Implementation of Zak-OTFS Modulation using Time and Frequency Windowing

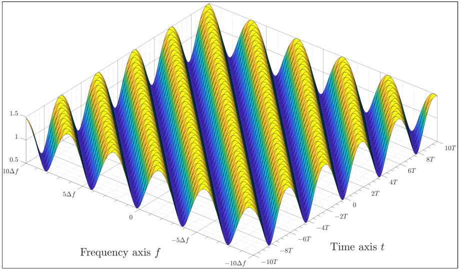
Swaroop Gopalam

Joint work with Iain Collings, Stephen Hanly, Hazer Inaltekin, Phil Whiting and Sibiraj B. Pillai

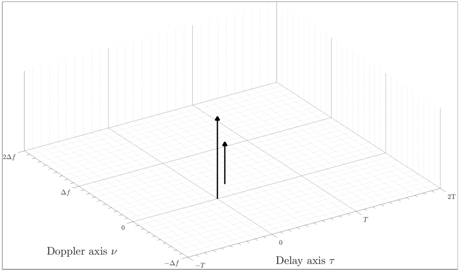
Macquarie University, Sydney, Australia
swaroop.gopalam@mq.edu.au

February 20, 2024

Motivation for delay-Doppler domain modulation

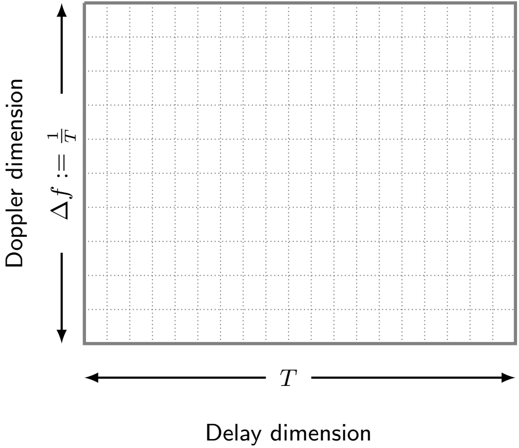


TF Representation

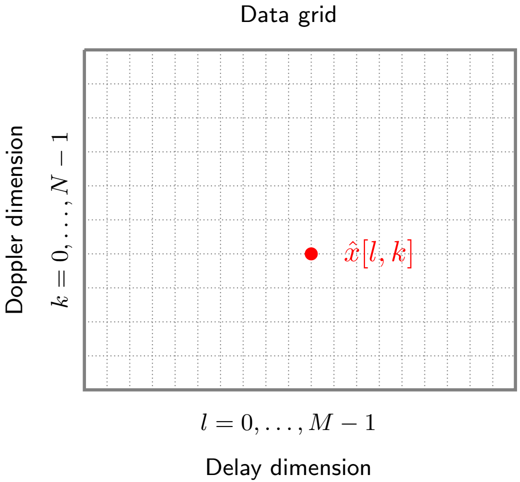


DD Representation

Delay-Doppler domain modulation

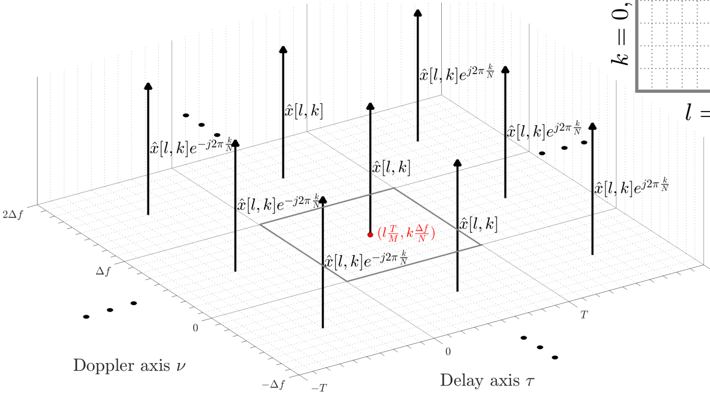


Delay-Doppler domain modulation



Zak-OTFS modulation

Zak-OTFS data signal $x_{\mathsf{dd}}(\tau,\nu)$ in the DD domain



Data grid

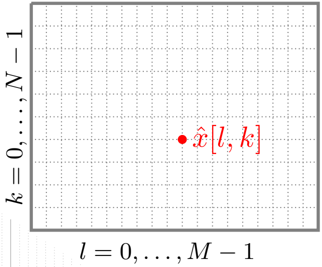
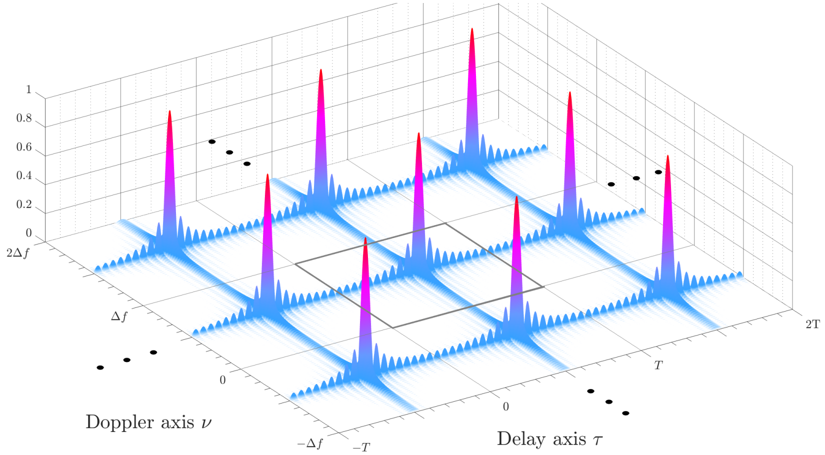
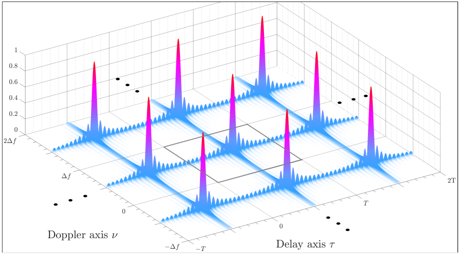
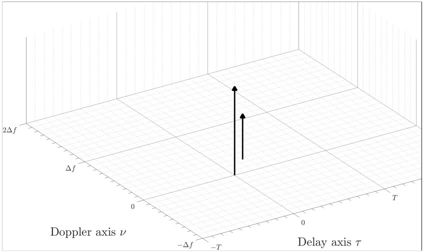


Illustration of a Zak-OTFS Filtered basis function



Channel operation in DD domain

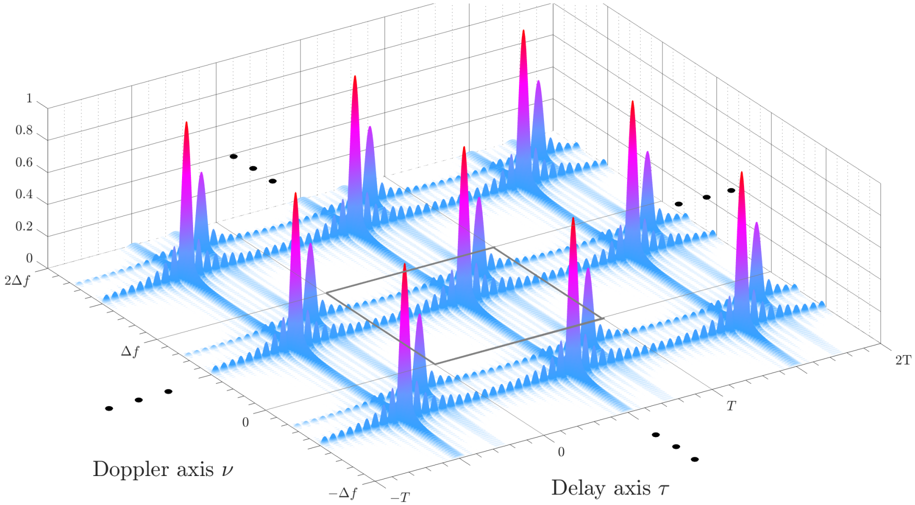




Input Signal $x_{\mathsf{dd}}(\tau, \nu)$

DD Channel $h(\tau, \nu)$

Twisted Convolution operation of the Channel

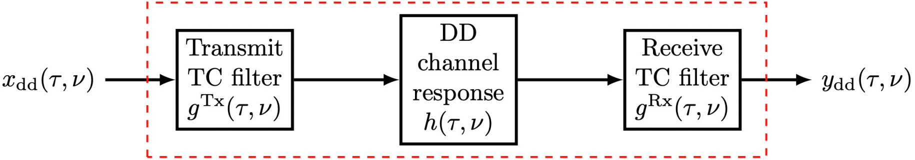


- Filtering in Zak-OTFS modulation is peformed using a two dimensional DD domain *twisted convolution* filter.

Definition

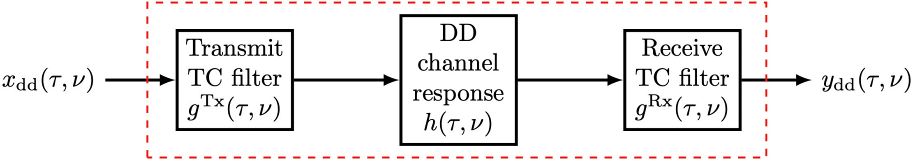
Twisted convolution of two DD functions $a(\tau,\nu)$, $b(\tau,\nu)$ is defined as

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



$$y_{dd}(\tau,\nu) = g^{Rx} *_\sigma \left(h *_\sigma \left(g^{Tx} *_\sigma x_{dd}(\tau,\nu)\right)\right)$$ [1]

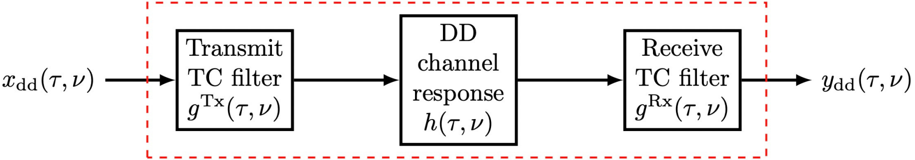
[1] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "OTFS - A mathematical foundation for communication and radar sensing in the delay-Doppler domain," *IEEE BITS the Information Theory Magazine*, vol. 2, no. 2, pp. 36-55, 2022.



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

Zak-OTFS I/O Relationship[2]

Effective DD channel response : $g^{\mathrm{Rx}} *_\sigma h *_\sigma g^{\mathrm{Tx}}(\tau,\nu)$



$$y_{\mathrm{dd}}(\tau,\nu) = \underbrace{g^{\mathsf{Rx}} *_\sigma h *_\sigma g^{\mathsf{Tx}}(\tau,\nu)}_{h_{\mathrm{dd}}(\tau,\nu)} *_\sigma x_{\mathrm{dd}}(\tau,\nu)$$

[2] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "OTFS-Predictability in the delay-Doppler domain and its value to communication and radar sensing," *arXiv preprint* arXiv:2302.08705, 2023.

Question (1)

How do you implement the TC filters in the time domain?

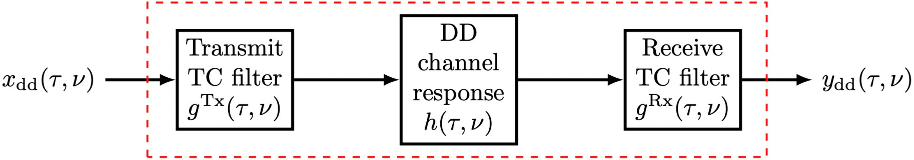
Question (2)

How do you generate the time domain signal for transmission?

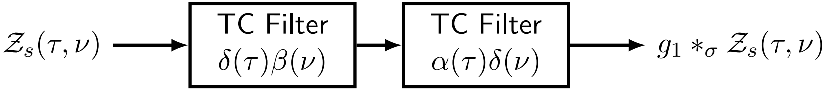
[3] S. Gopalam, I. B. Collings, S. V. Hanly, H. Inaltekin, S. R. B. Pillai and P. Whiting, "Zak-OTFS Implementation via Time and Frequency Windowing," in IEEE Transactions on Communications, doi: 10.1109/TCOMM.2024.3366403.

Type-1 TC Filters

Effective DD channel response : $g^{\mathrm{Rx}} *_\sigma h *_\sigma g^{\mathrm{Tx}}(\tau,\nu)$



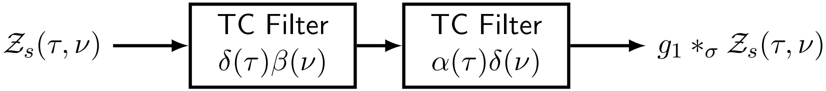
$$g_1(\tau,\nu) := \alpha(\tau)\beta(\nu)$$



DD domain processing steps

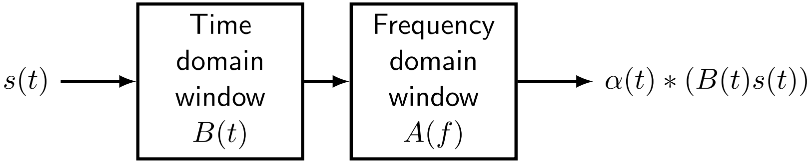
Type-1 TC Filters

$$g_1(\tau,\nu) := \alpha(\tau)\beta(\nu)$$



DD domain processing steps

$$\beta(\nu) \xrightarrow{\mathcal{F}_\nu^-} B(t); \quad \alpha(\tau) \xrightarrow{\mathcal{F}_\tau} A(f)$$



Equivalent time domain processing steps

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*

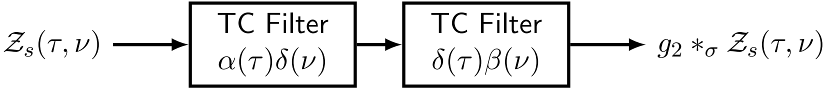
$$g_1 *_\sigma s(t) = \alpha(\tau)\delta(\nu) *_\sigma \delta(\tau)\beta(\nu) *_\sigma s(t) \tag{23}$$

$$= \int A(f) \left(\int B(t')s(t')e^{-j2\pi t' f} df \right) e^{j2\pi f t} dt \tag{24}$$

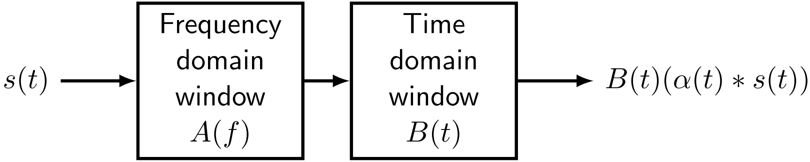
$$= \mathrm{IFT}(A(f)\ \mathrm{FT}\,(B(t)s(t))). \tag{25}$$

Type-2 TC Filters

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

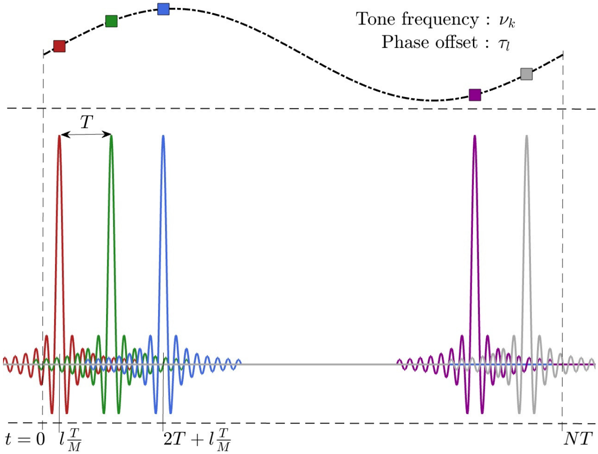


DD domain processing steps

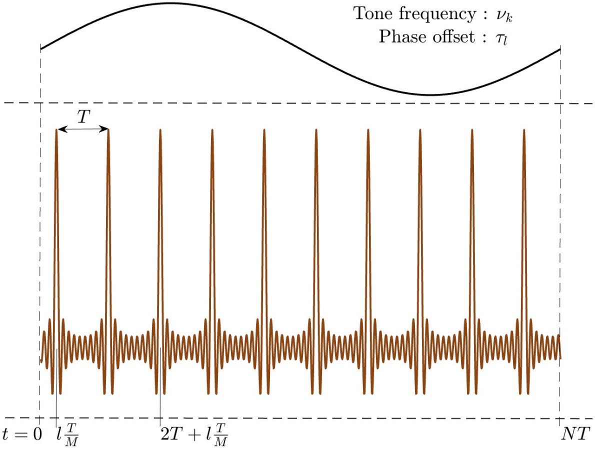


Equivalent time domain processing steps

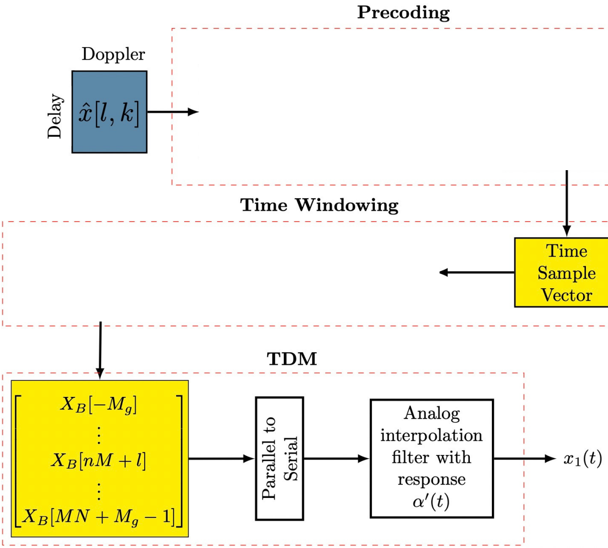
Type-1 Transmit Pulsone



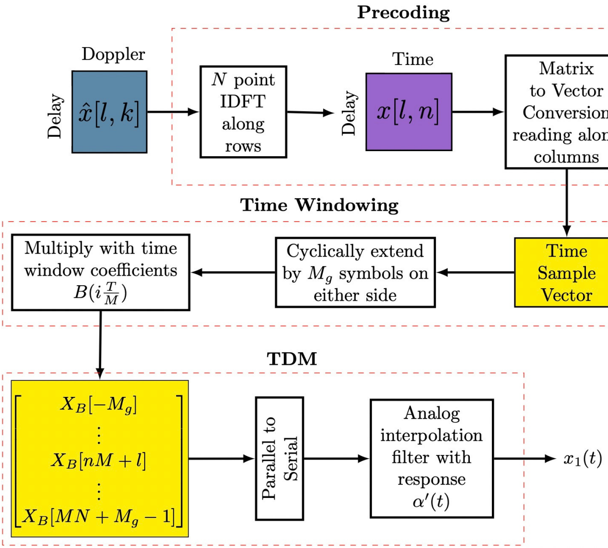
$$\phi_{l,k}^{g_1}(t) = \sum_{\tau \in \{\tau_l + nT\}_{n\in\mathbb{Z}}} \overbrace{B(\tau)e^{j2\pi\nu_k(\tau-\tau_l)}}^{\text{Windowed Tone Sample}} \overbrace{\alpha(t-\tau)}^{\text{Pulse}}$$



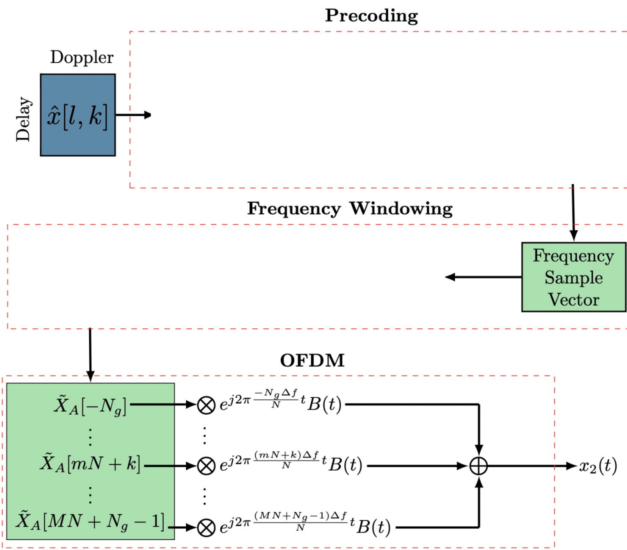
$$\phi^{g_2}_{l,k}(t) = \overbrace{B(t)e^{j2\pi\nu_k(t-\tau_l)}}^{\text{Windowed Tone}} \overbrace{\sum_{m\in\mathbb{Z}} A(m\Delta f + \nu_k)e^{j2\pi m\Delta f(t-\tau_l)}}^{\text{Periodic Pulse Train}} \quad (1)$$


Precoding
Doppler
Delay
$\hat{x}[l,k]$
Time Windowing
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 $\alpha'(t)$
$x_1(t)$

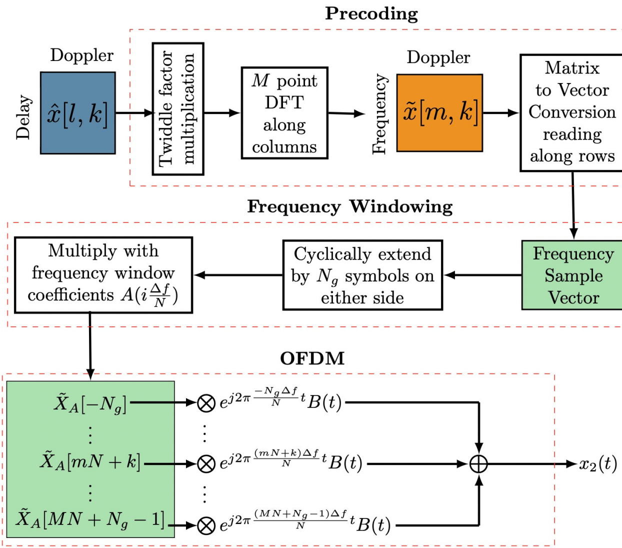
Type-1 Implementation



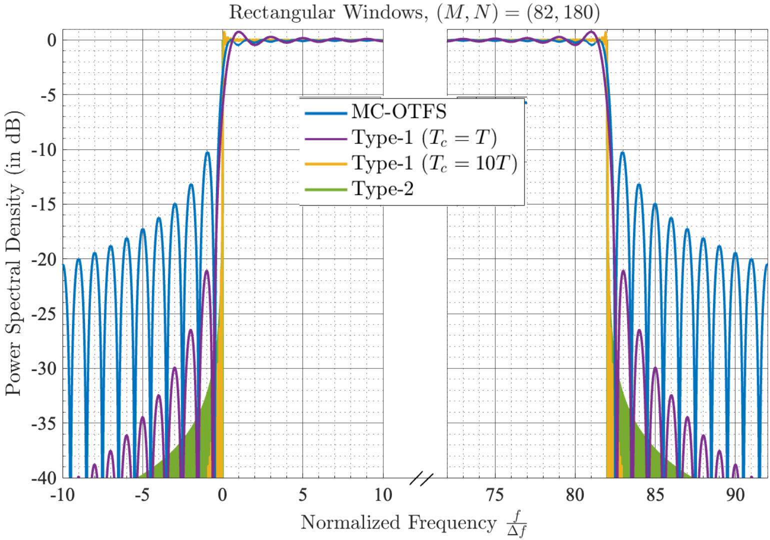
Type-2 Implementation



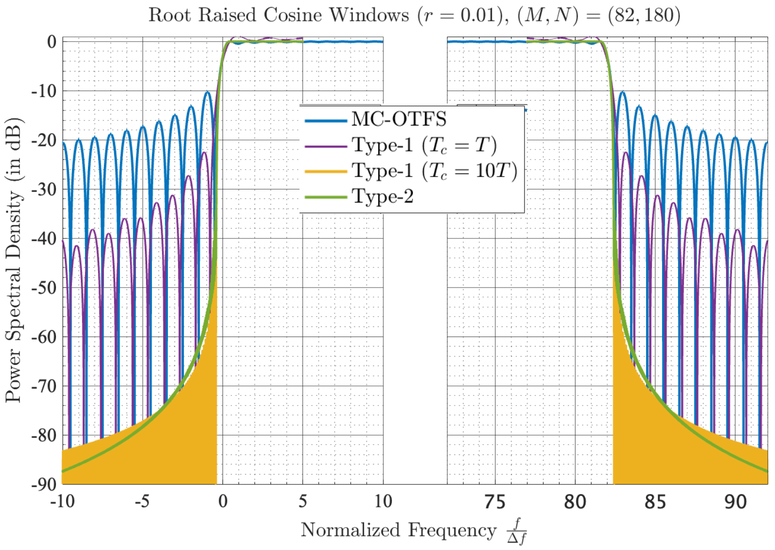
Type-2 Implementation

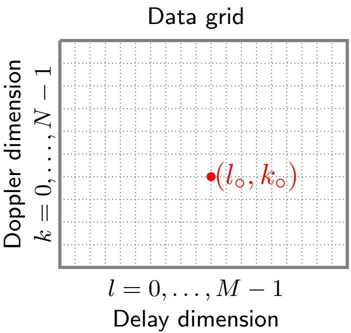


Frequency Domain Comparison



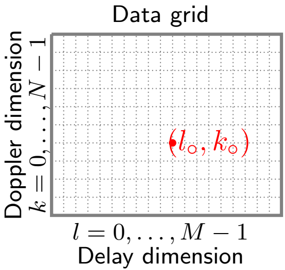
Frequency Domain Comparison





The pilot response for a location $(l_\circ, k_\circ)$ is

$$h_{\mathsf{plt}}[l, k | l_\circ, k_\circ] = h_{\mathsf{dd}}[l - l_\circ, k - k_\circ]$$

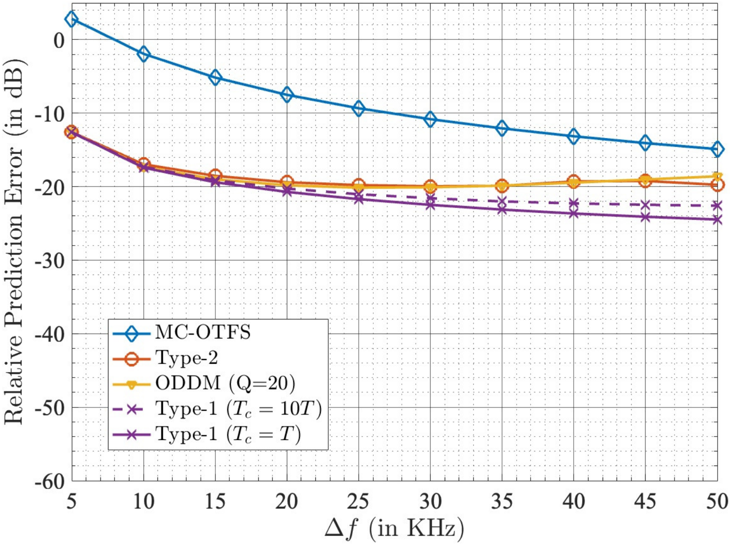


The pilot response[4] for a location $(l_\circ, k_\circ)$ is

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

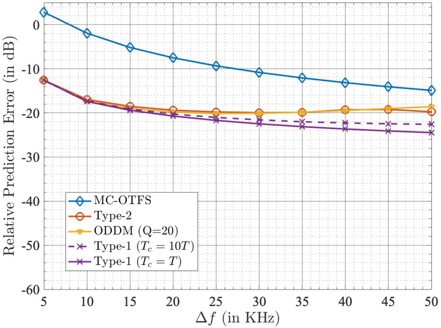
[4] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "OTFS-Predictability in the delay-Doppler domain and its value to communication and radar sensing," *arXiv preprint* arXiv:2302.08705, 2023.

Channel Predictability with Rectangular Windows

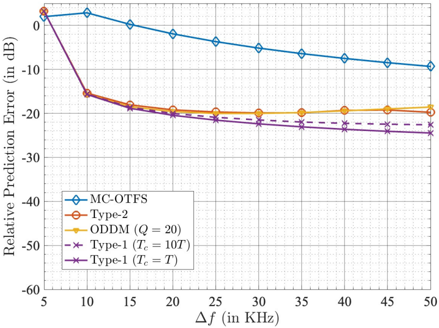


(c) For $\nu_{\max} = 3.7$ KHz

Channel Predictability with Rectangular Windows

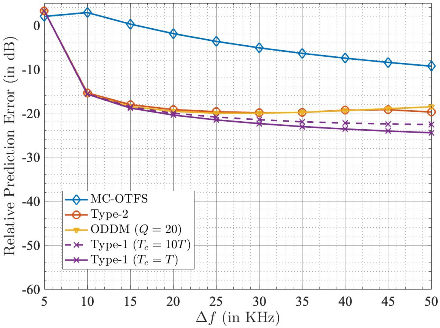


(d) For $\nu_{\max} = 3.7$ KHz

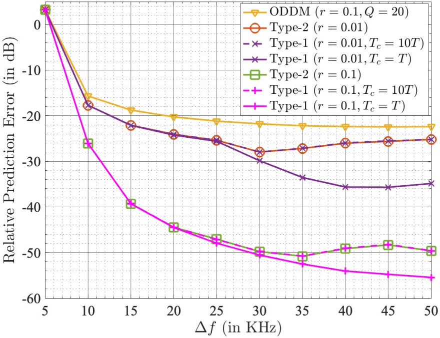


(e) For $\nu_{\max} = 7.4$ KHz

Channel Predictability with Root Raised Cosine Windows



(f) Rectangular for $\nu_{\max} = 7.4$ KHz



(g) RRC for $\nu_{\max} = 7.4$ KHz

- We have formulated Type-1 and Type-2 TC filters and showed that they can be implemented using time and frequency windowing.
- We have proposed two practical time domain implementations of Zak-OTFS that correspond to Type-1 and Type-2 TC filters.
- Both our proposed Zak-OTFS implementations have superior performance over the original MC-OTFS scheme, in terms of out-of-band emissions and channel predictability.
- Type-2 Implementation uses less symbol time and bandwidth overall compared to Type-1 implementation, and has less out-of-band emissions.
- Type-1 has slightly better channel predictability performance on account of using more symbol time and/or bandwidth.