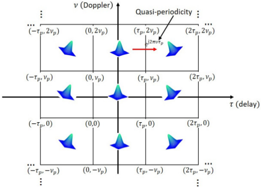
Learning in Zak-OTFS

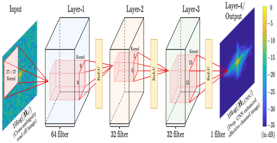
Workshop on Wireless Communication Technologies for the Next Decade
IIT Kanpur



A. Chockalingam
IISc, Bangalore

Joint work with Chetan Devendra Kabade and Arpan Das

16 August 2025



Outline I

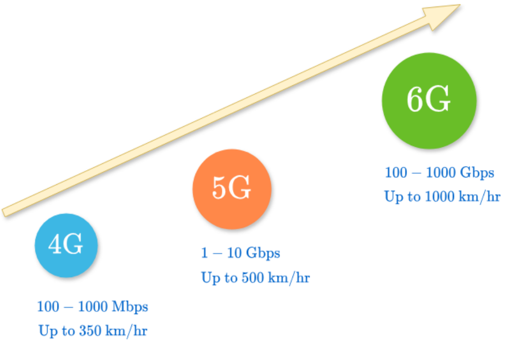
1. Zak-OTFS - A new waveform
   - Why a new waveform?
2. Zak transform / Inverse Zak transform
3. Zak-OTFS transceiver
4. Channel estimation in Zak-OTFS
   - Learning in delay-Doppler channel estimation[1]
5. Summary

[1] C. D. Kabade, A. Das, and A. Chockalingam, *Zak-OTFS with Superimposed Spread Pilot: CNN-Aided Channel Estimation,* to be presented in *IEEE PIMRC'2025 Workshop on Emerging Modulation Techniques Towards 6G Networks*, Istanbul, Sep. 2025.

Wireless systems evolution

- Demand for increased
  - data rate, spectral efficiency, energy efficiency (earlier focus)
  - mobility, # use cases, radar sensing support (new/current focus)

- 2G and 3G used CDMA (voice driven)
- 4G and 5G use OFDM, 5G uses massive MIMO (Internet/data driven)
- 6G and beyond: expect to be AI driven
- Several emerging technologies (including new waveforms) in 6G



[1] M. Z. Chowdhury, M. Shahjalal, S. Ahmed and Y. M. Jang, "6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957-975, Jul. 2020

Zak-OTFS - A new waveform

- Zak-OTFS
  - a modulation waveform as well as a radar sensing waveform in the delay-Doppler (DD) domain

- An analogy

- Waveform for LTI channels: CP-OFDM
- Information domain: Frequency domain
- Theory for describing and understanding: Fourier theory
- Transform: Fourier transform
- Fourier transform
  - Invented: 1822 (J.B.J.Fourier)
  - For modulation: 1966 (OFDM)
- Operation: Linear convolution

- Waveform for LTV channels: Zak-OTFS
- Information domain: Delay-Doppler domain
- Theory for describing and understanding: Zak theory
- Transform: Zak transform
- Zak transform
  - Invented: 1967 (Joshua Zak)
  - For modulation: 2022 (Zak-OTFS)
- Operation: Twisted convolution

Why a new waveform?

- Historically
  - PHY waveform has been a key differentiator between different generations of wireless
  - FDMA (1G) $\rightarrow$ TDMA (2G) $\rightarrow$ CDMA (2G,3G) $\rightarrow$ OFDM (4G,5G) $\rightarrow$??
- New use cases are emerging
  - High-mobility support
    - High-speed trains, aeroplanes
  - Non-terrestrial networks (NTN)
    - Drones, UAVs, LEOS
  - Radar sensing support
    - Autonomous cars/vehicles
- Legacy waveforms may not be adequate to meet new demands

Zak transform

- Zak transform
  - Parameterized by parameters $(\tau_{\mathrm{p}}, \nu_{\mathrm{p}})$ with $\tau_{\mathrm{p}}\nu_{\mathrm{p}} = 1$
    - τ_{p}: Doppler period, ν_{p}: Doppler period
  - Maps a time domain signal to a unique quasi-periodic DD domain signal

- Zak transform of a continuous time domain signal $a(t)$ is defined as

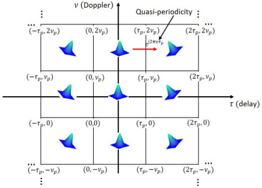
$$a(\tau, \nu) = Z_t\left(a(t)\right) \triangleq \sqrt{\tau_p} \sum_{k \in \mathbb{Z}} a(\tau + k\tau_{\mathrm{p}}) e^{-j2\pi\nu k\tau_{\mathrm{p}}}$$

- Quasi-periodicity
  - For any $n, m \in \mathbb{Z}$, $a(\tau, \nu)$ satisfies

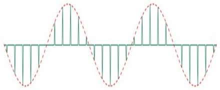
$$a(\tau + n\tau_{\mathrm{p}}, \nu + m\nu_{\mathrm{p}}) = e^{j2\pi n\nu\tau_{\mathrm{p}}} a(\tau, \nu)$$

  - Periodic along Doppler, and periodic with a multiplicative phase term $e^{j2\pi n\nu\tau_{\mathrm{p}}}$ along delay

- $a(\tau, \nu)$ - a DD pulse



- $a(t)$ - Pulsone

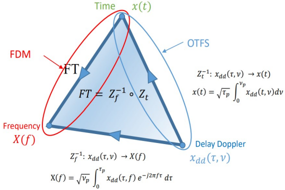


Inverse Zak transform

- Inverse Zak transform
  - Gives the time domain realization of a quasi-periodic DD domain signal
  - Exists only for DD functions which are quasi-periodic
  - Inverse Zak transform of a DD signal $a(\tau, \nu)$ is defined as

$$a(t) = Z_t^{-1}(a(\tau, \nu)) \triangleq \sqrt{\tau_p} \int_0^{\nu_p} a(t, \nu) d\nu$$

- Transform triangle



- Twisted convolution $(*_\sigma)$
  - TC between two DD functions $a(\tau, \nu)$ and $b(\tau, \nu)$ is defined as

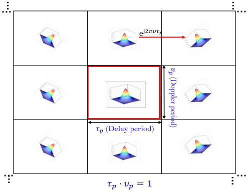
$$a(\tau, \nu) *_\sigma b(\tau, \nu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(\tau', \nu') b(\tau - \tau', \nu - \nu') e^{j2\pi\nu'(\tau - \tau')} d\tau' d\nu'$$

  - Associative. Non-commutative
  - Twisted convolution operation preserves quasi-periodicity

Information multiplexing in DD domain

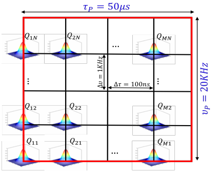
- Basic information carrier: a DD domain pulse (a pulsone in time domain)

- Fundamental DD period, $\mathcal{D}_0$ (red box):
  $\mathcal{D}_0 = \{(\tau, \nu) : 0 \leq \tau < \tau_{\mathrm{p}},\ 0 \leq \nu < \nu_{\mathrm{p}}\}$



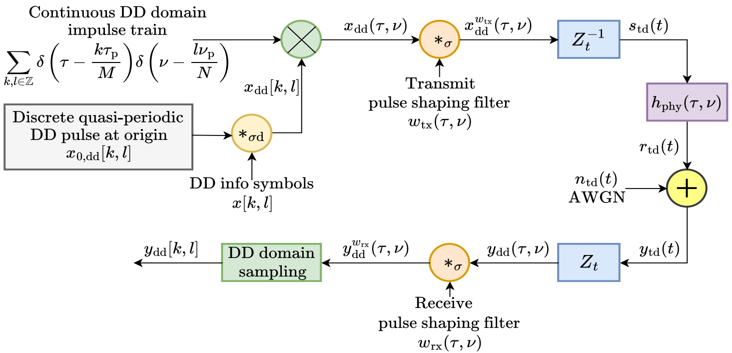
- Period lattice Λ_{p}
- τ_{p} is sliced into M delay bins
  - Delay resolution: $\Delta\tau = \frac{\tau_{\mathrm{p}}}{M}$
- ν_{p} is sliced into N Doppler bins
  - Doppler resolution: $\Delta\nu = \frac{\nu_{\mathrm{p}}}{N}$
- MN symbols mounted on MN DD bins in $\mathcal{D}_0$ (information grid)

- Information grid/lattice



- Example: $\tau_{\mathrm{p}} = 50\mu$s, $\nu_{\mathrm{p}} = 20$kHz
  - $B = 10$ MHz
  - $\Delta\tau = \frac{1}{B} = 0.1\ \mu$s
  - $M = \frac{\tau_{\mathrm{p}}}{\Delta\tau} = \frac{50\mu\text{s}}{0.1\mu\text{s}} = 500$
  - $T = 1$ ms
  - $\Delta\nu = \frac{1}{T} = 1$ kHz
  - $N = \frac{\nu_{\mathrm{p}}}{\Delta\nu} = \frac{20\ \text{kHz}}{1\ \text{kHz}} = 20$
  - $B = M\nu_{\mathrm{p}}$, $T = N\tau_{\mathrm{p}}$, $BT = MN$

Zak-OTFS transceiver



- End-to-end I/O relation (continuous)

$$y_{\mathrm{dd}}^{w_{\mathrm{rx}}}(\tau,\nu) = \underbrace{w_{\mathrm{rx}}(\tau,\nu) *_\sigma h_{\mathrm{phy}}(\tau,\nu) *_\sigma w_{\mathrm{tx}}(\tau,\nu)}_{\triangleq\, h_{\mathrm{eff}}(\tau,\nu) \ \text{(effective channel)}} *_\sigma x_{\mathrm{dd}}(\tau,\nu) + \underbrace{w_{\mathrm{rx}}(\tau,\nu) *_\sigma n_{\mathrm{dd}}(\tau,\nu)}_{\triangleq\, n_{\mathrm{dd}}^{w_{\mathrm{rx}}}(\tau,\nu)}$$

- DD domain sampling on the information grid

$$y_{\mathrm{dd}}[k,l] = y_{\mathrm{dd}}^{w_{\mathrm{rx}}}\left(\tau = \frac{k\tau_p}{M}, \nu = \frac{l\nu_p}{N}\right), \quad k,l \in \mathbb{Z}$$

End-to-end I/O relation

- End-to-end I/O relation (discrete) [2]

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

- Vectorized form of end-to-end I/O relation [3]: $\mathbf{y} = \mathbf{H}_{\text{eff}}\mathbf{x} + \mathbf{n}$

  $\mathbf{H}_{\text{eff}} \in \mathbb{C}^{MN \times MN}$: effective channel matrix

$$\mathbf{H}_{\text{eff}}[k'N + l' + 1, kN + l + 1] = \sum_{m,n \in \mathbb{Z}} h_{\text{eff}}[k' - k - nM, l' - l - mN] e^{j2\pi nl/N} e^{j2\pi \frac{(l'-l-mN)(k+nM)}{MN}} \quad (1)$$

  $\mathbf{x}, \mathbf{y}, \mathbf{n} \in \mathbb{C}^{MN \times 1}$, $\mathbf{x}_{kN+l+1} = x_{\text{dd}}[k,l]$, $\mathbf{y}_{kN+l+1} = y_{\text{dd}}[k,l]$, $\mathbf{n}_{kN+l+1} = n_{\text{dd}}[k,l]$

- Closed-form expressions for $h_{\text{eff}}[k,l]$ and noise covariance
  - Derived for sinc and Gaussian filters in [4]

- Channel estimation problem: Estimation of $\mathbf{H}_{\text{eff}}$ matrix

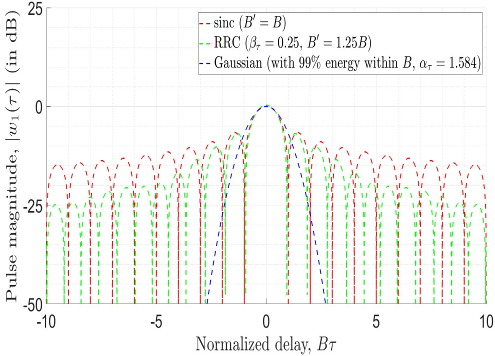
[2] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "OTFS — a mathematical foundation for communication and radar sensing in the delay-Doppler domain," *IEEE BITS The Inform. Theory Mag.*, vol. 2, no. 2, pp. 36-55, 1 Nov. 2022.

[3] 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," *IEEE BITS The Inform. Theory Mag.*, vol. 3, no. 2, pp. 7-31, Jun. 2023

[4] A. Das, F. Jesbin, and A. Chockalingam, "Closed-form expressions for I/O relation in Zak-OTFS with different delay-Doppler filters," *IEEE Trans. Veh. Tech.*, 2025. doi: 10.1109/TVT.2025.3564419.

Pulse shaping filters

- Sinc filter: $w_{\text{tx}}(\tau,\nu) = \sqrt{BT}\,\text{sinc}(B\tau)\,\text{sinc}(T\nu)$
- Gaussian filter[5]: $w_{\text{tx}}(\tau,\nu) = \left(\frac{2\alpha_\tau B^2}{\pi}\right)^{\frac{1}{4}} e^{-\alpha_\tau B^2\tau^2} \left(\frac{2\alpha_\nu T^2}{\pi}\right)^{\frac{1}{4}} e^{-\alpha_\nu T^2\nu^2}$



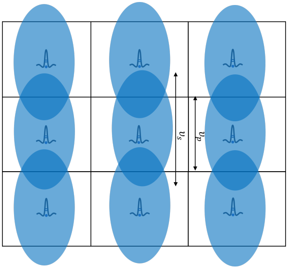
- Rx filter is matched to the Tx filter: $w_{\text{rx}}(\tau,\nu) = w^*_{\text{tx}}(-\tau,-\nu)e^{j2\pi\nu\tau}$

[5] 99% energy contained within bandwidth B and time T corresponds to $\alpha_\tau = \alpha_\nu = 1.584$.

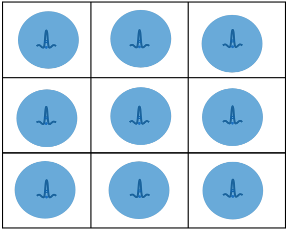
Choice of $(\tau_\mathrm{p}, \nu_\mathrm{p})$

- Crystallization condition: $\tau_{max} < \tau_\mathrm{p}$ and $\nu_{max} < \nu_\mathrm{p}$
    - τ_{max}: maximum delay spread of the effective channel
    - ν_{max}: maximum Doppler spread of the effective channel
- Choose τ_p and ν_p such that the crystallization condition is satisfied

- Non-crystalline regime (results in DD aliasing)

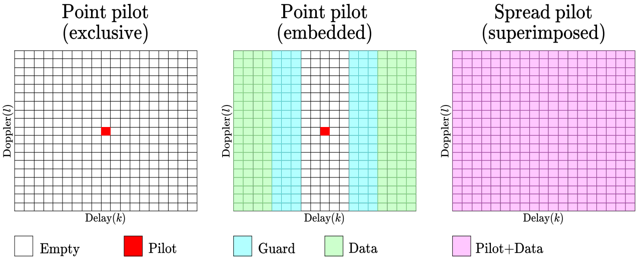


- Crystalline regime



DD domain channel estimation

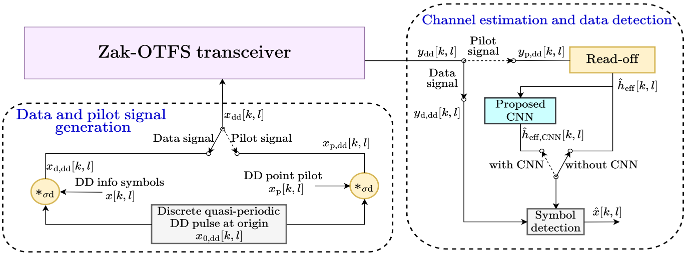
- Types of pilot frames



- Two approaches of effective channel estimation
  - Model-dependent approach:
    - Estimate underlying physical channel parameters to obtain $\hat{h}_{\text{eff}}[k,l]$, i.e., $\{\hat{\tau}_i,\ \hat{\nu}_i,\ \hat{h}_i\}\text{s} \rightarrow \hat{h}_{\text{phy}}(\tau,\nu) \rightarrow \hat{h}_{\text{eff}}(\tau,\nu) \rightarrow \hat{h}_{\text{eff}}[k,l] \rightarrow \mathbf{H}_{\text{eff}}$
  - Model-free approach: Direct read-off to obtain $\hat{h}_{\text{eff}}[k,l]$

$$\hat{h}_{\text{eff}}[k,l] = \begin{cases} y_{\text{p,dd}}\left[k+\frac{M}{2}, l+\frac{N}{2}\right] e^{-j\pi\frac{l}{N}}, & -\frac{M}{2} \le k < \frac{M}{2}, \\ & -\frac{N}{2} \le l < \frac{N}{2}, \\ 0, & \text{otherwise} \end{cases}$$

Channel estimation (Exclusive pilot)



- Transmission scheme
  - Send a pilot frame followed by data frames
  - Estimate $\hat{\mathbf{H}}_{\mathrm{eff}}$ during pilot frame and use it for symbol detection in data frames
- DD point pilot at $(k_{\mathrm{p}}, l_{\mathrm{p}})$: $x_{\mathrm{p}}[k,l] = \delta[k-k_{\mathrm{p}}]\delta[l-l_{\mathrm{p}}], \ (k_{\mathrm{p}}, l_{\mathrm{p}}) = (M/2, N/2)$
- Exclusive point pilot signal:

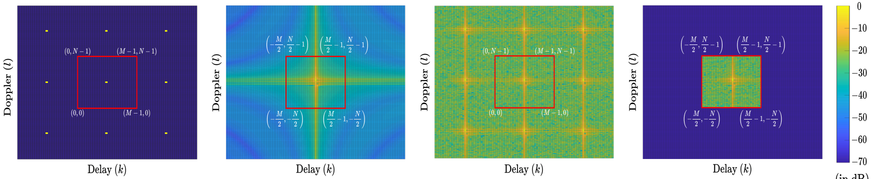
$$x_{\mathrm{p,dd}}[k,l] = \delta[k-k_{\mathrm{p}}]\delta[l-l_{\mathrm{p}}] *_{\sigma \mathrm{d}} x_{0,\mathrm{dd}}[k,l]$$

$$= \sum_{n,m\in\mathbb{Z}} \delta[k-k_{\mathrm{p}}-nM]\delta[l-l_{\mathrm{p}}-mN] e^{j2\pi \frac{nl_{\mathrm{p}}}{N}}, \ k,l \in \mathbb{Z}$$

- Data signal: $x_{\mathrm{d,dd}}[k,l] = x[k,l] *_{\sigma \mathrm{d}} x_{0,\mathrm{dd}}[k,l]$

  $x[k,l], 0 \le k \le M-1, \ 0 \le l \le N-1$ are the MN information symbols

Model-free channel estimation



(a) Point pilot signal $|x_{\mathrm{p,dd}}[k,l]|$ (in dB)

(b) True effective channel $|h_{\mathrm{eff}}[k,l]|$ (in dB)

(c) Received pilot signal $|y_{\mathrm{p,dd}}[k,l]|$ (in dB)

(d) Effective channel estimate $|\hat{h}_{\mathrm{eff}}[k,l]|$ (in dB)

- Received pilot signal

$$
\begin{aligned}
y_{\mathrm{p,dd}}[k,l] &= h_{\mathrm{eff}}[k,l] *_{\sigma \mathrm{d}} x_{\mathrm{p,dd}}[k,l] + n_{\mathrm{dd}}[k,l] \\
&= \underbrace{h_{\mathrm{eff}}[k-k_{\mathrm{p}}, l-l_{\mathrm{p}}]e^{j\pi\frac{\left(l-\frac{N}{2}\right)}{N}}}_{\text{Effective channel}} + \underbrace{n_{\mathrm{dd}}[k,l]}_{\text{Receiver noise}^{(i)}} \\
&\underbrace{+ \sum_{m,n\in\mathbb{Z},\ (m,n)\neq(0,0)} h_{\mathrm{eff}}[k-(k_{\mathrm{p}}+nM), l-(l_{\mathrm{p}}+mN)]e^{j2\pi\frac{nl_{\mathrm{p}}}{N}} e^{j2\pi\frac{(l-l_{\mathrm{p}}-mN)(k_{\mathrm{p}}+nM)}{MN}}}_{\text{DD aliasing}^{(ii)}}
\end{aligned}
$$

- Effective channel estimate read-off

$$
\hat{h}_{\mathrm{eff}}[k,l] = \begin{cases} y_{\mathrm{p,dd}}\left[k+\frac{M}{2}, l+\frac{N}{2}\right] e^{-j\pi\frac{l}{N}}, & -\frac{M}{2} \le k < \frac{M}{2}^{(iii)}, \\ & -\frac{N}{2} \le l < \frac{N}{2}, \\ 0, & \text{otherwise} \end{cases}
$$

Model-free channel estimation

- Advantages
  1. Simple
  2. Natural and effective in acquiring fractional DDs
- Drawbacks
  1. Read-off provides an estimate only over a limited region ($\mathcal{F}$) in the DD plane
     - Does not provide the estimate for the region outside ($\mathcal{F}^c$)
     - This affects estimation performance depending on the pulse shaping characteristics of the filter used
     - A poorly localized pulse shape results in increased degradation
  2. The read-off samples are corrupted by the DD aliases and receiver noise
- Learning approach to address the drawbacks
  - Treat the read-off samples as an DD image
  - Use learning techniques to enhance the quality of this 'image'

Channel images

- True effective channel image (ground truth)

$$\mathbf{H}_{\mathrm{I}}[k', l'] = h_{\text{eff}}[k' - (n'+1)M, l' - (m'+1)N],$$

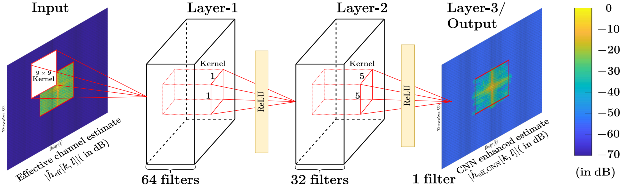
$k' = 0, \ldots, 2(n'+1)M$, $l' = 0, \ldots, 2(m'+1)M$, where n, m in (1) are $n \in [-n', n']$, $m \in [-m', m']$
$m' = n' = 2$ is adequate to consider dominant terms in the sum in (1)

- Effective channel estimate image

$$\hat{\mathbf{H}}_{\mathrm{I}}[k', l'] = \begin{cases} \hat{h}_{\text{eff}}[k' - (n'+1)M, l' - (m'+1)N], \\ \qquad \text{for } -\frac{M}{2} \leq k' - (n'+1)M < \frac{M}{2}, \\ \qquad\quad -\frac{N}{2} \leq l' - (m'+1)N < \frac{N}{2}, \\ 0, \quad \text{otherwise} \end{cases}$$

$k' = 0, \ldots, 2(n'+1)M$, $l' = 0, \ldots, 2(m'+1)M$

- $\hat{\mathbf{H}}_{\mathrm{I}}$ complex-valued $\rightarrow \Re(\hat{\mathbf{H}}_{\mathrm{I}})$ and $\Im(\hat{\mathbf{H}}_{\mathrm{I}})$ serves as independent real-valued inputs to the same network



- **CNN architecture:** Three layer hierarchical network[6]
  - First layer: 64 filters with 9×9 kernel $\rightarrow$ ReLU activation
  - Second layer: 32 filters with 1×1 kernel $\rightarrow$ ReLU activation
  - Third layer: Single 5×5 filter
- $\hat{\mathbf{H}}_{\mathrm{I,CNN}}$: Enhanced effective channel image
- Enhanced channel estimate

$$\hat{h}_{\mathrm{eff,CNN}}[k,l] = \mathbf{H}_{\mathrm{I}}[k + (n'+1)M, l + (m'+1)N],$$

$$k = -(n'+1)M, \ldots, (n'+1)M, \; l = -(m'+1)M, \ldots, (m'+1)M$$

[6] C. Dong, C. C. Loy, K. He and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. and Mach. Intel.*, vol. 38, no. 2, pp. 295-307, Feb. 2016.

CNN parameters

Parameters	Values
Training pilot SNR (in dB)	15
Training data size	10000
Testing data size	2000
Batch size	128
Number of epochs	1000
Learning rate	0.001
Optimizer	Adam
Total trainable CNN parameters	8129
Stride	1
Padding	same (input size preserved)

- **Training methodology:** Optimize loss function to minimize MSE

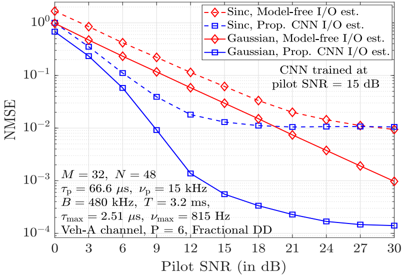
$$\mathcal{L}(\Theta) = \frac{1}{N_s} \sum_{i=0}^{N_s-1} \Bigg(\left\| f_{\mathrm{CNN}}\left(\Theta; \Re(\hat{\mathbf{H}}_{\mathrm{I}}^{(i)})\right) - \Re(\mathbf{H}_{\mathrm{I}}^{(i)}) \right\|_F^2 + \left\| f_{\mathrm{CNN}}\left(\Theta; \Im(\hat{\mathbf{H}}_{\mathrm{I}}^{(i)})\right) - \Im(\mathbf{H}_{\mathrm{I}}^{(i)}) \right\|_F^2 \Bigg)$$

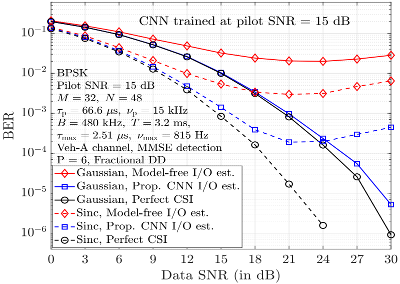
  - Pair $(\hat{\mathbf{H}}_{\mathrm{I}}^{(i)}, \mathbf{H}_{\mathrm{I}}^{(i)})$: ith realization of training data set
  - $f_{\mathrm{CNN}}(.)$: CNN function
  - Θ: Trainable network parameters
- Used PyTorch ML libraries and Nvidia RTX 3090 GPU

Zak-OTFS system parameters

Parameter	Values
Channel type	Vehicular-A
Relative power (dB)	0.0, -1.0, -9.0, -10.0, 15.0, -20.0
Relative delay (μs)	0, 0.31, 0.71, 1.09, 1.73, 2.51
Bandwidth (B)	480 kHz
Time duration (T)	3.2 ms
Maximum Doppler spread ($\nu_{\max}$)	815 Hz
Maximum delay spread ($\tau_{\max}$)	2.51 μs
Delay period (τ_{p})	66.6 μs
Doppler period (ν_{p})	15 kHz
No. of delay bins (M)	32
No. of Doppler bins (N)	48
Symbol detection	MMSE detection

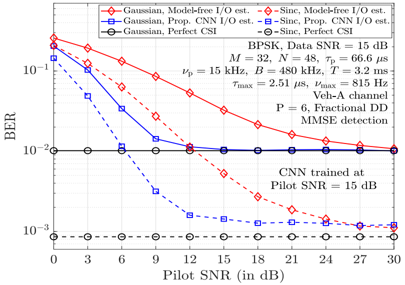
CNN enhanced NMSE and BER performance

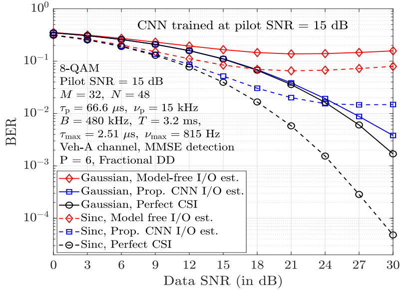




- NMSE vs Pilot SNR:
  - Improved NMSE performance across pilot SNR values for both Gaussian ans sinc filters, despite being trained at single pilot SNR of 15 dB
  - Gaussian: high DD localization $\rightarrow$ Lower NMSE than sinc
- BER vs Data SNR (BPSK):
  - Improved BER performance for both Gaussian (close to perfect CSI) and sinc filters
  - Sinc: nulls at information grid points $\rightarrow$ Lower BER than Gaussian

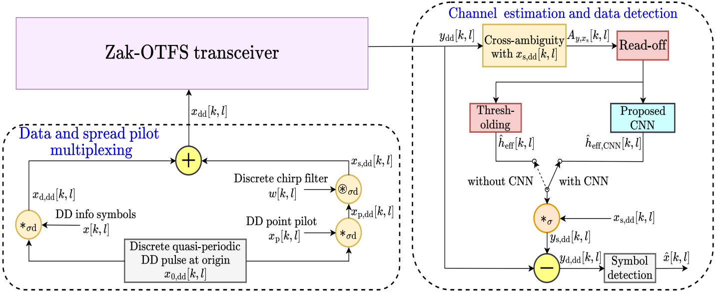
CNN enhanced BER performance





- BER vs Pilot SNR (BPSK):
  - Outperforms conventional model-free method for both Gaussian ans sinc filters
  - For both filters, achieves BER performance close to the perfect CSI at low pilot SNRs
- Improved BER performance with 8-QAM for both Gaussian (close to perfect CSI) and sinc filters across data SNRs

Zak-OTFS with superimposed spread-pilot



- Data and spread-pilot are superimposed on the same frame[7]
- Advantages: No throughput loss due to pilot. Better PAPR
- Spreading filter[8] $w[k,l]$ applied to the point pilot to obtain spread pilot

$$x_{\mathrm{s,dd}}[k,l] = w[k,l] \circledast_{\sigma d} x_{\mathrm{p,dd}}[k,l]$$

- Data and pilot multiplexed together for transmission

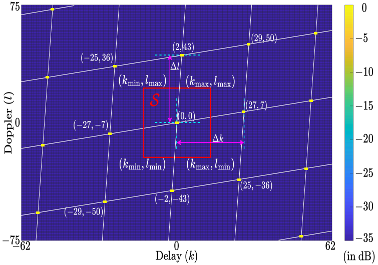
$$x_{\mathrm{dd}}[k,l] = \sqrt{E_{\mathrm{d}}}\, x_{\mathrm{d,dd}}[k,l] + \sqrt{E_{\mathrm{p}}}\, x_{\mathrm{s,dd}}[k,l]$$

[7] M. Ubadah, S. K. Mohammed, R. Hadani, S. Kons, A. Chockalingam, and R. Calderbank, *Zak-OTFS for integration of sensing and communication*, online arxiv.org/abs/2404.04182, 5 Apr 2024.

[8] Chirp filter $w[k,l] = \frac{1}{MN} e^{j2\pi \frac{q(k^2+l^2)}{MN}}$, q: slope-parameter

Channel estimation (superimposed spread pilot)

- Basic idea
  - Exploit the nature of the self ambiguity[9] of the spread pilot $x_{\mathrm{s,dd}}[k,l]$ for channel estimation
  - Self ambiguity $A_{x_\mathrm{s},x_\mathrm{s}}[k,l]$ of the spread pilot is supported on a twisted lattice Λ_q (w.r.t. the period lattice Λ_p)
  - $A_{x_\mathrm{s},x_\mathrm{s}}[k,l]$:



- M, N $\rightarrow$ odd primes
- q $\rightarrow$ relative prime to M and N
- e.g., $M = 31, N = 37, q = 3$

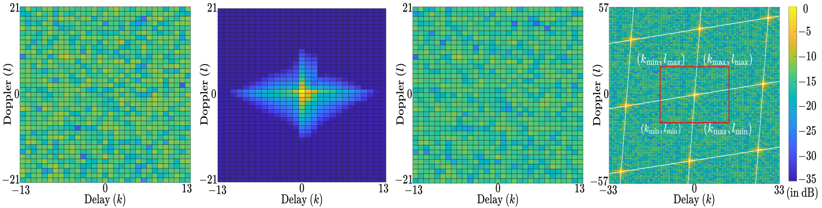
  - The cross-ambiguity $A_{y,x_\mathrm{s}}[k,l]$ between $y_{\mathrm{dd}}[k,l]$ and $x_{\mathrm{s,dd}}[k,l]$ has the effective channel supported on the self-ambiguity lattice Λ_q

[9] The cross-ambiguity function between two discrete DD domain signals $a[k,l]$ and $b[k,l]$ is given by $A_{a,b}[k,l] = \sum_{k'=0}^{M-1}\sum_{l'=0}^{N-1} a[k',l']b^*[k'-k,l'-l]e^{-j2\pi\frac{l(k'-k)}{MN}}$

Channel estimation (superimposed spread pilot)

- Procedure
  - Compute $A_{y,x_s}[k,l]$ (cross ambiguity between the received signal and the spread pilot signal)
  - Cross-ambiguity output has the effective channel supported on the self-ambiguity lattice Λ_{q}
  - Simply read off the cross-ambiguity output in the support set $\mathcal{S}$
  - Use the read-off samples as $\hat{h}_{\mathrm{eff}}[k,l]$ values to construct the $\hat{H}_{\mathrm{eff}}$ matrix
- Issue
  - Cross-ambiguity output has data interference, DD aliases, and noise
  - This compromises estimation quality
- Solution approach to address the issue
  - Treat the cross-ambiguity read-off as a corrupted 'DD image'
  - Use learning techniques to enhance this 'image'

Channel estimation (superimposed spread pilot)



(a) Transmitted signal $|x_{\mathrm{dd}}[k,l]|$ (in dB) (b) True effective channel $|h_{\mathrm{eff}}[k,l]|$ (in dB) (c) Received signal $|y_{\mathrm{dd}}[k,l]|$ (in dB) (d) Cross-ambiguity between received signal and spread pilot $|A_{y,x_s}[k,l]|/\sqrt{E_p}$ (in dB)

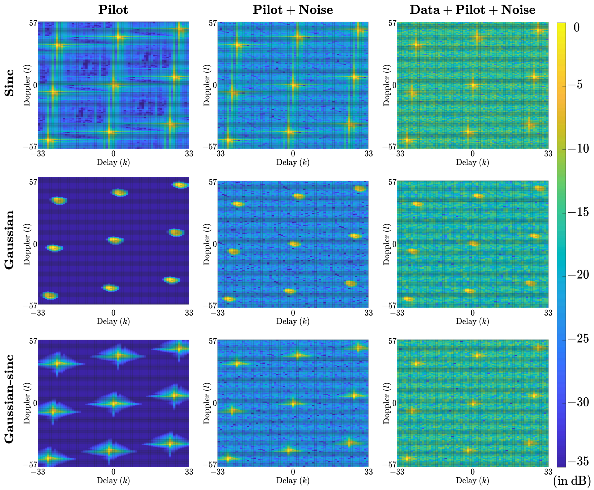
- $$A_{y,x_{\mathrm{s}}}[k,l] = \underbrace{\sqrt{E_{\mathrm{p}}}\; h_{\mathrm{eff}}[k,l]}_{\text{Effective channel}} + \underbrace{\sqrt{E_{\mathrm{p}}} \sum_{(k_i,l_i)\in\Lambda_q,(k_i,l_i)\neq(0,0)} e^{j\theta_i} h_{\mathrm{eff}}[k-k_i, l-l_i] e^{j2\pi\frac{(l-l_i)k_i}{MN}}}_{\text{DD aliasing}}$$
$$+ \underbrace{\sqrt{E_{\mathrm{d}}}\; h_{\mathrm{eff}}[k,l] *_{\sigma\mathrm{d}} A_{x_{\mathrm{d}},x_{\mathrm{s}}}[k,l]}_{\text{Data interference}} + \underbrace{A_{n,x_{\mathrm{s}}}[k,l]}_{\text{Receiver noise}}$$

- Effective channel estimated reading off within support set $\mathcal{S}$

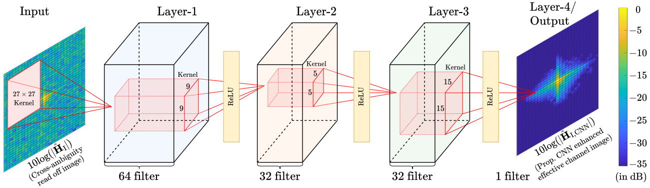
$$\hat{h}_{\mathrm{eff}}[k,l] = \begin{cases} A_{y,x_{\mathrm{s}}}[k,l]/\sqrt{E_{\mathrm{p}}}, & \text{for } (k,l)\in\mathcal{S} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Channel estimation (superimposed spread-pilot)

- Cross ambiguity output $A_{y,x_s}[k, l]$ for different filters



CNN framework



- **CNN architecture:** Four layer hierarchical network
  - First layer: 64 filters with 27 × 27 kernel, ReLU activation
  - Second layer: 32 filters with 9 × 9 kernel, ReLU activation
  - Third layer: 32 filters with 5 × 5 kernel, ReLU activation
  - Fourth layer: Single 15 × 15 filter, linear activation
- $\mathbf{H}_I$: True effective channel image (i.e., ground truth)
- $\hat{\mathbf{H}}_I$: Training image constructed from cross-ambiguity
- $\hat{\mathbf{H}}_{I,\text{CNN}}$: Output of the CNN network
- Enhanced channel estimate

$$\hat{h}_{\text{eff,CNN}}[k,l]=\begin{cases}\hat{\mathbf{H}}_{I,\text{CNN}}\left[k-k_{\min}, l-l_{\min}\right], \\ \qquad k_{\min}\le k\le k_{\max}, l_{\min}\le l\le l_{\max} \\ 0, \quad \text{otherwise}\end{cases}$$

CNN parameters

Parameters	Values
Training data SNR (in dB)	15
Training PDR (in dB)	0, 5, 20, 25, 30, 35
Training data size	600000 (100000 per PDR value)
Batch size	64
Number of epochs	50
Learning rate	0.0005
Total trainable CNN parameters	245473
Stride	1
Padding	same (input size preserved)

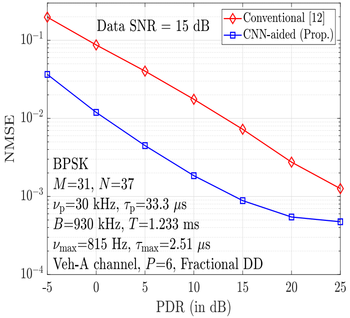
- **Training methodology:** Optimize loss function to minimize NMSE

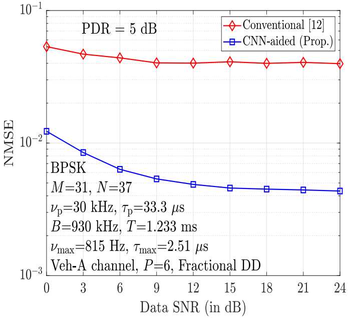
$$\mathcal{L}(\Theta) = \frac{1}{N_s}\sum_{i=0}^{N_s-1}\left(\frac{\left\|f_{\mathrm{CNN}}\left(\Theta; \Re(\hat{\mathbf{H}}_{\mathrm{l}}^{(i)})\right) - \Re(\mathbf{H}_{\mathrm{l}}^{(i)})\right\|_F^2}{\left\|\Re(\mathbf{H}_{\mathrm{l}}^{(i)})\right\|_F^2} + \frac{\left\|f_{\mathrm{CNN}}\left(\Theta; \Im(\hat{\mathbf{H}}_{\mathrm{l}}^{(i)})\right) - \Im(\mathbf{H}_{\mathrm{l}}^{(i)})\right\|_F^2}{\left\|\Im(\mathbf{H}_{\mathrm{l}}^{(i)})\right\|_F^2}\right)$$

  - $f_{\mathrm{CNN}}(.)$: CNN function
  - Θ: Trainable network parameters
  - $||.||_{\mathrm{F}}$: Frobenius norm

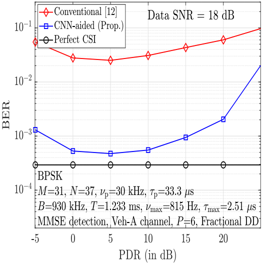
- ADAM optimizer with dynamically adjusted learning rate used

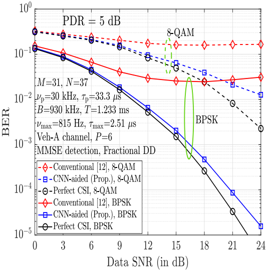
NMSE performance

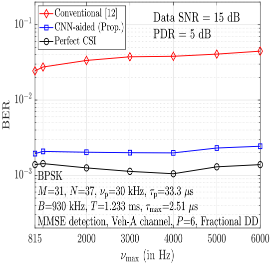




- Almost an order in magnitude improvement in NMSE with the CNN-aided estimation method
- NMSE performance better across all data SNRs, despite being trained only at 15 dB







- U-shaped BER vs PDR curve
  - low PDR → low pilot SNR → higher NMSE → higher BER
  - high PDR → high pilot SNR → high interference to data → higher BER
- CNN trained at various $\nu_{\max}$ values, at 5 dB PDR and 15 dB data SNR → good performance for different Dopplers

Summary

- Zak-OTFS is a promising waveform for communication and radar sensing
- Learning techniques can be exploited for improved transceiver design
- Lot of scope of further research

Summary

- Zak-OTFS is a promising waveform for communication and radar sensing
- Learning techniques can be exploited for improved transceiver design
- Lot of scope of further research

Thank you