Practical Implementation of Zak-OTFS at WINLABS
Towards 6G waveform design

K. Venkatesh¹, S. Ramachandran²

¹Duke University
²Rutgers University

Supervised by:--
Dr. R. Calderbank, Dr. N. B. Mandayam
Dr. I. Seskar, Dr. S. K. Mohammed

Assisted by:--
M. Prasanthi

December 4, 2023
Table of Contents

1 Background

2 Zak-OTFS Implementation at WINLABS
1 Background

2 Zak-OTFS Implementation at WINLABS
Information symbols are multiplexed in the frequency domain
Information symbols are multiplexed in the frequency domain

1-D transform in the frequency domain (IFFT/FFT)
OFDM

- Information symbols are multiplexed in the frequency domain
- 1-D transform in the frequency domain (IFFT/FFT)
- Orthogonality among the sub-carriers is the key
OFDM

- Information symbols are multiplexed in the frequency domain
- 1-D transform in the frequency domain (IFFT/FFT)
- Orthogonality among the sub-carriers is the key
- High Doppler channels causes ICI
Information symbols are multiplexed in the delay Doppler (DD) domain
Information symbols are multiplexed in the delay Doppler (DD) domain.

Performs good in doubly dispersive channels.
LTV multipath channels can be represented as a function of
- Time-Frequency, $H(t, f)$,
LTV multipath channels can be represented as a function of

- Time-Frequency, $H(t, f)$,
- Time-Delay, $g(t, \tau)$,
Channel representation

LTV multipath channels can be represented as a function of

- Time-Frequency, $H(t, f)$,
- Time-Delay, $g(t, \tau)$,
- Delay-Doppler, $h(\tau, \nu)$
LTV multipath channels can be represented as a function of

- Time-Frequency, $H(t, f)$,
- Time-Delay, $g(t, \tau)$,
- Delay-Doppler, $h(\tau, \nu)$
LTV multipath channels can be represented as a function of
- Time-Frequency, $H(t, f)$,
- Time-Delay, $g(t, \tau)$,
- Delay-Doppler, $h(\tau, \nu)$
OTFS Key features

- The channel response in the DD domain is invariant for a larger observation time
OTFS Key features

- The channel response in the DD domain is invariant for a larger observation time
- Each symbol experiences nearly constant channel gain
OTFS Key features

- The channel response in the DD domain is invariant for a larger observation time
- Each symbol experiences nearly constant channel gain
- Channel interaction with transmit symbols is 2-D convolution rather than multiplication
Table of Contents

1. Background

2. Zak-OTFS Implementation at WINLABS
One-step conversion from DD to time and time to DD domains
Zak-OTFS/OTFS 2.0

- One-step conversion from DD to time and time to DD domains
- More robust to large channel spreads compared to OTFS 1.0
Zak-OTFS/OTFS 2.0

- One-step conversion from DD to time and time to DD domains
- More robust to large channel spreads compared to OTFS 1.0
- When operating in a crystalline regime, the predictability of the I/O relation is simple.
Design Flow

1. Zak-OTFS baseband Modulation
   - Designed in MATLAB

2. Up-convert and Transmit
   - Taken care by SDR/USRP

3. Channel
   - Can be wired or wireless (based on our experiment)

4. Receive and Down-convert
   - Taken care by SDR/USRP

5. Zak-OTFS baseband Demodulation
   - Designed in MATLAB
The information symbols $a[0], a[1], \cdots, a[MN - 1]$ are multiplexed in DD domain such that $x[k, l] = a[k + Ml]$, where $k = 0, 1, \cdots, M - 1$ and $l = 0, 1, \cdots, N - 1$. 

![Diagram of Zak-OTFS baseband modulation process](image)
Zak-OTFS baseband de-modulation

\[ r_{td}^{\text{real}}[n] \rightarrow r_{td}[n] \rightarrow r_{td}(t) \rightarrow y_{dd}[\tau, \nu] \rightarrow \text{Twisted Convolution} \rightarrow x_{dd}^{w_{rx}}[\tau, \nu] \rightarrow \text{DD domain Sampling} \rightarrow y_{dd}[k, l] \]
Implementation

Objective - 1

Zak-OTFS baseband Modulation

Designed in MATLAB

Wireless (modeled)

Zak-OTFS baseband Demodulation

Designed in MATLAB

Objective - 2

Data File

Up-convert and Transmit

Taken care by SDR/USRP

Received and Down-convert

Taken care by SDR/USRP

Data File

Channel

Wired (SB1)
Zak-OTFS Transmit signal with Point Data (Theory)

\[ R\{x(t)\} \]

\[ \cos(2\pi v_0 (t - \tau_0)) \]

\[ Z_{\tau}^{-1}: x_{dd}(\tau, v) \rightarrow x(t) \]

\[ x(t) = \sqrt{\tau_p} \int_0^{\nu_p} x_{dd}(\tau, v) \, dv \]

\[ \nu_p = \frac{1}{\tau_p} \]
Zak-OTFS Transmit signal with Point Data (Practical)
DD spreading function

- DD spreading function, $h(\tau, \nu) = \sum_{i=1}^{P} h_i \delta(\tau - \tau_i) \delta(\nu - \nu_i)$
- Received signal, $y(t) = \sum_{i=1}^{P} h_i x(t - \tau_i) e^{j2\pi\nu_i(t-\tau_i)}$
MATLAB Experiment with DD spreading function

**DD Spreading Function**

\[ h(\tau, \nu) = \sum_{i=1}^{p} h_i \delta(\tau - \tau_i)\delta(\nu - \nu_i) \]

at \( \tau = \frac{k \tau_p}{M} \), \( \nu = \frac{l \nu_p}{N} \)

**DD Received signal**

\[ y[k', l'] = \sum_{k,l \in \mathbb{Z}} h_{dd}[k' - k, l' - l] x[k, l] e^{j2\pi(l' - l)k/MN} \]

where \( h_{dd}[k, l] = w_{rx}[k, l] \ast_{\sigma} h[k, l] \ast_{\sigma} w_{tx}[k, l] \)

---

DD input signal

DD channel

DD Received signal
Project Goals

Stage-1 of Implementation
We use MATLAB to generate Zak-OTFS transmit samples and transmit them using SB1 (ORBIT), where the medium of transmission is a wire. The received samples are processed again in MATLAB.

Stage-2 of Implementation
Signal Processing is same as that of Stage-1, but the medium of transmission is wireless. We use MIMO facility in WINLABS to perform this experiment.

Stage-3 of Implementation
Signal Processing and channel is same as that of Stage-1. But we use SB4 (ORBIT), where the medium of transmission is a wire with a programmable attenuator. Using the results at this stage, we will produce an SNR versus BER curve.

Stage-4 of Implementation
Signal Processing and channel is same as that of Stage-1. But we use Grid (ORBIT) or MIMO facilities, where the medium of transmission is wireless but the transmit and receive antennas are very far apart.

Stage-5 of Implementation
We need to convert MATLAB code to C/C++ or some code that is compatible with GNU Radio, where there is a lot of scope to add additional blocks so as to design Zak-OTFS-based real-time transmission and reception.

Stage-6 of Implementation
We will be transmitting and receiving the signal over mm-wave frequencies.

Note: For these 5 stages of implementation, we will be using sub-6 GHz frequencies (around 2.4 GHz) to transmit and receive signals.


THANK YOU

QUESTIONS & SUGGESTIONS