means of a DD domain twisted convolution (TC) filter [7], where the twisted convolution operation of two DD domain functions is defined as follows:

Definition 2. Twisted convolution *of two DD functions $a(\tau,\nu)$, $b(\tau,\nu)$ is defined as*

$$a *_\sigma b(\tau,\nu) := \iint a(\tau',\nu')b(\tau-\tau',\nu-\nu')e^{j2\pi\nu'(\tau-\tau')}d\tau'd\nu'.$$

Note that twisted convolution is a two dimensional convolution, with a multiplicative phase term that couples the two dimensions.

Let $g(\tau,\nu)$ denote a generic TC filter in the DD domain. Now let $g^{\text{Tx}}(\tau,\nu)$ specifically denote the transmit filter of Zak-OTFS. The transmitted DD domain signal in Zak-OTFS is therefore given by $x_{\text{dd}}^{g^{\text{Tx}}}(\tau,\nu) := g^{\text{Tx}} *_\sigma x_{\text{dd}}(\tau,\nu)$.

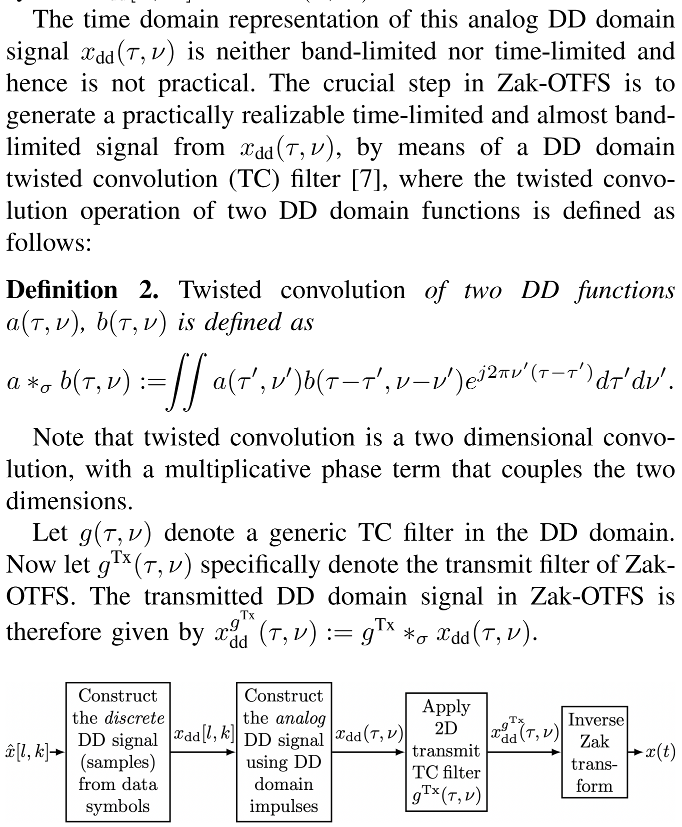

$\hat{x}[l,k]$
Construct the *discrete* DD signal (samples) from data symbols
$x_{\text{dd}}[l,k]$
Construct the *analog* DD signal using DD domain impulses
$x_{\text{dd}}(\tau,\nu)$
Apply 2D transmit TC filter $g^{\text{Tx}}(\tau,\nu)$
$x_{\text{dd}}^{g^{\text{Tx}}}(\tau,\nu)$
Inverse Zak trans-form
$x(t)$

Fig. 1. A generic Zak-OTFS transmitter.

A generic Zak-OTFS transmitter implementation based on these steps is shown in Fig. 1, where the *inverse Zak transform* converts the DD domain signal to time domain, defined as $x(t) := \frac{1}{\Delta f}\int_0^{\Delta f} x_{\text{dd}}^{g^{\text{Tx}}}(t,\nu)d\nu$.

C. Doubly Dispersive Channel

The output of the channel, *i.e.* the received time domain signal, ignoring the additive noise, is given by

$$y(t) = \iint h(\tau,\nu)x(t-\tau)e^{j2\pi\nu(t-\tau)}d\tau d\nu \tag{5}$$

where $h(\tau,\nu)$ is the delay-Doppler response (also known as delay-Doppler spreading function) of the doubly dispersive channel. Taking the Zak-transform of (5), results in a twisted convolution relation in the DD domain as

$$\begin{aligned}\mathcal{Z}_y(\tau,\nu) &= h *_\sigma \mathcal{Z}_x(\tau,\nu) &(6)\\ &= h *_\sigma (g^{\text{Tx}} *_\sigma x_{\text{dd}})(\tau,\nu) &(7)\end{aligned}$$

where (7) follows from the fact that the transmitted DD signal $\mathcal{Z}_x(\tau,\nu)$ is transmit TC filtered version of $x_{\text{dd}}(\tau,\nu)$.

D. Receiver Processing

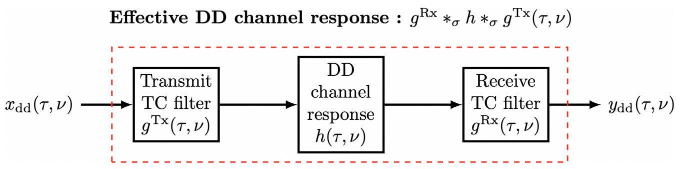

Effective DD channel response : $g^{\text{Rx}} *_\sigma h *_\sigma g^{\text{Tx}}(\tau,\nu)$
$x_{\text{dd}}(\tau,\nu)$
Transmit TC filter $g^{\text{Tx}}(\tau,\nu)$
DD channel response $h(\tau,\nu)$
Receive TC filter $g^{\text{Rx}}(\tau,\nu)$
$y_{\text{dd}}(\tau,\nu)$

Fig. 2. Effective DD Channel Model for Zak-OTFS.

At the receiver, a receive TC filter $g^{\text{Rx}}(\tau,\nu)$ is applied to the received signal $\mathcal{Z}_y(\tau,\nu)$ to obtain the TC filtered received signal $y_{\text{dd}}(\tau,\nu) := g^{\text{Rx}} *_\sigma \mathcal{Z}_y(\tau,\nu)$. Substituting $\mathcal{Z}_y(\tau,\nu)$ from (7), the effective DD domain input-output (I/O) relation of Zak-OTFS is obtained as

$$y_{\text{dd}}(\tau,\nu) = h_{\text{dd}}(\tau,\nu) *_\sigma x_{\text{dd}}(\tau,\nu) \tag{8}$$

where $h_{\text{dd}}(\tau,\nu) := g^{\text{Rx}} *_\sigma h *_\sigma g^{\text{Tx}}(\tau,\nu)$ is the effective DD channel response which is a cascade of three TC filters, 1) transmit TC filter, 2) TC filter corresponding to the channel response and 3) receive TC filter. This effective channel structure is possible due to the associativity property of the twisted convolution operation. An illustration of the effective DD channel for Zak-OTFS is shown in Fig. 2.

Finally, the TC filtered received signal $y_{\text{dd}}(\tau,\nu)$ is sampled on the grid points $(\tau,\nu) = (l\frac{T}{M}, k\frac{\Delta f}{N})$ for $l,k \in \mathbb{Z}\times\mathbb{Z}$ to obtain the discrete DD domain samples. In the discrete DD domain, Zak-OTFS leads to an input-output relation given by

$$y_{\text{dd}}[l,k] = \sum_{l',k'\in\mathbb{Z}} h_{\text{dd}}[l-l',k-k']x_{\text{dd}}[l',k']e^{j2\pi\frac{(k-k')l'}{MN}} \tag{9}$$

where

$$h_{\text{dd}}[l,k] := h_{\text{dd}}(\tau,\nu)\Big|_{\tau=l\frac{T}{M},\nu=k\frac{\Delta f}{N}} \tag{10}$$

is the sampled effective channel response.

The channel response can be estimated using a single DD pilot symbol because all data symbols undergo the same transformation by the channel, as in (9), which is termed *channel predictability* [7], [8]. This is the case when the spread of the effective channel response is small and there is no DD domain aliasing. The spread of the effective channel is not only determined by the delay spread and Doppler spread of the physical channel but also by the choice of the twisted convolution filters.

E. Practical Implementation Challenges of Zak-OTFS

Note that the generic Zak-OTFS formulation presented above is entirely in the DD domain, as shown for the transmitter in Fig. 1 and for the effective channel in Fig. 2. For actual implementation in practice, it is necessary to generate equivalent time domain operations for each of the processing blocks. Moreover, the generic Zak-OTFS transmitter requires the construction of the infinite 2D analog signal $x_{\text{dd}}(\tau,\nu)$, however it is not possible to generate the time domain signal corresponding to $x_{\text{dd}}(\tau,\nu)$, since it is an idealized theoretical construction that is neither time-limited nor band-limited.

The key challenges are therefore: 1) the construction of TC filters in the time domain, which are required at both the transmitter and the receiver, and 2) the construction of the time domain transmitted signal.

In this paper, we address these challenges by proposing two general classes of TC filters, that we call Type-1 and Type-2, that we show can be implemented using frequency and time windowing methods. We also take a direct approach to generating the TC filtered time domain signal for each class of the proposed TC filters. In the following sections, we show that the time domain signal corresponding to our proposed Type-1 TC filters can be generated by a Time Division Multiplexing (TDM) approach using an interpolation filter, and the time domain signal corresponding to our proposed Type-2 TC filters can be generated using an OFDM approach.

III. Twisted Convolution Filters and Windowing

In this section, we present two general classes of TC filters (Type-1 and Type-2) that can be implemented using time and frequency windowing.

We start with some preliminary definitions and results. The following lemma describes the effect of the twisted convolution operation on the time domain input signal.

Lemma 1. *For a time domain signal $s(t)$ and delay-Doppler filter function $g(\tau,\nu)$, the TC filtered signal $s^g(t)$ is given by*

$$s^g(t) = \iint g(\tau',\nu')s(t-\tau')e^{j2\pi\nu'(t-\tau')}d\tau' d\nu' \tag{11}$$

and its Zak transform satisfies

$$\mathcal{Z}_{s^g}(\tau,\nu) = g *_\sigma \mathcal{Z}_s(\tau,\nu) \tag{12}$$

Proof. See Appendix. □

Note that it will be convenient for us to use the following notation in the time domain

$$g *_\sigma s(t) := s^g(t) \tag{13}$$

to indicate that the relationship is a twisted convolution in the DD domain as in Lemma 1. Hence, $g *_\sigma s(t)$ is the time domain signal which has the DD domain representation $g *_\sigma \mathcal{Z}_s(\tau,\nu)$.

Let $S^g(f) := \int g *_\sigma s(t)e^{-j2\pi ft}dt$ denote the Fourier transform of the TC filtered signal $s^g(t)$. Taking the Fourier transform of both sides of (11), we obtain

$$S^g(f) = \iint g(\tau', \nu')S(f-\nu')e^{-j2\pi f\tau'}d\nu' d\tau' \qquad (14)$$

where $S(f)$ is the Fourier transform of $s(t)$.

A. Frequency and Time Windowing

The following two lemmas present TC filtering formulations that are equivalent to frequency domain windowing and time domain windowing respectively. These windowing functions are a vital component of our proposed practical time domain Zak-OTFS implementation.

Lemma 2. *Consider a TC filter $g(\tau, \nu)$ with a DD response that only has a spread along the delay axis, given by $\alpha(\tau)$,* i.e.

$$g(\tau, \nu) = \alpha(\tau)\delta(\nu) \qquad (15)$$

Then, for an input time domain signal $s(t)$, the Fourier transform of the TC filtered signal is given by

$$S^g(f) := \int g *_\sigma s(t)e^{-j2\pi ft}dt = A(f)S(f) \qquad (16)$$

where

$$A(f) := \int \alpha(\tau)e^{-j2\pi f\tau}d\tau \qquad (17)$$

is the Fourier transform of $\alpha(\tau)$, and $S(f) := \int s(t)e^{-j2\pi ft}dt$ is the Fourier transform of $s(t)$.

Proof. From (14), we obtain $S^g(f) = A(f)S(f)$. □

Remark 1. *Note that the LHS of* (16) *is the Fourier transform of the TC filtered signal $g *_\sigma s(t)$, where filter $g(\tau, \nu) = \alpha(\tau)\delta(\nu)$, whereas the RHS is the product of the complex valued frequency domain window $A(f)$ with the Fourier transform of $s(t)$. Hence, Lemma 2 shows that applying a frequency domain window, $A(f)$, to a signal is equivalent to twisted convolution of the signal with the TC filter $\alpha(\tau)\delta(\nu)$.*

Lemma 3. *Consider a TC filter $g(\tau, \nu)$ that has a DD response that only has a spread along the Doppler axis, given by $\beta(\nu)$,* i.e.

$$g(\tau, \nu) = \delta(\tau)\beta(\nu) \qquad (18)$$

Then, for an input time domain signal $s(t)$, the TC filtered signal is

$$g *_\sigma s(t) = B(t)s(t), \qquad (19)$$

where

$$B(t) := \int \beta(\nu)e^{j2\pi t\nu}d\nu \qquad (20)$$

is the inverse Fourier transform of $\beta(\nu)$.

Proof. From (11), we obtain $s^g(t) = s(t)B(t)$. □

Remark 2. *Note that the LHS of* (19) *is the TC filtered signal $g *_\sigma s(t)$, where filter $g(\tau, \nu) = \delta(\tau)\beta(\nu)$, whereas the RHS is the product of the complex valued time domain window $B(t)$ with the signal $s(t)$. Hence, Lemma 3 shows that applying a time domain window, $B(t)$, to a signal, is equivalent to twisted convolution of the signal with the TC filter $\delta(\tau)\beta(\nu)$.*

B. Type-1 and Type-2 TC Filters

We now propose a general class of windowing functions that can be used to realise general TC filters that have a spread along both delay and Doppler axes. We present two classes of TC filters: Type-1 filters are a product of a delay spreading function, $\alpha(\tau)$, and a Doppler spreading function, $\beta(\nu)$, as follows:

$$g_1(\tau, \nu) = \alpha(\tau)\beta(\nu) \qquad (21)$$

and the Type-2 filters have an extra multiplicative sinusoidal component as follows:

$$g_2(\tau, \nu) = \alpha(\tau)\beta(\nu)e^{j2\pi\tau\nu} \qquad (22)$$

In the following sections we will present efficient practical implementations of Zak-OTFS modulation corresponding to the Type-1 and Type-2 filters. We also point out that the proposed TC filters in [8] belong to the general class of Type-1 filters.

We now present our result on realizing Type-1 and Type-2 filters, where we use FT$(\cdot)$ and IFT$(\cdot)$ to represent the Fourier transform and inverse Fourier transform respectively.

Theorem 1.

- *For an input signal $s(t)$ and a Type-1 TC filter with response $g_1(\tau, \nu) = \alpha(\tau)\beta(\nu)$, the TC filtered time domain signal is given by*

$$g_1 *_\sigma s(t) = \alpha(\tau)\delta(\nu) *_\sigma \delta(\tau)\beta(\nu) *_\sigma s(t) \qquad (23)$$
$$= \int A(f)\left(\int B(t')s(t')e^{-j2\pi t'f}df\right)e^{j2\pi ft}dt \qquad (24)$$
$$= \text{IFT}(A(f)\ \text{FT}\left(B(t)s(t)\right)). \qquad (25)$$

- *For an input signal $s(t)$ and a Type-2 TC filter with response $g_2(\tau, \nu) = \alpha(\tau)\beta(\nu)e^{j2\pi\nu\tau}$, the TC filtered time domain signal is given by*

$$g_2 *_\sigma s(t) = \delta(\tau)\beta(\nu) *_\sigma \alpha(\tau)\delta(\nu) *_\sigma s(t) \qquad (26)$$
$$= B(t)\int\left(A(f)\int s(t')e^{-j2\pi ft'}dt'\right)e^{j2\pi ft}dt \qquad (27)$$
$$= B(t)\text{IFT}\left(A(f)\text{FT}\left(s(t)\right)\right). \qquad (28)$$

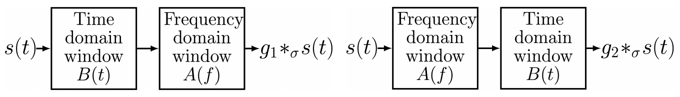
where $B(t) = \int \beta(\nu)e^{j2\pi t\nu}d\nu$ and $A(f) = \int \alpha(\tau)e^{-j2\pi f\tau}d\tau$.

Proof. The proof follows from Lemma 2 and Lemma 3 using the associativity property of twisted convolution. □

Remark 3.

- *From* (25) *in Theorem 1, we see that applying a time domain window $B(t)$ to a signal, followed by a frequency domain window $A(f)$, is equivalent to performing a twisted convolution of the signal with a Type-1 TC filter $g_1(\tau, \nu) = \alpha(\tau)\beta(\nu)$.*
- *From* (28) *in Theorem 1, we see that applying a frequency domain window $A(f)$ to a signal, followed by a time*

domain window $B(t)$, is equivalent to performing a twisted convolution of the signal with a Type-2 TC filter $g_2(\tau,\nu) = \alpha(\tau)\beta(\nu)e^{j2\pi\nu\tau}$.


$s(t)$
Time domain window $B(t)$
Frequency domain window $A(f)$
$g_1 *_\sigma s(t)$
$s(t)$
Frequency domain window $A(f)$
Time domain window $B(t)$
$g_2 *_\sigma s(t)$

(a) Type-1 TC filter $g_1(\tau,\nu)$. (b) Type-2 TC filter $g_2(\tau,\nu)$.

Fig. 3. Practical TF domain operations for implementation of Type-1 and Type-2 TC filters using time-frequency windowing.

Hence, we have shown that Type-1 and Type-2 TC filters can be practically implemented using time and frequency windowing as illustrated in Fig. 3.

IV. Twisted convolution filtered Zak-OTFS Basis functions

In this section, we formulate the time domain basis functions for Zak-OTFS, corresponding to our Type-1 and Type-2 transmit TC filters from the previous section. In Section V, we will propose time domain methods to generate these basis functions.

A. Unfiltered Basis Functions

Let $\tau_l := l\frac{T}{M}$ and $\nu_k := k\frac{\Delta f}{N}$ for each $(l,k) \in \mathbb{Z}\times\mathbb{Z}$. The analog Zak-OTFS symbol $x_{\text{dd}}(\tau,\nu)$ in the DD domain from (4) can be expressed as

$$x_{\text{dd}}(\tau,\nu) = \sum_{l=0}^{M-1}\sum_{k=0}^{N-1} \hat{x}[l,k]\mathcal{Z}_{\phi_{l,k}}(\tau,\nu) \quad (29)$$

where $\phi_{l,k}(t)$ is the data carrying basis function corresponding to the DD index pair (l,k) (and data symbol $\hat{x}[l,k]$), given by

$$\phi_{l,k}(t) := T\sum_{n=-\infty}^{\infty} e^{j2\pi\nu_k nT}\delta(t-\tau_l-nT) \quad (30)$$

and its Zak tranform is the quasi-periodic DD impulse train

$$\mathcal{Z}_{\phi_{l,k}}(\tau,\nu) = \sum_{(m,n)\in\mathbb{Z}\times\mathbb{Z}} e^{j2\pi\nu_k nT}\delta(\tau-\tau_l-nT)\,\delta(\nu-\nu_k-m\Delta f) \quad (31)$$

Note from (30) that the unfiltered basis function $\phi_{l,k}(t)$ is a *pulsone* with ideal impulses, *i.e.* it is a tone modulated impulse train, and is hence neither band-limited nor time-limited. To realize the Zak-OTFS framework, the TC filtered basis functions, which are time-limited and almost band-limited, must be generated as time domain signals.

B. TC Filtered Basis Functions

We now apply Theorem 1 to obtain the TC filtered basis functions, corresponding to Type-1 and Type-2 TC filters.

From Theorem 1, the Zak-OTFS basis function $\phi^{g_1}_{l,k}(t) := g_1 *_\sigma \phi_{l,k}(t)$ corresponding to a Type-1 transmit TC filter $g_1(\tau,\nu) = \alpha(\tau)\beta(\nu)$ is given by

$$\phi^{g_1}_{l,k}(t) = T\sum_{n\in\mathbb{Z}} B(\tau_l+nT)e^{j2\pi\nu_k nT}\alpha(t-\tau_l-nT) \quad (32)$$

whereas the Zak-OTFS basis function $\phi^{g_2}_{l,k}(t) := g_2 *_\sigma \phi_{l,k}(t)$ corresponding to a Type-2 transmit TC filter $g_2(\tau,\nu) = \alpha(\tau)\beta(\nu)e^{j2\pi\nu\tau}$ is given by

$$\phi^{g_2}_{l,k}(t) = TB(t)\sum_{n\in\mathbb{Z}} \alpha(t-\tau_l-nT)e^{j2\pi\nu_k nT} \quad (33)$$

$$= TB(t)\mathcal{Z}_\alpha(t-\tau_l,\nu_k) \quad (34)$$

C. Pulsone Decomposition

Both Type-1 and Type-2 TC filtered basis functions have a pulsone decomposition, which will provide insight into their structure.

From (32), the pulsone decomposition of Type-1 TC filtered basis function $\phi^{g_1}_{l,k}(t)$ is as follows:

$$\phi^{g_1}_{l,k}(t) = T\sum_{\tau\in\{\tau_l+nT\}_{n\in\mathbb{Z}}} \underbrace{B(\tau)e^{j2\pi\nu_k(\tau-\tau_l)}}_{\text{Windowed Tone Sample}}\underbrace{\alpha(t-\tau)}_{\text{Pulse}} \quad (35)$$

which can be interpreted as a train of pulses modulating the windowed tone samples. The pulse $\alpha(t-\tau)$ modulates the sample $B(\tau)e^{j2\pi\nu_k(\tau-\tau_l)}$ for each $n\in\mathbb{Z}$ and $\tau = \tau_l+nT$. This is illustrated in Fig. 4(a), for a rectangular window $B(t)$. The time windowed tone samples (represented by the square shaped markers) are shown above the pulses. Each colored pulse corresponds to the windowed tone sample of the same color.

To show the pulsone decomposition of the Type-2 TC filtered basis function $\phi^{g_2}_{l,k}(t)$, we introduce the notion of a *Dual Zak transform* which acts on the Fourier transform (*i.e.* frequency domain representation) of a signal, and is defined as follows. We define the Dual Zak transform of a function $X(f)$ as

$$\mathcal{D}_X(\tau,\nu) := \sum_{m\in\mathbb{Z}} X(\nu+m\Delta f)e^{j2\pi\tau m\Delta f} \quad (36)$$

The following lemma presents the relationship between the Dual Zak transform and the Zak transform, which we will use for the pulsone decomposition.

Lemma 4. *Let $X(f)$ be the Fourier transform of $x(t)$, then*

$$\mathcal{D}_X(\tau,\nu) = Te^{-j2\pi\nu\tau}\mathcal{Z}_x(\tau,\nu) \quad (37)$$

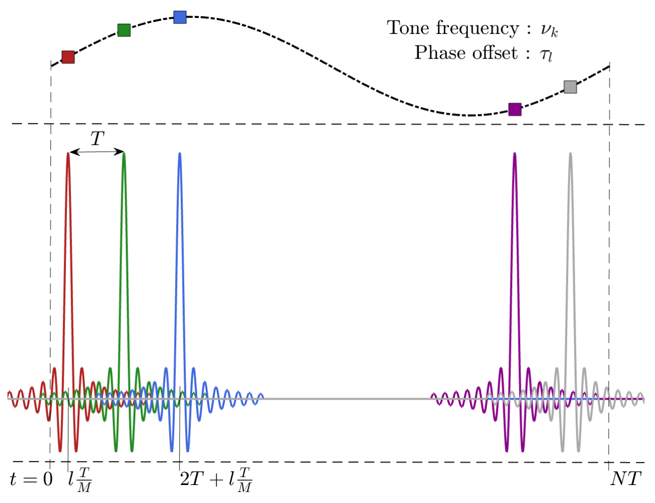
Proof. See Appendix. □

From Lemma 4 and (34), we obtain the pulsone decomposition of Type-2 TC filtered basis function $\phi^{g_2}_{l,k}(t)$ as

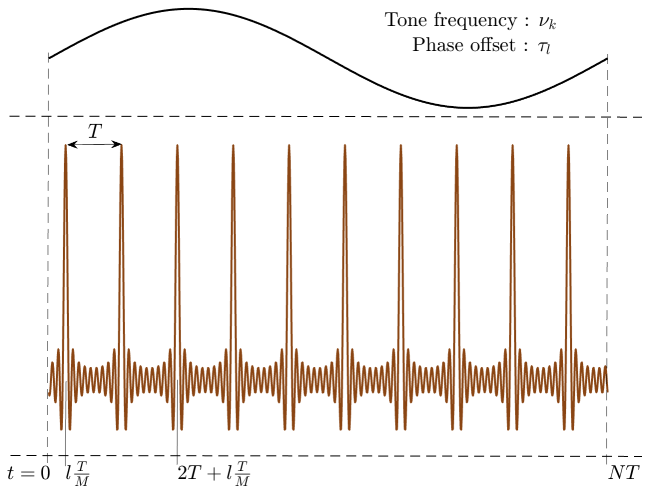
$$\phi^{g_2}_{l,k}(t) = B(t)e^{j2\pi\nu_k(t-\tau_l)}\mathcal{D}_A(t-\tau_l,\nu_k) \quad (38)$$

$$= \underbrace{B(t)e^{j2\pi\nu_k(t-\tau_l)}}_{\text{Windowed Tone}}\underbrace{\sum_{m\in\mathbb{Z}} A(m\Delta f+\nu_k)e^{j2\pi m\Delta f(t-\tau_l)}}_{\text{Periodic Pulse Train}} \quad (39)$$

which is the product of a windowed tone and a pulse train. The windowed tone component is a sinusoid $e^{j2\pi\nu_k(t-\tau_l)}$ multiplied by a time limiting window $B(t)$. The pulse train component can be interpreted as a Discrete Time Fourier Transform (DTFT) type function of the frequency domain window samples. Note that the pulse train component is periodic



(a) Type-1 basis function: Train of pulses modulating the samples of a time windowed tone.



(b) Type-2 basis function: Product of a periodic pulse train and a time windowed tone.

Fig. 4. Time domain illustration of the two types of Zak-OTFS basis functions with rectangular windows.

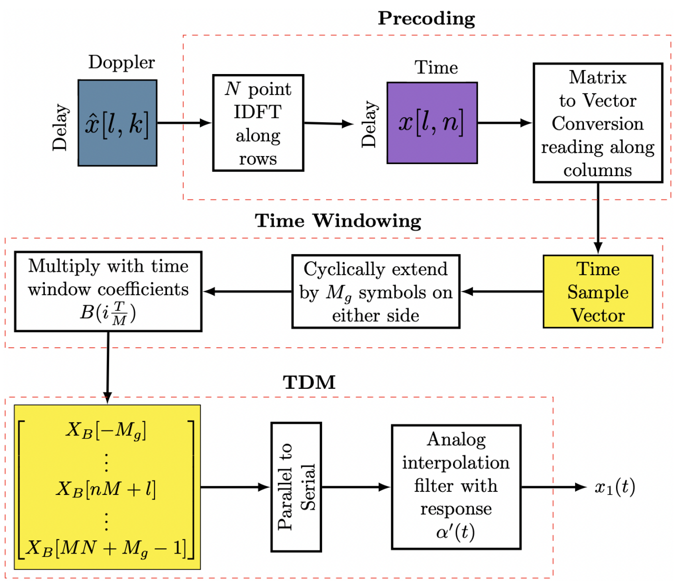

Precoding
Doppler
Delay
$\hat{x}[l,k]$
N point IDFT along rows
Time
Delay
$x[l,n]$
Matrix to Vector Conversion reading along columns
Time Windowing
Multiply with time window coefficients $B(i\frac{T}{M})$
Cyclically extend by M_g symbols on either side
Time Sample Vector
TDM
$X_B[-M_g]$
$X_B[nM+l]$
$X_B[MN+M_g-1]$
Parallel to Serial
Analog interpolation filter with response $\alpha'(t)$
$x_1(t)$

Fig. 5. Block Diagram of Type-1 Zak-OTFS transmitter for a time window $B(t)$ with support $[-M_g\frac{T}{M}, NT+M_g\frac{T}{M})$ where M_g is an integer less than M.

with period T, due to the periodicity of the $e^{j2\pi m\Delta f(t-\tau_l)}$ sinusoid function. The Type-2 basis function is illustrated in Fig. 4(b), for a rectangular window $B(t)$. The time windowed tone (in black) is shown above the periodic pulse train (brown).

V. Proposed Zak-OTFS Implementation

In this section, we present our proposed implementations of Zak-OTFS corresponding to our Type-1 and Type-2 transmit TC filters.

A. *Type-1 TC Filtered Zak-OTFS Implementation Using an Interpolation Filter*

In this section, we present our Zak-OTFS implementation corresponding to our Type-1 TC filters. We show that the Type-1 TC filtered Zak-OTFS waveform can be implemented by a TDM approach using an interpolation filter with filter response $\alpha'(x) := T\alpha(\frac{T}{M}x)$. The overall approach is illustrated in Fig. 5 for the case of a time window $B(t)$ with support $[-M_g\frac{T}{M}, NT+M_g\frac{T}{M})$ for any integer $M_g < M$.

We now present the detailed steps involved in the implementation, for the general case, and show that the Type-1 TC filtered basis functions are synthesized by the proposed method.

1) Precoding: In the precoding step, the delay-time domain symbols $x[l,n]$ for $n = 0,\ldots,N-1$ and $l = 0,\ldots,M-1$ are obtained by taking an N-point Inverse Discrete Fourier Transform (IDFT) of the DD domain data symbols $\hat{x}[l,k]$ along the Doppler dimension k as

$$x[l,n] := \sum_{k=0}^{N-1} \hat{x}[l,k]e^{j2\pi\frac{nk}{N}}. \tag{40}$$

The delay-time domain symbols are periodically extended along the time frame dimension n, for the range $n \in \mathbb{Z} - \{0,\ldots,N-1\}$ as

$$x[l,n] = x[l,(n)_N], \tag{41}$$

where $(n)_N$ represents $n \mod N$.

2) Time Domain Windowing: The delay-time domain symbols are then serialized to obtain time domain samples as

$$X[nM+l] = x[l,n] \tag{42}$$

for each $n \in \mathbb{Z}$ and $l = 0,\ldots,M-1$.

The time domain samples $X[i]$ are multiplied by time domain window coefficients $B[i] := B(i\frac{T}{M})$, to obtain windowed samples $X_B[i] := X[i]B[i]$ for $i \in \mathbb{Z}$. In practice, the time-domain window $B(t)$ has a finite support and hence only a finite number of windowed samples need to be considered.

3) Interpolation Filter: Finally, an analog interpolation filter with response $\alpha'(x) := T\alpha(\frac{T}{M}x)$ is applied to the windowed samples to obtain the signal

$$x_1(t) = \sum_{i\in\mathbb{Z}} X_B[i]\alpha'\left(\frac{t}{T_s} - i\right) \tag{43}$$

$$= T\sum_{i\in\mathbb{Z}} X_B[i]\alpha\left(t - iT_s\right) \tag{44}$$

where $T_s := \frac{T}{M} = \frac{1}{M\Delta f}$ is the sampling interval.

In practice, the interpolation filter can be implemented by pulse shaping digitally followed by digital-to-analog conversion. Digital pulse shaping requires upsampling of the time domain symbols $X_B[i]$; the upsampled symbols are filtered (*i.e.* convolved) with the samples of $\alpha'(t)$ (also obtained at the upsampled rate).

The following Theorem 2 shows that the signal $x_1(t)$ is in fact the Zak-OTFS symbol that modulates the data symbols $\hat{x}[l,k]$ using the Type-1 TC filtered basis functions $\phi^{g_1}_{l,k}(t)$ with TC filter $g_1(\tau,\nu) := \alpha(\tau)\beta(\nu)$.

Theorem 2. *The signal $x_1(t)$ in* (43) *can be expressed as*

$$x_1(t) = \sum_{k=0}^{N-1}\sum_{l=0}^{M-1} \hat{x}[l,k]\phi^{g_1}_{l,k}(t) \tag{45}$$

where $g_1(\tau,\nu) := \alpha(\tau)\beta(\nu)$ is the Type-1 TC filter response and the Type-1 TC filtered Zak-OTFS basis function $\phi^{g_1}_{l,k}(t)$ is given in (32).

Proof. See Appendix. □

B. Type-2 TC Filtered Zak-OTFS Implementation Using DD-OFDM Framework

In this section, we present our Zak-OTFS implementation corresponding to the Type-2 TC filters. Our method to synthesize Type-2 TC filtered Zak-OTFS basis functions (and the corresponding Zak-OTFS signal) is a generalization of our recently proposed DD-OFDM framework [11]. Our new approach here includes a different precoding technique and introduces new time and frequency windowing steps. An important difference is that the tone component in DD-OFDM does not include a phase offset, that appears in the windowed tone of Type-2 Zak-OTFS. This is related to the phase offset between the Zak transform and the Dual Zak transform, as seen in Lemma 4.

The overall approach is illustrated in Fig. 6 for the case of a frequency window $A(f)$ with support $[-N_g\frac{\Delta f}{N}, M\Delta f + N_g\frac{\Delta f}{N})$ for any integer $N_g < N$. We now present the detailed steps involved in the implementation, for the general case.

1) DD-OFDM Precoding: Our scheme starts by modifying the approach in [11] to introduce so called phase offset *twiddle factors* $e^{-j2\pi\frac{lk}{MN}}$ that multiply the data symbols $\hat{x}[l,k]$ for $k = 0,\dots,N-1$ and $l = 0,\dots,M-1$. A Discrete Fourier Transform (DFT) is then taken along the delay dimension to obtain the frequency-Doppler domain symbols as

$$\tilde{x}[m,k] = \sum_{l=0}^{M-1} \hat{x}[l,k]e^{-j2\pi\frac{lk}{MN}}e^{-j2\pi\frac{lm}{M}} \tag{46}$$

for $m = 0,\dots,M-1$.

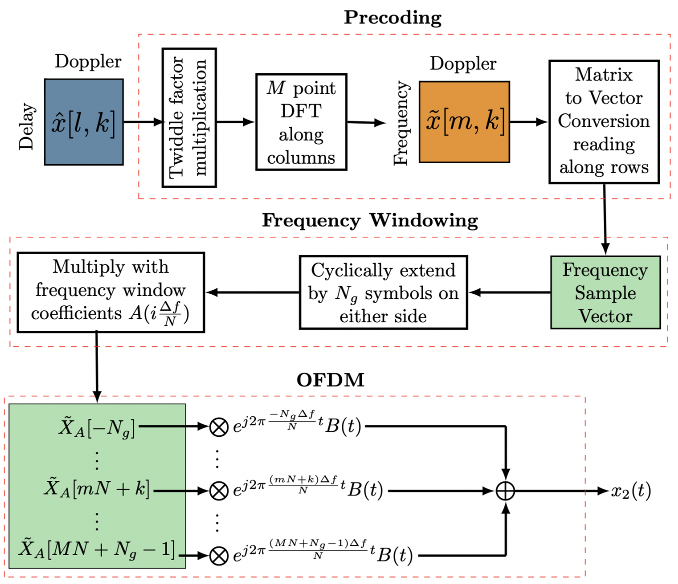

Precoding
Doppler
Delay
$\hat{x}[l,k]$
Twiddle factor multiplication
M point DFT along columns
Doppler
Frequency
$\tilde{x}[m,k]$
Matrix to Vector Conversion reading along rows
Frequency Windowing
Frequency Sample Vector
Cyclically extend by N_g symbols on either side
Multiply with frequency window coefficients $A(i\frac{\Delta f}{N})$
OFDM
$\tilde{X}_A[-N_g]$
$\tilde{X}_A[mN+k]$
$\tilde{X}_A[MN+N_g-1]$
$e^{j2\pi\frac{-N_g\Delta f}{N}t}B(t)$
$e^{j2\pi\frac{(mN+k)\Delta f}{N}t}B(t)$
$e^{j2\pi\frac{(MN+N_g-1)\Delta f}{N}t}B(t)$
$x_2(t)$

Fig. 6. Block Diagram of Type-2 Zak-OTFS transmitter for a frequency window $A(f)$ with support $[-N_g\frac{\Delta f}{N}, M\Delta f + N_g\frac{\Delta f}{N})$ where N_g is an integer less than N.

We extend the frequency-Doppler domain symbols periodically along the frequency frame dimension m, for the range $m \in \mathbb{Z} - \{0,\dots,M-1\}$ as

$$\tilde{x}[m,k] := \tilde{x}[(m)_M, k] \tag{47}$$

where $(m)_M$ represents $m \mod M$.

2) Frequency Domain Windowing: We now serialize the frequency-Doppler domain symbols to obtain the frequency domain samples $\tilde{X}[i]$ for $i \in \mathbb{Z}$ as

$$\tilde{X}[mN+k] = \tilde{x}[m,k] \tag{48}$$

We multiply these frequency domain samples $\tilde{X}[i]$ with corresponding frequency domain window coefficients $A[i] := A(i\frac{\Delta f}{N})$, to obtain the windowed frequency domain samples $\tilde{X}_A[i] := \tilde{X}[i]A[i]$ for $i \in \mathbb{Z}$. In practice, the frequency domain window $A(f)$ has a finite support, and hence only a finite number of samples need to be considered.

3) OFDM Waveform Generation: Next, the filtered frequency samples are placed on OFDM subcarriers with subcarrier spacing $\frac{\Delta f}{N}$ (which we note is much smaller than the Doppler spread, in contrast to standard OFDM). The frequency sample $\tilde{X}_A[i]$ is modulated on subcarrier index i. The resulting OFDM waveform with a time domain window $B(t)$ is

$$x_2(t) = \sum_{i\in Z} \tilde{X}_A[i]B(t)e^{j2\pi i\frac{\Delta f}{N}t} \tag{49}$$

Note that the resulting OFDM subcarrier shape in the frequency domain is a function of the time domain window $B(t)$.

In practice, OFDM pulse shaping is implemented digitally and then digital-to-analog conversion is done to generate the analog signal $x_2(t)$. The OFDM pulse shaping can be implemented via an IDFT on the frequency samples $\tilde{X}_A[i]$ followed by multiplication with samples of the window $B(t)$, to construct the time domain signal samples.

The following theorem shows that the signal $x_2(t)$ is in fact the Zak-OTFS symbol that modulates the data symbols $\hat{x}[l,k]$ using the Type-2 TC filtered basis functions $\phi^{g_2}_{l,k}(t)$ with TC filter $g_2(\tau,\nu) := \alpha(\tau)\beta(\nu)e^{j2\pi\nu\tau}$.

Theorem 3. *The signal $x_2(t)$ in (49) can be expressed as*

$$x_2(t) = \sum_{l=0}^{M-1}\sum_{k=0}^{N-1} \hat{x}[l,k]\phi^{g_2}_{l,k}(t) \tag{50}$$

where $g_2(\tau,\nu) = \alpha(\tau)\beta(\nu)e^{j2\pi\nu\tau}$ is the Type-2 TC filter response and the corresponding TC filtered Zak-OTFS basis function $\phi^{g_2}_{l,k}(t)$ is given in (39).

Proof. See Appendix. □

Hence, we have shown that Zak-OTFS with Type-2 TC filters can be realized using generalized DD-OFDM (generalized from [11], as described above). Thus, a wide variety of pulse shapes for Zak-OTFS can be achieved using our approach. This was not previously possible. And moreover, these pulse shapes can be designed using standard OFDM filter design.

C. Receiver Implementation

In the receiver, the output of the DD channel is filtered with the receive TC filter $g^{\text{Rx}}(\tau,\nu)$, as shown in Fig. 2, and the resultant signal $g^{\text{Rx}} *_\sigma y(t)$ (which is given by $y_{\text{dd}}(\tau,\nu)$ in the DD domain) is then time domain sampled. These samples are then used to generate the receive DD domain samples $y_{\text{dd}}[l,k]$ in (9), using the discrete Zak transform operation. This final step converting from the time domain samples into the DD domain is also done for standard MC-OTFS, as in [6].

The receive TC filter $g^{\text{Rx}}(\tau,\nu)$ can be designed to meet desired performance metrics (for example, the most basic design is to have $g^{\text{Rx}}(\tau,\nu) = \delta(\tau)\delta(\nu)$), and can be implemented in the time domain using Theorem 1.

D. Computational Complexity Comparison

The computational complexities of our two proposed implementations are given in Table I for general window functions, $A(f)$ with support $[-N_g\frac{\Delta f}{N}, M\Delta f + N_g\frac{\Delta f}{N})$ for Type-2, and $B(t)$ with support $[-M_g\frac{T}{M}, NT + M_g\frac{T}{M})$ for Type-1, where N_g and M_g are arbitrary integers less than N and M respectively.

TABLE I
COMPLEXITY COMPARISON

Type-1	$MN\log(N) + (MN + 2M_g)$
Type-2	$MN + NM\log(M) + (MN + 2N_g)$
ODDM [9]	$MN\log(N)$
MC-OTFS [2]	$MN\log(N)$

For Type-1 implementation, the precoding computational complexity is $MN\log(N)$ and the time windowing complexity is $MN + 2M_g$. For Type-2, the precoding complexity is MN for twiddle factor multiplication and $MN\log(M)$ for the DFT operations, and the frequency windowing complexity is $MN+2N_g$. Note that in some cases, the filters have structure that can be exploited to reduce the complexity.

For example, consider rectangular windows $A(f)$ with $N_g = 0$ and $B(t)$ with $M_g = 0$. For this case, the digital implementation of Type-2 only requires a computational complexity of $MN\log(N)$, to obtain the time domain signal samples for digital-to-analog conversion, which is the same as for MC-OTFS and ODDM.

VI. Delay-Doppler Domain Comparison of the Two Approaches

In this section, we consider our two implementation approaches in terms of the delay-Doppler domain characteristics, namely DD domain pulse shape and effective DD channel response. We show that the performance is identical in terms of pulse shaping along delay and Doppler dimensions, and the spread of the channel response. In the next section, we will present a time-frequency domain comparison and show that Type-2 Zak-OTFS is more spectrally efficient in practice.

Throughout this section, we will illustrate our derivations (which are for general window functions) with figures that correspond to the particular case of using rectangular windows $A(f) = \frac{1}{W_{\text{f}}}\mathbb{I}_{[0,W_{\text{f}})}(f)$ and $B(t) = \frac{1}{W_{\text{t}}}\mathbb{I}_{[0,W_{\text{t}})}(t)$ for band limiting and time limiting, where $\mathbb{I}_S(x)$ is the indicator function which equals 1 if $x \in S$ and 0 otherwise, and $W_{\text{f}} = M\Delta f$ and $W_{\text{t}} = NT$. Hence, Figs. 7, 8, and 9 correspond to delay and Doppler filter components

$$\alpha(\tau) = e^{j\pi\frac{\tau}{1/W_{\text{f}}}}\operatorname{sinc}\left(\frac{\tau}{1/W_{\text{f}}}\right) \tag{51}$$

$$\beta(\nu) = e^{-j\pi\frac{\nu}{1/W_{\text{t}}}}\operatorname{sinc}\left(\frac{\nu}{1/W_{\text{t}}}\right) \tag{52}$$

respectively. Note that other windowing functions will be employed in later sections (in particular, root raised cosine with various roll-off factors), and compared with the rectangular functions.

A. TC Filter Response

The magnitude of the TC filter response is identical for the two approaches, and is given by

$$|g_i(\tau,\nu)| = |\alpha(\tau)||\beta(\nu)| \tag{53}$$

for TC filter type $i = 1,2$. The filter response in (53) is illustrated in Fig. 7 for rectangular windows.

B. DD Pulse Shape

The DD domain representation of the Zak-OTFS basis functions for the Type-1 and Type-2 TC filter implementations are given as

$$\mathcal{Z}_{\phi^{g_1}_{l,k}}(\tau,\nu) = \sum_{(m,n)\in\mathbb{Z}\times\mathbb{Z}} \alpha(\tau-\tau_l-nT)\beta(\nu-\nu_k-m\Delta f) e^{-j2\pi\frac{ml}{M}}e^{-j2\pi\nu_k\tau_l}e^{j2\pi\nu(nT+\tau_l)} \tag{54}$$

$$\mathcal{Z}_{\phi^{g_2}_{l,k}}(\tau,\nu) = \sum_{(m,n)\in\mathbb{Z}\times\mathbb{Z}} \alpha(\tau-\tau_l-nT)\beta(\nu-\nu_k-m\Delta f) e^{j2\pi\frac{kn}{N}}e^{j2\pi\tau\nu}e^{-j2\pi\tau(m\Delta f+\nu_k)}. \tag{55}$$

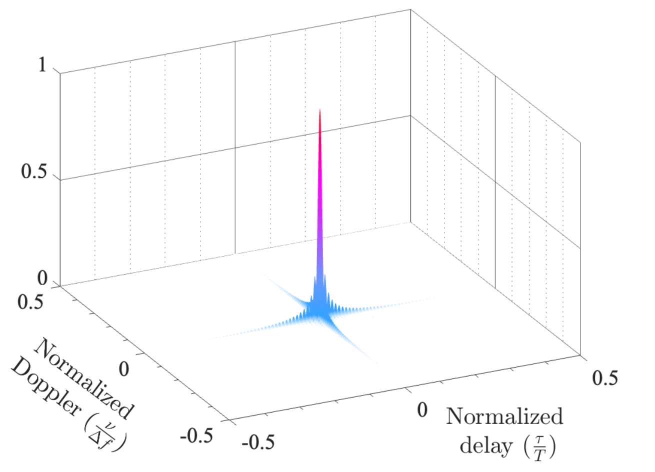


Fig. 7. $|g_i(\tau, \nu)|$: Magnitude of the filter response with rectangular windows.

We note from (54) and (55) that the DD domain shape of the basis functions in the two Zak-OTFS implementations only differ in the phase terms in the summation. The magnitude of DD domain representation $|\mathcal{Z}_{\phi^{g_i}_{l,k}}(\tau, \nu)|$ is shown in Fig. 8 for rectangular windows.

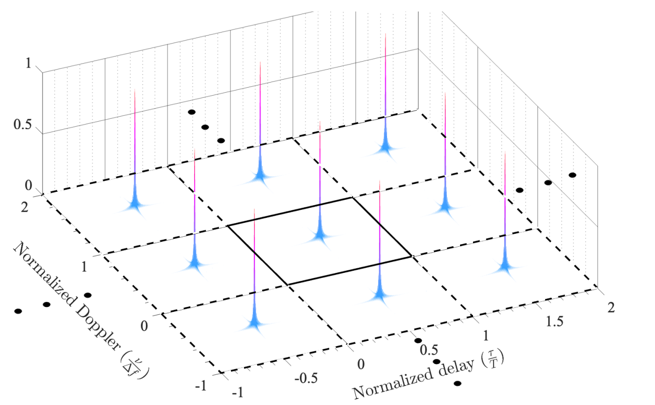


Fig. 8. $|\mathcal{Z}_{\phi^{g_i}_{l,k}}(\tau, \nu)|$: TC filtered basis function in the DD domain for $(l, k) = \left(\frac{M}{2}, \frac{N}{2}\right)$.

In either approach, the TC filtered basis function in the DD domain is a quasi periodic two dimensional pulse train, as shown in Fig. 8. The magnitude of the pulse at location (τ_0, ν_0), where $\tau_0 = (nM + l)\frac{T}{M}$ and $\nu_0 = (mN + k)\frac{\Delta f}{N}$, is

$$|\alpha(\tau - \tau_0)\beta(\nu - \nu_0)| \tag{56}$$

for $(m, n) \in \mathbb{Z} \times \mathbb{Z}$, $(l, k) \in \{0, \ldots, M-1\} \times \{0, \ldots, N-1\}$. Hence, the shape of each pulse is determined by the filter response. Furthermore, the shape $|\alpha(\tau - \tau_0)|$ along the delay axis is determined by the frequency domain window through (17) and the shape along the Doppler axis is similarly determined by the time domain window through (20).

C. Effective Channel Response

In this section, we consider the received signal

$$y_{\text{dd}}(\tau, \nu) = h *_\sigma g^{\text{Tx}} *_\sigma x_{\text{dd}}(\tau, \nu) \tag{57}$$

which is obtained after sending the transmit TC filtered Zak-OTFS symbol through the channel, with a basic receive TC filter $g^{\text{Rx}}(\tau, \nu) = \delta(\tau)\delta(\nu)$ that corresponds to not windowing the signal.

For this setup, the effective channel response is given by

$$h^{g^{\text{Tx}}}_{\text{dd}}(\tau, \nu) := h *_\sigma g^{\text{Tx}}(\tau, \nu) \tag{58}$$

We consider a sparse path channel with P paths labelled $p = 0, \ldots, P-1$, where each path p is characterized by a gain h_p, a delay τ_p and a Doppler shift ν_p. We obtain the effective DD channel response for the two types of transmit TC filters (which correspond to the two Zak-OTFS implementations) as

$$h^{g_1}_{\text{dd}}(\tau, \nu) = \sum_{p=0}^{P-1} h_p \alpha(\tau - \tau_p)\beta(\nu - \nu_p)e^{j2\pi\nu_p(\tau - \tau_p)} \tag{59}$$

$$h^{g_2}_{\text{dd}}(\tau, \nu) = \sum_{p=0}^{P-1} h_p \alpha(\tau - \tau_p)\beta(\nu - \nu_p)e^{j2\pi\nu(\tau - \tau_p)} \tag{60}$$

Note that the response of an individual path only differs in the sinusoidal phase term in the two approaches. In both approaches, for an individual path, the shape of the effective channel response along the Doppler axis is given by the filter response $\beta(\nu - \nu_p)$, *i.e.* the DFT of time domain window centered at Doppler shift ν_p. Similarly, the shape along the delay axis is given by $\alpha(\tau - \tau_p)$, *i.e.* the IDFT of the frequency domain window centered at delay τ_p. Hence, we can conclude that both approaches have identical DD channel spread and DD pulse shaping.

For illustration, consider a channel with four paths with path gains $[1, 1/\sqrt{2}, 1, 1/\sqrt{2}]$, delays $[0, \frac{1}{3}, \frac{2}{3}, 1] \times \tau_{\max}$ and Doppler shifts $[-1, -\frac{1}{2}, \frac{1}{2}, 1] \times \nu_{\max}$, where $\nu_{\max} = 20\frac{\Delta f}{N}$ and $\tau_{\max} = 60\frac{T}{M}$. Fig. 9 shows the magnitude of the effective channel response, which is a superposition of four pulses, each corresponding to a path. Note that the effective spread of the channel is higher than the physical channel spread due to the side-lobes of the TC filter response. As noted previously, for predictability of the Zak-OTFS channel response, it is required that the effective channel delay spread is small (less than T) and the effective channel Doppler spread is less than Δf.

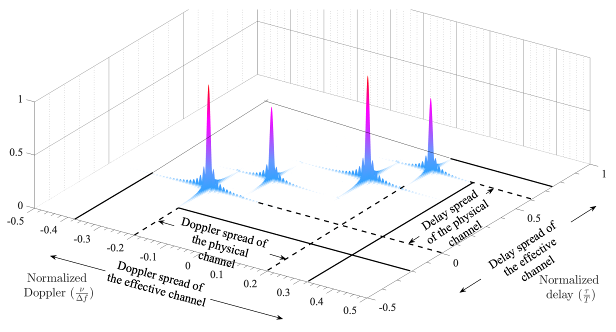


Fig. 9. $|h^{g_i}_{\text{dd}}(\tau, \nu)|$: DD domain effective channel for the considered four path channel.

The channel coefficients in the discrete DD domain for the two approaches are not the same, in general, since the phase terms associated with each corresponding path are different, as can be seen below.

$$h^{g_1}_{\text{dd}}[l,k] = \sum_{p=0}^{P-1} h_p \alpha(\tau_l - \tau_p)\beta(\nu_k - \nu_p) e^{j2\pi\nu_p(\tau_l-\tau_p)} \quad (61)$$

$$h^{g_2}_{\text{dd}}[l,k] = \sum_{p=0}^{P-1} h_p \alpha(\tau_l - \tau_p)\beta(\nu_k - \nu_p) e^{j2\pi\nu_k(\tau_l-\tau_p)} \quad (62)$$

However, other important factors such as the predictability of the channel response, and the sparsity of the effective channel matrix, are identical for the two approaches if using same windows since these are determined by the DD pulse shape and the side-lobe decay.

In summary, the effective DD channel response and DD pulse shape depend on the transmit TC filter, which in turn depends on the time and frequency windows used in our schemes. As will be shown in Section VIII, channel predictability depends on having a sufficiently compact DD pulse shape, and this requires sufficient bandwidth and time windowing resources. Furthermore, the quality of channel estimators and the capacity of the resulting channels also depends on the DD pulse shape.

VII. Time and Frequency domain Comparison of the two Approaches

This section considers our two implementation approaches from a time domain and frequency domain perspective and discusses implementation constraints. We show that our proposed Type-2 Zak-OTFS approach is the practical option for implementing windows with finite support. Although Type-1 Zak-OTFS can be modified for practical implementation, we show that the Type-2 Zak-OTFS approach has better spectral efficiency (transmitted data symbols per second per Hz).

A. Time Domain Comparison

For rectangular windows, note from (32) in Section IV (and using (51) for $\alpha(\tau)$) that the Type-1 TC basis function $\phi^{g_1}_{l,k}(t)$ is given by

$$\phi^{g_1}_{l,k}(t) = M \sum_{n=0}^{N-1} \underbrace{e^{j2\pi\frac{kn}{N}}}_{n^{\text{th}}\text{ sample of tone}} \underbrace{e^{j\pi\left(\frac{t}{T/M}-(nM+l)\right)} \text{sinc}\left(\frac{t}{T/M} - (nM+l)\right)}_{n^{\text{th}}\text{ sinc pulse centered at } nT + l\frac{T}{M}} \quad (63)$$

whereas the Type-2 TC basis function $\phi^{g_2}_{l,k}(t)$ is given by

$$\phi^{g_2}_{l,k}(t) := \underbrace{e^{j2\pi\nu_k(t-\tau_l)}\mathbb{I}_{[0,NT)}(t)}_{\text{Time limited tone}} \underbrace{\sum_{m=0}^{M-1} e^{j2\pi\frac{m}{M}\left(\frac{t}{T/M}-l\right)}}_{\text{Periodic pulse train}} \quad (64)$$

from (39) in Section IV.

For rectangular windows, it is clear from (63) that Type-1 TC basis functions are not time-limited for any pair (l,k) due to the side-lobes of the sinc function. Hence, the corresponding Zak-OTFS symbol is also not time-limited. In fact, the Type-1 Zak-OTFS symbol is not time-limited for any frequency window $A(f)$ that has a finite support, which includes rectangular windows and root raised cosine (RRC) windows, considered in [8]. As a consequence, the symbols considered in [8] cannot be realized in practice. However, we will show in the following subsection, that a modified Type-1 Zak-OTFS symbol can be realized by allowing roll-off of the frequency window.

In contrast, the Type-2 TC filtered basis function is strictly time-limited to the interval $[0, NT)$ from (64) and can be realized using our proposed DD-OFDM implementation. Our Type-2 Zak-OTFS implementation with finite support windows always leads to a time-limited waveform and a finite number of DD-OFDM sub-carrier symbols.

B. Practical Implementation of Type-1 Zak-OTFS

As discussed in the previous section, the Type-1 Zak-OTFS implementation is not possible for a window $A(f)$ that has a finite support, since it results in an unlimited time-domain symbol. We now show that a modified Type-1 symbol can be realized by allowing roll-off of the frequency domain window.

We propose that the response of the interpolation filter for Type-1 implementation be chosen to be $\tilde{\alpha}(t) := \alpha(t)\mathbb{I}_{[-\frac{T_c}{2},\frac{T_c}{2}]}(t)$, *i.e.* the response of $\alpha(t)$ is truncated to be in the range $[-\frac{T_c}{2}, \frac{T_c}{2}]$. For this modified practical Type-1 Zak-OTFS implementation, the modified basis function is

$$\tilde{\phi}^{g_1}_{l,k}(t) = T \sum_{n\in\mathbb{Z}} \tilde{\alpha}(t - \tau_l - nT) B(\tau_l + nT) e^{j2\pi\nu_k nT} \quad (65)$$

Note that this basis function corresponds to a modified Type-1 TC filter $\tilde{\alpha}(\tau)\beta(\nu)$. Hence, the modified frequency domain window is $\tilde{A}(f) := A(f) * T_c \text{sinc}\left(\frac{f}{1/T_c}\right)$, whereas the time domain window remains as $B(t)$. Here, $*$ represents convolution. Note that the modified frequency window $\tilde{A}(f)$ does not have a finite support (even though the original window $A(f)$ has a finite support). Hence, we say this Type-1 symbol is windowed by the rolled-off version of $A(f)$. It can be noted that as T_c approaches infinity, the rolled-off implementation gets closer to the Type-1 Zak-OTFS symbol corresponding to filter $\alpha(t)$ (and the strictly band-limited window $A(f)$).

In this case, the time duration of the Zak-OTFS symbol is $NT + T_c$, *i.e.* $[-\frac{T_c}{2}, NT + \frac{T_c}{2})$. It is clear that practical implementation of Type-1 requires $\frac{T_c}{NT}$ additional time resources compared to Type-2. For example, for $T_c = 10T$, and $N = 100$, Type-1 implementation requires 10% more time resources. In the next section, we also show how the choice of T_c affects the frequency resource usage of Type-1 implementation compared to a Type-2 implementation.

C. Frequency Domain Comparison

In this section we compare the power spectral density of the Zak-OTFS symbol for our modified practical Type-1 implementation, and for our Type-2 implementation.

Let $\tilde{\Phi}^{g_1}_{l,k}(f) = \int \tilde{\phi}^{g_1}_{l,k}(t)e^{-j2\pi ft}dt$ be the Fourier transform of the modified Type-1 basis function $\tilde{\phi}^{g_1}_{l,k}(t)$, and $\Phi^{g_2}_{l,k}(f) = \int \phi^{g_2}_{l,k}(t)e^{-j2\pi ft}dt$ be the Fourier transform of the Type-2 basis function. They are given by

$$\tilde{\Phi}^{g_1}_{l,k}(f) = T \underbrace{\left(A(f) * T_c \mathrm{sinc}\left(\frac{f}{1/T_c}\right)\right)}_{\text{Frequency domain window}} e^{-j2\pi f \tau_l} \sum_{n\in\mathbb{Z}} B(\tau_l + nT) e^{-j2\pi(f-\nu_k)nT} \quad (66)$$

$$\Phi^{g_2}_{l,k}(f) = \sum_{m\in\mathbb{Z}} A(m\Delta f + \nu_k) e^{-j2\pi\tau_l(m\Delta f+\nu_k)} \beta(f - m\Delta f - \nu_k) \quad (67)$$

where $*$ denotes convolution.

It can be noted from (66) that the effective frequency window component of the modified Type-1 implementation is the window $A(f)$ convolved with a sinc function $T_c \mathrm{sinc}\left(\frac{f}{1/T_c}\right)$, due to the truncation of the filter response. This convolution leads to side-lobes outside the band of window $A(f)$, as well as an expansion of the first-null bandwidth. As T_c increases, the side-lobes decay at a higher rate and the bandwidth expansion reduces. The trade-off from increasing T_c is that the Type-1 Zak-OTFS symbol has a longer time duration.

From (67), the Fourier transform of the Type-2 basis function consists of a sum of frequency domain pulses. The side-lobes are a function of the frequency domain pulse shape $\beta(f)$ (which is determined by the time-domain window $B(t)$).

We now compare the expected power spectral density of our two practical implementations, with rectangular windows and with RRC windows. We also compare with the original MC-OTFS scheme [2] and with ODDM [9].

Fig. 10 presents the power spectral density comparison for rectangular windows. Note that MC-OTFS has the highest side-lobe levels of all schemes, followed by ODDM. Comparing the Zak-OTFS schemes, Type-1 with $T_c = 10T$ has the most rapid side-lobe roll-off, followed by Type-2, and then Type-1 with $T_c = T$. Note that for our Type-1 implementations, the first null is at approximately $0.5/T_c$, as can be seen from the zoomed in plots. Hence, our Type-1 implementation uses $M\Delta f + \frac{1}{T_c}$ bandwidth and $NT + T_c$ time resources, while our Type-2 implementation only uses $M\Delta f + \frac{\Delta f}{N}$ bandwidth and NT time resources. For example, Type-1 with $T_c = T$ requires 1.2% more bandwidth, and a 0.6% longer symbol duration compared to Type-2, and leads to 9 dB higher side-lobe levels, while Type-1 with $T_c = 10T$ requires 5.6% longer symbol duration compared to Type-2.

Fig. 11 presents the power spectral density comparison for more practical RRC windows with roll-off factors $r = 0.01$ and 0.1. Note, by comparing with Fig. 10, that the performance of our Type-2 implementation is significantly better with RRC windows, compared to rectangular windows. For example, the side-lobe levels for Type-2 with RRC for small roll-off factor $r = 0.01$ are $\approx$ 36 dB lower at $f = -5\Delta f$, compared with rectangular windows. In fact, our Type-2 implementation performance with RRC windows is better than all the schemes considered for rectangular windows in Fig. 10.

Clearly, there is a first-null bandwidth expansion with RRC windows, for both $r = 0.01$ and 0.1, compared to rectangular windows (Fig. 10). In terms of RRC side-lobe comparison, for $r = 0.01$, the side-lobe performance of Type-2 is similar

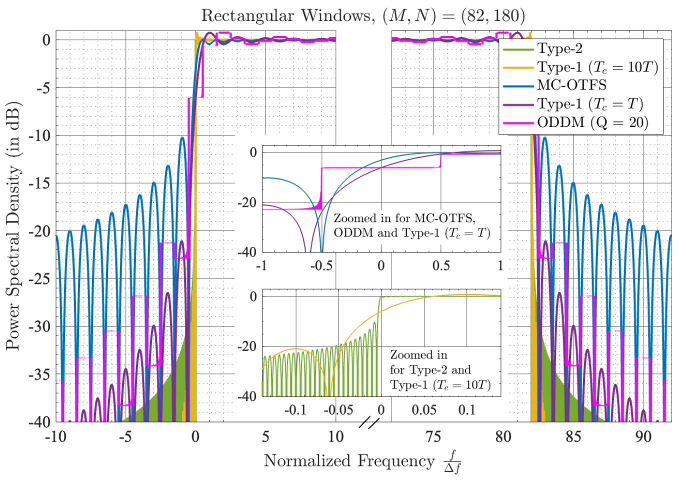


Fig. 10. Expected Power Spectral Density of the two implementations with rectangular windows, along with MC-OTFS and ODDM.

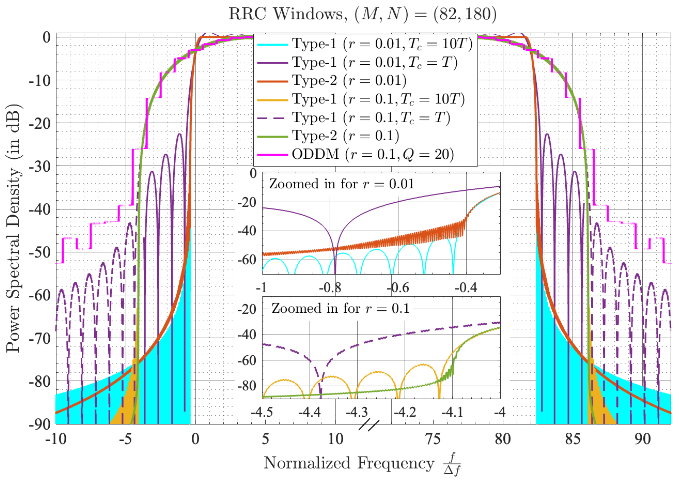


Fig. 11. Expected Power Spectral Density comparison of the two implementations with RRC windows, and ODDM.

to Type-1 with $T_c = 10T$; and both of those are superior to Type-1 with $T_c = T$. For $r = 0.1$, the side-lobe performance of Type-2 is significantly superior to all the other schemes.

VIII. Channel Predictability

In this section, we demonstrate that the channel response can be estimated in an efficient way, for each of our Zak-OTFS implementations. Since all data symbols $\hat{x}[l,k]$ undergo the same transformation by the channel, it is only necessary to use one of them as a pilot symbol to obtain all channel taps. This is termed *channel predictability* [8]. We compare our Type-1 and Type-2 implementations in terms of channel predictability, for rectangular and RRC windows.

A. Pilot Response

From (9) and (3), the output response to a pilot signal $\hat{x}[l',k'] = \delta[l' - l_\circ]\delta[k' - k_\circ]$ for a DD data grid location $(l_\circ, k_\circ) \in \{0,\dots,M-1\} \times \{0,\dots,N-1\}$ is given by

$$y_{\mathrm{dd}}[l,k] = e^{j2\pi\frac{l_\circ(k-k_\circ)}{MN}} h_{\mathrm{plt}}[l,k|l_\circ,k_\circ] \quad (68)$$

for $(l,k) \in \mathbb{Z} \times \mathbb{Z}$, where

$$h_{\text{plt}}[l,k|l_\circ,k_\circ] := \; h_{\text{dd}}[l-l_\circ,k-k_\circ] + \sum_{(m,n)\in\mathbb{Z}\times\mathbb{Z}-(0,0)} e^{j2\pi\frac{ml_\circ}{M}} e^{-j2\pi\frac{nk}{N}} h_{\text{dd}}[l-l_\circ+nM,k-k_\circ+mN] \quad (69)$$

where $h_{\text{dd}}[\cdot,\cdot]$ is the sampled effective delay-Doppler channel response as defined in (10). We call $h_{\text{plt}}[l,k|l_\circ,k_\circ]$ the *untwisted pilot response* (UPR), since the overall output response in (68) includes a phase twist term. Note that the UPR is a superposition of the sampled effective channel response and its aliased phase twisted copies.

The effective channel samples (or taps) can be estimated from the UPR as

$$\begin{aligned}\hat{h}_{\text{dd}}[l,k|l_\circ,k_\circ] &:= h_{\text{plt}}[l+l_\circ,k+k_\circ|l_\circ,k_\circ] \quad (70)\\ &= h_{\text{dd}}[l,k] \; + \\ \sum_{(m,n)\in\mathbb{Z}\times\mathbb{Z}-(0,0)} & h_{\text{dd}}[l+nM,k+mN]e^{j2\pi\frac{ml_\circ}{M}}e^{-j2\pi\frac{n(k+k_\circ)}{N}} \quad (71)\end{aligned}$$

for $(l,k) \in \mathcal{S}(h_{\text{dd}})$, where $\mathcal{S}(h_{\text{dd}}) \subseteq \mathbb{Z}^2$ is the support of the effective channel response $h_{\text{dd}}[\cdot,\cdot]$. Note that the summation in (71) is the estimation error term due to DD domain aliasing, for a given pilot location $(l_\circ,k_\circ) \in \{0,\dots,M-1\} \times \{0,\dots,N-1\}$ and a DD tap location $(l,k) \in \mathcal{S}(h_{\text{dd}})$.

B. Channel Predictability

The measured UPR perfectly estimates the channel taps in the case when $\mathcal{S}(h_{\text{dd}})$ is limited to a $M \times N$ grid. For example, suppose that $\mathcal{S}(h_{\text{dd}}) = \{0,\dots,M-1\}\times\{-N/2,\dots N/2-1\}$, then all the aliased terms in (71) equal zero for $(l,k) \in \mathcal{S}(h_{\text{dd}})$, since the aliased copy corresponding to $(n,m) \neq (0,0)$ has the support $\{nM,\dots,(n+1)M-1\}\times\{mN-N/2,\dots mN+N/2-1\}$ which has no overlap with $\mathcal{S}(h_{\text{dd}})$. Hence in this scenario, the effective channel taps are perfectly estimated from the UPR of any pilot $(l_\circ,k_\circ)$ as given in (70), *i.e.* $\hat{h}_{\text{dd}}[l,k|l_\circ,k_\circ] = h_{\text{dd}}[l,k]$ for all $(l_\circ,k_\circ) \in \{0,\dots,M-1\} \times \{0,\dots,N-1\}$. Furthermore, the UPR for all pairs $(l_\circ,k_\circ)$ can be obtained by any single pilot. In this scenario, the channel is said to be perfectly *predictable* [8].

However in general, the support of the effective channel is not always finite because of the spread due to DD pulse shape and also due to the physical channel parameter constraints. In such situations, the effective channel taps will not be estimated perfectly, and further, an error will be incurred when the UPR for a pilot at $(l_\circ,k_\circ)$ is used to predict the UPR of another pilot, located at $(l',k') \neq (l_\circ,k_\circ)$. Hence to measure the channel *predictability* in general scenarios, we define the relative channel prediction error as

$$E(h_{\text{dd}}) := \frac{\sum_{(l,k)\in\mathcal{S}_{h_{\text{dd}}}} \left|\hat{h}_{\text{dd}}[l,k|0,0] - \hat{h}_{\text{dd}}[l,k|M/2,N/2]\right|^2}{\sum_{(l,k)\in\mathcal{S}_{h_{\text{dd}}}} \left|\hat{h}_{\text{dd}}[l,k|M/2,N/2]\right|^2}$$

following [8]. The relative channel prediction error measures the relative error between the predicted taps of a pilot at $(0,0)$ and a pilot at $(M/2,N/2)$. Note that $E(h_{\text{dd}})$ equals zero when the channel is perfectly predictable. Hence, the channel predictability is said to be high, when $E(h_{\text{dd}})$ is small.

Note that high channel predictability means it is possible to detect the data symbols at different DD locations in the input data grid using the effective channel taps as measured by just a single pilot. This is therefore crucial for simple and efficient channel estimation, and for data symbol detection performance.

We now demonstrate channel predictability in various types of channels. To do this, we use the following Theorem 4, which provides UPR equations for Type-1 and Type-2 Zak-OTFS implementations in a sparse P path channel.

Theorem 4. *Consider a sparse path channel with a delay-Doppler response* $h(\tau,\nu) = \sum_{p=0}^{P-1} h_p\delta(\tau-\tau_p)\delta(\nu-\nu_p)$, *and let the receive TC filter be* $g^{\text{Rx}}(\tau,\nu) = \delta(\tau)\delta(\nu)$. *Then,*

- *The UPR for a Type-1 Zak-OTFS with transmit TC filter* $g_1(\tau,\nu) = \alpha(\tau)\beta(\nu)$ *is given by* (72)*;*
- *The UPR for a Type-2 Zak-OTFS with transmit TC filter* $g_2(\tau,\nu) = \alpha(\tau)\beta(\nu)e^{j2\pi\tau\nu}$ *is given by* (73)*;*

for pilot location $(l_\circ,k_\circ) \in \{0,\dots,M-1\} \times \{0,\dots,N-1\}$ *and* $(l,k) \in \mathbb{Z}\times\mathbb{Z}$, *where* $l_p := \frac{\tau_p}{T/M}$ *and* $k_p := \frac{\nu_p}{\Delta f/N}$.

Proof. The proof follows by substituting the effective channel responses (59),(60) in the definition of UPR given in (69), and then using Lemma 4. We omit the detailed proof due to lack of space. ☐

We now demonstrate channel predictability for three general classes of channels in the following subsections: the underspread integer channel, the overspread integer channel, and the underspread non-integer channel. Note that non-sparse channels can also be handled similarly to non-integer channels since it is sufficient to estimate the *effective channel taps* under high predictability. We consider our Type-2 Zak-OTFS implementation with rectangular windows.

Consider rectangular windows $A(f) = \frac{1}{M\Delta f}\mathbb{I}_{[0,M\Delta f)}(f)$ and $B(t) = \frac{1}{NT}\mathbb{I}_{[0,NT)}(t)$. By evaluating the Zak transform of the time window and the Dual Zak transform of the frequency

$$h_{\text{plt}}^{g_1}[l,k|l_\circ,k_\circ] = \sum_{p=0}^{P-1} h_p e^{-j2\pi\frac{l_\circ(k-k_\circ-k_p)}{MN}} e^{j2\pi\frac{k(l-l_\circ-l_p)}{MN}} \mathcal{D}_A\left((l-l_\circ-l_p)\frac{T}{M},(k-k_p)\frac{\Delta f}{N}\right) \mathcal{Z}_B\left(l_\circ\frac{T}{M},(k-k_\circ-k_p)\frac{\Delta f}{N}\right) \quad (72)$$

$$h_{\text{plt}}^{g_2}[l,k|l_\circ,k_\circ] = \sum_{p=0}^{P-1} h_p e^{-j2\pi\frac{(l-l_p)(k-k_\circ-k_p)}{MN}} e^{j2\pi\frac{k(l-l_\circ-l_p)}{MN}} \mathcal{D}_A\left((l-l_\circ-l_p)\frac{T}{M},k_\circ\frac{\Delta f}{N}\right) \mathcal{Z}_B\left((l-l_p)\frac{T}{M},(k-k_\circ-k_p)\frac{\Delta f}{N}\right) \quad (73)$$

window in (73) of Theorem 4 for rectangular windows, and using (70), the estimated channel taps for pilot (l', k') are

$$\hat{h}^{g_2}_{\text{dd}}[l,k|l',k'] = \sum_{p=0}^{P-1} h_p e^{-j2\pi\left\{\frac{l+l'-l_p}{M}\right\}\left(\frac{k-k_p}{N}\right)} e^{j2\pi\frac{(k+k')(l-l_p)}{MN}} \mathcal{F}_M\left(l_p - l\right)\mathcal{F}_N\left(k-k_p\right) \quad (74)$$

where $\{x\} := x - \lfloor x \rfloor$ is the fractional part of x, and $\mathcal{F}_N(x) := \frac{1}{N}\sum_{n=0}^{N-1} e^{-j2\pi\frac{nx}{N}}$ is the Dirichlet sinc function. $\mathcal{F}_N(x)$ is sometimes also referred to as *periodic* sinc function and as *aliased* sinc function, because it can be alternatively expressed as $\sum_{m\in\mathbb{Z}} e^{\frac{-j\pi(x-mN)}{N}}\text{sinc}(x-mN)$.

Hence, the estimated channel for a pilot placed at $(0,0)$ is

$$\hat{h}^{g_2}_{\text{dd}}[l,k|0,0] = \sum_{p=0}^{P-1} h_p e^{-j2\pi\left\{\frac{l-l_p}{M}\right\}\left(\frac{k-k_p}{N}\right)} e^{j2\pi\frac{k(l-l_p)}{MN}} \mathcal{F}_M\left(l_p - l\right)\mathcal{F}_N\left(k-k_p\right) \quad (75)$$

and the estimated channel for a pilot at $(M/2, N/2)$ is

$$\hat{h}^{g_2}_{\text{dd}}[l,k|M/2,N/2] = \sum_{p=0}^{P-1} h_p e^{-j2\pi\left\{\frac{l-l_p}{M}+\frac{1}{2}\right\}\left(\frac{k-k_p}{N}\right)} e^{j2\pi\frac{(k+\frac{N}{2})(l-l_p)}{MN}} \mathcal{F}_M\left(l_p - l\right)\mathcal{F}_N\left(k-k_p\right) \quad (76)$$

Recall from (62) that effective channel response is $h^{g_2}_{\text{dd}}[l,k] = \sum_{p=0}^{P-1} h_p e^{j\pi(l-l_p)} e^{-j\pi(k-k_p)} e^{j2\pi\frac{k(l-l_p)}{MN}}\text{sinc}(l - l_p)\text{sinc}(k-k_p)$, which is not localized due to the side-lobes of the sinc pulses. This means that aliasing in the DD domain as in (69) cannot be avoided, and this manifests in the form of *Dirichlet* sinc functions $\mathcal{F}_M$ and $\mathcal{F}_N$, along with phase terms in (74). Clearly, perfect predictability is not possible in general, because the side-lobes of the sinc function lead to a non-compact channel response and DD domain aliasing.

However in the special case of an integer path underspread channel, perfect predictability is possible as we will show in the next section. The underlying reason is that the nulls of the sinc pulses, line up exactly at the DD grid points in an integer channel. This means no aliasing from the sinc side-lobes and that the sampled channel response is compact. Aliasing of the main-lobes is also avoided because the channel is underspread.

C. Perfectly Predictable Underspread Integer Channel

Consider an integer path channel, where $k_p \in \mathbb{Z}$ and $l_p \in \mathbb{Z}$. We call the channel *underspread* if $-N/2 \le k_p \le N/2 - 1$ for all p, and $0 \le l_p \le M-1$ for all p. This corresponds to having Doppler spread $\max_p \nu_p - \min_p \nu_p \le \Delta f$ and delay spread $\max_p \tau_p - \min_p \tau_p \le T$. Note that the support of the channel is $\mathcal{S}(h_{\text{dd}}) = \{0,\ldots,M-1\}\times\{-N/2,\ldots,N/2-1\}$.

Lemma 5. *An underspread integer channel is perfectly predictable for a Type-2 Zak-OTFS implementation with rectangular windows.*

Proof. Let $(x)_N := x \mod N$ for integer x. Since $\mathcal{F}_N(x) = \delta[(x)_N]$ for integer x; in an integer channel, (75), (76) become

$$\hat{h}^{g_2}_{\text{dd}}[l,k|0,0] = \sum_{p=0}^{P-1} h_p e^{-j2\pi\left\{\frac{l-l_p}{M}\right\}\left(\frac{k-k_p}{N}\right)} e^{j2\pi\frac{k(l-l_p)}{MN}} \delta[(l-l_p)_M]\delta[(k-k_p)_N] \quad (77)$$

$$\hat{h}^{g_2}_{\text{dd}}[l,k|M/2,N/2] = \sum_{p=0}^{P-1} h_p e^{-j2\pi\left\{\frac{l-l_p}{M}+\frac{1}{2}\right\}\left(\frac{k-k_p}{N}\right)} e^{j2\pi\frac{\left(k+\frac{N}{2}\right)(l-l_p)}{MN}} \delta[(l-l_p)_M]\delta[(k-k_p)_N]. \quad (78)$$

Note that $l - l_p \in \{-(M-1),\ldots,M-1\}$ and $k - k_p \in \{-(N-1),\ldots,N-1\}$, since both $(l,k) \in \mathcal{S}(h_{\text{dd}})$ and $(l_p,k_p) \in \mathcal{S}(h_{\text{dd}})$. Hence, $\delta[(k-k_p)_N] = \delta[k-k_p]$ and $\delta[(l-l_p)_M] = \delta[l-l_p]$. Therefore, (77) and (78) become

$$\hat{h}^{g_2}_{\text{dd}}[l,k|0,0] = \sum_{p=0}^{P-1} h_p \delta[l-l_p]\delta[k-k_p] \quad (79)$$

$$\hat{h}^{g_2}_{\text{dd}}[l,k|M/2,N/2] = \sum_{p=0}^{P-1} h_p \delta[l-l_p]\delta[k-k_p] \quad (80)$$

which are exactly same as the effective channel taps $h^{g_2}_{\text{dd}}[\cdot,\cdot]$. The relative channel prediction error is 0 since the channel estimates in (79) and (80) are identical. This shows that the integer underspread channel is perfectly predictable. ☐

D. Non-predictable Taps in an Overspread Integer Channel

An overspread channel leads to non-predictability even with integer paths, as we show here. The underlying reason is aliasing of the main lobes of the sinc pulses in $h^{g_2}_{\text{dd}}[.,.]$.

Consider an overspread integer channel with 4 paths, where the path 0 has a normalized delay of $l_0 = 0$ and a normalized Doppler shift $k_0 = -N/2$. For path 1, $k_1 = N/2+1$, $0 \le l_1 \le M-1$. For path 2, $l_2 = M+1$ and $-N/2 \le k_2 \le N/2-1$, and for path 3, $l_3 = 1$ and $k_3 = k_2$.

This channel is overspread because the Doppler spread $(k_1 - k_0)\frac{\Delta f}{N} > \Delta f$ and the delay spread $(l_2 - l_0)\frac{T}{M} > T$. We now demonstrate that an overspread channel leads to non-predictable taps.

From (77) in the proof of Lemma 5, we obtain the channel estimate of pilot $(0,0)$ as

$$\hat{h}^{g_2}_{\text{dd}}[l,k|0,0] = h_0\delta[l]\delta\left[k+\frac{N}{2}\right] + h_1\delta[l-l_1]\delta\left[k+\frac{N}{2}-1\right] + \left(h_2 e^{-j2\pi\frac{k_2}{N}} + h_3\right)\delta[l-1]\delta\left[k-k_3\right] \quad (81)$$

for $l = 0,\ldots,M-1$ and $k = -N/2+1,\ldots,N/2-1$. Note that the tap gain and location corresponding to path 0 is measured correctly. However, for path 1, the Doppler tap location is measured to be $-N/2+1$, but this is not the actual path Doppler shift of $N/2+1$, due to aliasing. For path 2, the aliased delay shift has caused it to superpose onto path 3's tap $(1,k_3)$ and corrupt it. This demonstrates that the overspread taps cannot be measured accurately.

Furthermore, consider the channel estimate from pilot $(M/2, N/2)$ for this channel as follows:

$$\hat{h}^{g_2}_{\text{dd}}[l,k|M/2,N/2] = h_0\delta[l]\delta\left[k+\frac{N}{2}\right] - h_1\delta[l-l_1] \delta\left[k+\frac{N}{2}-1\right] + \left(-h_2 e^{-j2\pi\frac{k_2}{N}} + h_3\right)\delta[l-1]\delta\left[k-k_3\right]$$

Note that for the overspread taps, the measured gains here are completely different to what was measured from the pilot at

$(0,0)$. The relative prediction error for this example, $E(h_{\rm dd}) = \frac{4(|h_1|^2+|h_2|^2)}{\sum_{p=0}^{3}|h_p|^2}$ is very high, unless the overspread path gains h_1, h_2 are negligible, *i.e.* $|h_1|^2+|h_2|^2 \ll |h_0|^2+|h_3|^2$.

E. High Predictability in a General Underspread Non-Integer Channel

In a non-integer path channel, perfect predictability is not possible due to the side-lobes of the DD domain sinc pulses even when the channel is underspread. If a path does not fall exactly on a grid point, the effective channel $h_{\rm dd}^{g_2}[.,.]$ has infinitely many taps due to the side-lobes of the sinc pulse. The resulting DD domain aliasing is tolerable as long as the taps with significant energy are highly predictable, *i.e.* a small relative prediction error. We demonstrate the factors that effect channel predictability in a non-integer channel below.

Let $S_M < M/2$ (and $S_N < N/2$) denote the number of significant side-lobes of $\mathcal{F}_N$ (and $\mathcal{F}_M$) functions respectively, *i.e.* S_M is a positive integer such that $|\mathcal{F}_M(x)| < \epsilon_M$ for $S_M < |x| \le M/2$ for some small positive value ϵ_M. For example, $S_{100} = 30$ for $\epsilon_{100} = 0.0123$, and $S_{500} = 26$ for $\epsilon_{500} = 0.0123$.

Hence note from (75), the energy of a path p in a non-integer channel spreads into the taps $\{\lfloor l_p \rfloor - S_M, \ldots, \lfloor l_p \rfloor + S_M\} \times \{\lfloor k_p \rfloor - S_N, \ldots, \lfloor k_p \rfloor + S_N\}$, where its contribution is significant. The non-integer channel is underspread if the effective delay spread $2S_M + \max_p \lfloor l_p \rfloor \le M$ and $2S_M + \max_p \lfloor k_p \rfloor - \min_p \lfloor k_p \rfloor \le N$.

From (75), (76), we note that the prediction error for tap (l,k), $E[l,k] := \left| \hat{h}_{\rm dd}^{g_2}[l,k|0,0] - \hat{h}_{\rm dd}^{g_2}[l,k|M/2,N/2] \right|$ can be upper bounded as

$$E[l,k] \le \sum_{p=0}^{P-1} \Bigg| h_p \mathcal{F}_M(l_p - l)\, \mathcal{F}_N(k - k_p) \left(1 - e^{-j2\pi\left(\frac{k-k_p}{N}\right)\left(\left\{\frac{l-l_p}{M}+\frac{1}{2}\right\} - \left\{\frac{l-l_p}{M}\right\}\right)}\right) \Bigg| \quad (82)$$

Note that $\left\{\frac{l-l_p}{M}+\frac{1}{2}\right\} - \left\{\frac{l-l_p}{M}\right\} = 1/2$ if $\left\{\frac{l-l_p}{M}\right\} < 1/2$, and it equals $-\frac{1}{2}$ otherwise. In either case, the expression in (82) becomes

$$E[l,k] \le \sum_{p=0}^{P-1} \left| h_p \mathcal{F}_M(l_p - l)\, \mathcal{F}_N(k-k_p) \sin\left(\frac{\pi(k-k_p)}{2N}\right) \right|.$$

Note that $|\mathcal{F}_M(l_p - l)| \le \epsilon_M$ for paths p which do not contribute significantly to delay tap l, and similarly $|\mathcal{F}_N(k_p - k)| \le \epsilon_N$ for paths do not contribute significantly to Doppler tap k. Hence for the tap error from these paths to be negligible, ϵ_M, ϵ_N must be negligible, *i.e.* low side-lobe values of $\mathcal{F}_N, \mathcal{F}_M$ functions are needed, which requires M, N to be large values. For paths that do contribute significantly to the DD tap (l,k), note that the error depends on $|\sin\left(\frac{k-k_p}{N}\right)| \le \sin\left(\frac{S_N}{2N}\right)$. This is negligible when $S_N \ll N$, *i.e.* when the roll-off of side-lobes of $\mathcal{F}_N$ function is rapid.

For a given Zak-OTFS symbol time $W_{\rm t}$ and bandwidth $W_{\rm f}$, the time-bandwidth product MN is fixed, since $M = \frac{W_{\rm f}}{\Delta f}$, $N = \frac{W_{\rm t}}{T}$, and $\Delta f = \frac{1}{T}$, which implies $MN = W_{\rm t} W_{\rm f}$. This constraint means that M and N cannot both be chosen large independently of each other. A tradeoff between M and N is determined by the choice of Δf (or equivalently T) and this choice therefore impacts the channel predictability.

Apart from parameter choices and the channel, predictability also depends on the choice of windows, where similar observations can be made. For high predictability with general windows $A(f)$, $B(t)$, we require that the dual Zak transform $\mathcal{D}_A$ and the Zak transform $\mathcal{Z}_B$ functions in the UPR equation (73) have low side-lobe values and rapid roll-offs.

In the next subsection, using numerical simulations, we compare the channel predictability of Type-1 and Type-2 implementations for both rectangular windows and RRC windows, for various values of Δf, in a non-integer channel.

F. Numerical Results

In this section, we compare the predictability of Type-1 and Type-2 implementations, along with ODDM [9] and MC-OTFS [2] in the 3GPP Extended Vehicular A (EVA) channel model [12]. The path delays and gains of the channel are realized according to the power delay profile specified in the EVA model. For each path p, the Doppler shift $\nu_p = f_c \frac{v}{c}\cos(\theta_p)$ is realized according to Jakes' formula, where θ_p is uniformly randomly chosen between $-\pi$ and π. Here, v is the speed of the user equipment (UE), $f_c = 4$ GHz is the carrier frequency and c is the speed of light. We consider two UE speeds v of 500 kmph and 1000 kmph, which correspond to a maximum possible Doppler spread of 3.7 KHz and 7.4 KHz respectively. Note the channel model is a general non-integer delay and non-integer Doppler model. The simulation results are averaged over 10^4 channel realizations.

For the rectangular windows, we consider a bandwidth of $W_{\rm f} = 3.75$ MHz and a symbol time duration of $W_{\rm t} = 4$ ms. For the RRC windows with roll-off factor r, the bandwidth is $W_{\rm f}(1+r)$ and the symbol time duration is $W_{\rm t}(1+r)$. For each value of Δf, M is chosen to be $\frac{W_{\rm f}}{\Delta f}$, rounded down to the nearest even integer, and N is chosen to be $\frac{W_{\rm t}}{T}$, rounded down to the nearest even integer. The support of the channel is taken to be $\mathcal{S}(h_{\rm dd}) = \{-M/2, \ldots, M/2-1\} \times \{-N/2, \ldots, N/2-1\}$ for calculating the relative channel prediction error.

Fig. 12 presents the predictability comparison for rectangular windows at speed $v = 500$ kmph. Note that MC-OTFS has the worst performance of all the schemes. Type-1 ($T_c = T$) implementation has the best performance. This is because the delay pulse shape $\tilde{\alpha}(\tau)$ is localized to $[\frac{-T_c}{2}, \frac{T_c}{2}]$ due to the truncation of the filter response, which reduces the DD domain aliasing and improves predictability. However, as shown in Section VII-C, the improvement in predictability incurs a cost in terms of bandwidth expansion, additional symbol time and higher spectral side-lobe levels. Note for $\Delta f = 40$ KHz, $(M,N) = (82,180)$, and the spectral efficiency comparison for this case in presented in Section VII-C.

Fig. 13 presents the predictability comparison for rectangular windows at speed $v = 1000$ kmph. Firstly, note that since the maximum possible Doppler spread is 7.4 KHz in this case, the channel is overspread for $\Delta f = 5$ KHz, which leads to non-predictability as shown earlier. Hence, all the schemes

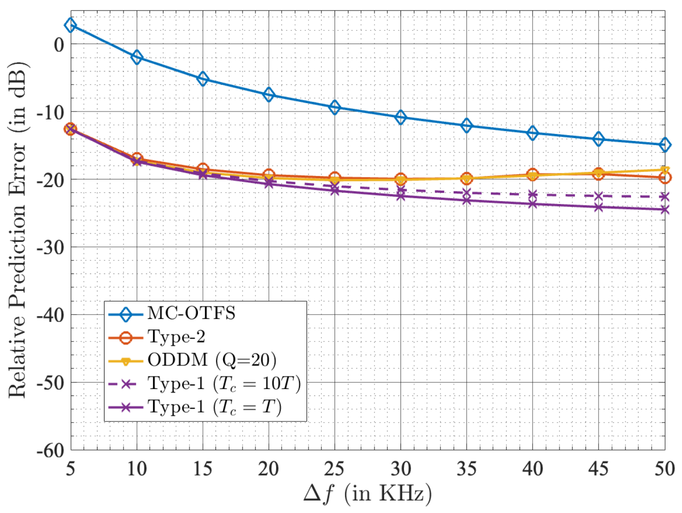


Fig. 12. Predictability comparison with rectangular windows at speed $v = 500$ kmph.

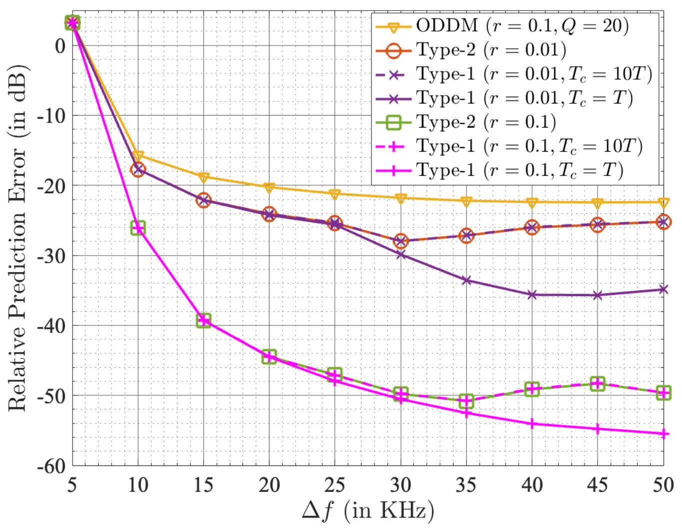


Fig. 14. Predictability comparisoin with RRC windows at 1000 kmph.

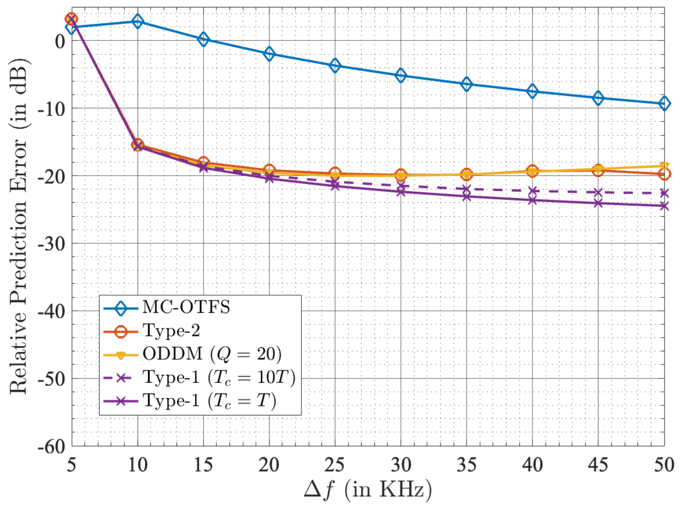


Fig. 13. Predictability comparisoin with rectangular windows at 1000 kmph.

have very high channel prediction error at $\Delta f = 5$ KHz. Also note that the prediction error for MC-OTFS has significantly increased over the entire range. This is because MC-OTFS does not have a predictable I/O relationship, unlike Zak-OTFS.

Fig. 14 presents the predictability comparison for RRC windows, at speed $v = 1000$ kmph. Note that predictability has improved significantly for all considered schemes, compared to the rectangular windows. Here, ODDM ($r = 0.1, Q = 20$) scheme has the worst performance, followed by Type-1 ($r = 0.01, T_c = 10T$) and Type-2 ($r = 0.01$) both of which have an identical performance. The Type-1 implementation with $T_c = T$ has best performance for both $r = 0.01, 0.1$.

Recall that the frequency window $A(f)$ and the delay pulse shape $\alpha(\tau)$ form a Fourier pair, and so do the time window $B(t)$ and Doppler pulse shape $\beta(\nu)$. Hence, spread of the time and frequency windows leads to a more localized pulse in the DD domain, which improves the channel predictability and increases the operational range of Zak-OTFS, *i.e.* the range of Doppler spread and delay spread values that can be handled. This is a key reason for the better predictability of RRC windows compared to rectangular windows, as well as the reason for superior performance of Type-1 implementation with $T_c = T$ compared to Type-2.

Clearly, there is a trade-off between channel predictability and spectral efficiency. For example, consider Type-1 ($T_c = T$) with rectangular windows and Type-2 ($r = 0.01$) with RRC windows. They both have nearly identical time-bandwidth product at $\Delta f = 40$ KHz; it is $1.018MN$ for Type-1 and $1.02MN$ for Type-2. The channel predictability values are also nearly identical; -24 dB for Type-1 and -26 dB for Type-2. However, the spectral out-of-band side-lobes are much higher for Type-1 as was shown in Fig. 10 and Fig. 11.

Hence, design of Zak-OTFS modulation requires a careful choice of TC filter type, windows, and choice of Δf to manage the trade-off between spectral efficiency, out-of-band emissions and channel predictability.

IX. Conclusions

In this paper, we have presented practical windowing methods to realize two types of twisted convolution filters. We have shown that these methods can be used to implement Zak-OTFS in two different ways. The Type-1 TC filter Zak-OTFS implementation can be performed using an interpolation filter on the precoded and windowed time domain symbols. The Type-2 TC filter Zak-OTFS implementation is based on modifying a precoded OFDM implementation called DD-OFDM. We show that our Type-2 implementation has an advantage over Type-1 due to better spectral efficiency, especially with practical RRC windows. We also compared the channel predictability of the two implementations, and demonstrated a trade-off between channel predictability and spectral efficiency.

Appendix

Proof of Lemma 1. From definition of the Zak transform in (1), $\mathcal{Z}_{s^g}(\tau, \nu) = \sum_{n \in \mathbb{Z}} s^g(\tau + nT)e^{-j2\pi\nu nT}$. Substituting

(11), we obtain

$$\mathcal{Z}_{s^g}(\tau,\nu) = \sum_{n\in\mathbb{Z}} \left(\iint g(\tau',\nu')s(\tau-\tau'+nT) e^{j2\pi\nu'(\tau-\tau'+nT)}d\tau' d\nu' \right) e^{-j2\pi\nu nT} \quad (83)$$

$$= \iint g(\tau',\nu') \left(\sum_{n\in\mathbb{Z}} s(\tau-\tau'+nT)e^{j2\pi(\nu-\nu')nT} \right) e^{j2\pi\nu'(\tau-\tau')}d\tau' d\nu' \quad (84)$$

$$= \iint g(\tau',\nu')\mathcal{Z}_s(\tau-\tau',\nu-\nu')e^{j2\pi\nu'(\tau-\tau')}d\tau' d\nu'$$

Hence, $\mathcal{Z}_{s^g}(\tau,\nu) = g *_\sigma \mathcal{Z}_s(\tau,\nu)$. □

Proof of Lemma 4. From the definition in (36), $\mathcal{D}_X(\tau,\nu) = \sum_{m\in Z} X(\nu+m\Delta f)e^{j2\pi\tau m\Delta f}$. Since $X(f)$ is the Fourier transform of $x(t)$, note that

$$\mathcal{D}_X(\tau,\nu) = \sum_{m\in\mathbb{Z}} \left(\int x(t)e^{-j2\pi(\nu+m\Delta f)t}dt \right) e^{j2\pi\tau m\Delta f}$$
$$= \int x(t)e^{-j2\pi\nu t} \sum_{m\in\mathbb{Z}} e^{-j2\pi m\Delta f(t-\tau)}dt \quad (85)$$

Since $\sum_{m\in\mathbb{Z}} e^{-j2\pi m\Delta f(t-\tau)}$ is the same as the impulse train $T\sum_{n\in\mathbb{Z}}\delta(t-\tau-nT)$, we obtain

$$\mathcal{D}_X(\tau,\nu) = Te^{-j2\pi\nu\tau}\sum_{n\in\mathbb{Z}} x(\tau+nT)e^{-j2\pi\nu nT} \quad (86)$$

Hence, $\mathcal{D}_X(\tau,\nu) = Te^{-j2\pi\nu\tau}\mathcal{Z}_x(\tau,\nu)$. □

Proof of Theorem 2. By definition of $X[i]$ in (42) and since $X_B[i] = X[i]B(iT_s)$, we obtain from (43) that

$$x_1(t) = T\sum_{n\in\mathbb{Z}}\sum_{l=0}^{M-1} x[l,n]B\left((nM+l)T_s\right) \alpha\left(t-(nM+l)T_s\right) \quad (87)$$

Since $T_s = \frac{T}{M}$, and from definition of $x[l,n]$ in (40),

$$x_1(t) = T\sum_{n\in\mathbb{Z}}\sum_{l=0}^{M-1}\sum_{k=0}^{N-1} \hat{x}[l,k]e^{j2\pi\frac{nk}{N}}B\left(nT+l\frac{T}{M}\right) \alpha\left(t-nT-l\frac{T}{M}\right) \quad (88)$$

Note from (32) that $x_1(t) = \sum_{k=0}^{N-1}\sum_{l=0}^{M-1}\hat{x}[l,k]\phi_{l,k}^{g_1}(t)$. □

Proof of Theorem 3. By definition of $\tilde{X}[i]$ in (48) and since $\tilde{X}_A[i] = \tilde{X}[i]A(i\frac{\Delta f}{N})$, we obtain from (49) that

$$x_2(t) = B(t)\sum_{m\in\mathbb{Z}}\sum_{k=0}^{N-1}\tilde{x}[m,k]A\left(m\Delta f+k\frac{\Delta f}{N}\right) e^{j2\pi(mN+k)\frac{\Delta f}{N}t} \quad (89)$$

From the definition of $\tilde{x}[m,k]$ in (46),

$$x_2(t) = B(t)\sum_{m\in\mathbb{Z}}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1}\hat{x}[l,k]e^{-j2\pi\frac{lT}{M}\frac{(mN+k)\Delta f}{N}} A\left(m\Delta f+k\frac{\Delta f}{N}\right)e^{j2\pi(mN+k)\frac{\Delta f}{N}t} \quad (90)$$

$$= \sum_{l=0}^{M-1}\sum_{k=0}^{N-1}\hat{x}[l,k]B(t)e^{j2\pi k\frac{\Delta f}{N}(t-\frac{lT}{M})} \sum_{m\in\mathbb{Z}} A\left(m\Delta f+k\frac{\Delta f}{N}\right)e^{j2\pi m\Delta f(t-l\frac{T}{M})} \quad (91)$$

Note from (39) that $x_2(t) = \sum_{k=0}^{N-1}\sum_{l=0}^{M-1}\hat{x}[l,k]\phi_{l,k}^{g_2}(t)$. □

References

[1] R. Hadani, S. Rakib, M. Tsatsanis, A. Monk, A. J. Goldsmith, A. F. Molisch, and R. Calderbank, "Orthogonal time frequency space modulation," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[2] R. Hadani and A. Monk, "OTFS: A new generation of modulation addressing the challenges of 5G," *arXiv preprint arXiv:1802.02623*, 2018.

[3] S. K. Mohammed, "Derivation of OTFS modulation from first principles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7619–7636, 2021.

[4] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, "Interference cancellation and iterative detection for orthogonal time frequency space modulation," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6501–6515, 2018.

[5] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for OTFS in delay-Doppler channels," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4906–4917, 2019.

[6] S. K. Mohammed, "Time-domain to delay-Doppler domain conversion of OTFS signals in very high mobility scenarios," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6178–6183, 2021.

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

[8] ——, "OTFS–Predictability in the delay-Doppler domain and its value to communication and radar sensing," *arXiv preprint arXiv:2302.08705*, 2023.

[9] H. Lin and J. Yuan, "Orthogonal delay-Doppler division multiplexing modulation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 11 024–11 037, 2022.

[10] S. Li, W. Yuan, Z. Wei, J. Yuan, B. Bai, and G. Caire, "On the pulse shaping for delay-Doppler communications," *arXiv preprint arXiv:2306.08704*, 2023.

[11] S. Gopalam, S. B. Pillai, P. Whiting, H. Inaltekin, I. B. Collings, and S. V. Hanly, "An OFDM based waveform for delay Doppler domain communication," *TechRxiv doi:10.36227/techrxiv.23723154*, 2023.

[12] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.104, 06 2023, version 18.2.0.