2024

OTFS Modulation for High Mobility Wireless Channels

Sandip K C
Minnesota State University, Mankato

OTFS Modulation for High Mobility Wireless Channels

By

Sandip K C

A Thesis Submitted in Partial Fulfillment of the Requirements for
Master of Science
In
Electrical Engineering

Minnesota State University
Mankato, Minnesota

July 2024

Date:

Title: **OTFS modulation for high mobility wireless channels**

Student name: **Sandip K C**

This thesis paper has been examined and approved.

Examining Committee:

Dr. Yikai Li, Advisor

Dr. Xuanhui Wu, Committee Member

Dr. Nannan He, Committee Member

Acknowledgements

I sincerely thank Dr. Yikai Li, my professor and advisor. His invaluable guidance and unwavering support have been the cornerstone of my thesis journey. His insights and expertise significantly shaped my work. I am also immensely thankful to Dr. Xuanhui Wu, our respected department chair. His encouragement and support for my thesis work have been instrumental in my pursuit of academic excellence. Furthermore, I sincerely thank Dr. Vincent (Qun) Zhang for the impact on my personal and academic growth. I am deeply grateful to these exceptional individuals for their guidance, support, and mentorship.

Contents

Abstract

OFDM Modulation for High Mobility Wireless Channels

Sandip K C
Minnesota State University, Mankato
2024

In today's world, billions of devices are being connected to the internet primarily through wireless means, making this connectivity increasingly important. Managing these connections becomes complex, especially for moving devices such as drones and high-speed trains, which require seamless and reliable connectivity even at high speeds. Maintaining a stable connection in fast-moving environments is challenging but essential for the operation of these technologies. Traditional communication systems like OFDM face significant challenges in high mobility environments, struggling to cope with the rapid changes and dynamic conditions that high-speed movements introduce. To address this, OTFS is proposed as a solution. This novel modulation framework operates in the delay-Doppler domain, allowing a two-dimensional representation of the channel's behavior over time and frequency. In this paper, various simulations of OTFS and OFDM were conducted under the same parameters and conditions. The results strongly advocate for the implementation of OTFS as a modulation scheme in future wireless communication systems, demonstrating superior performance and reliability compared to traditional OFDM, and highlight OTFS's potential to satisfy the growing need for high-speed, dependable communication in the rapidly advancing wireless technology landscape.

List of Acronyms

OTFS: Orthogonal Time Frequency Space

SMS: Short Message Service

TF: Time-Frequency

ZT: Zak Transform

QAM: Quadrature Amplitude Modulation

TD: Time Delay

CDMA: Code Division Multiple Access

DD: Delay-Doppler

OFDM: Orthogonal Frequency Division Multiplexing

DFT: Discrete Fourier Transform

LTI: Linear Time Invariant

UMTS: Universal Mobile Telecommunications System

S/P: Serial to Parallel

IFT: Inverse Fourier Transform

DZT: Discrete Zak Transform

SFFT: Symplectic Finite Fourier Transform

LoS: Line of Sight

ISI: Inter-Symbol Interference

IFFT: Inverse Fast Fourier Transform

ICI: Inter-Carrier Interference

LTV: Linear Time Variant

ISFFT: Inverse Symplectic Finite Fourier Transform

FT: Fourier Transform

AMPS: Advanced Mobile Phone Systems

IDFT: Inverse Discrete Fourier Transform

FDMA: Frequency Division Multiple Access

MIMO: Multiple Input Multiple Output

P/S: Parallel to Serial

FFT: Fast Fourier Transform

BPSK: Binary Phase Shift Keying

GSM: Global System for Mobile Communication

Chapter 1

Introduction

1.1 Historical overview and development of mobile communication systems

The evolution of mobile communication systems showcases the advancement of technology, which is evident through generational changes. Each new generation brings about improvements in technology and service offerings, meeting the increasing need for data speeds, enhanced reliability, and broader network coverage.

The inception of mobile communications was marked by the 1G systems in the 1980s, primarily focusing on analog voice services. It is coined AMPS in North America, which used analog FM. The introduction of GSM in 2G facilitated the development of data compression, error detection, and error correction techniques in digital communication, even in the presence of noises in the communication channel. It ushered in features like SMS and basic data services, laying the groundwork for more advanced applications.

In the early 2000s, the demand for higher data rates to facilitate internet access and multimedia services led to the development of 3G technology. Standards like UMTS and CDMA2000, mobile internet became a reality. The arrival of 3G marked a significant increase in data rates due to its use of wideband CDMA's spread spectrum technology [1]. This represented a substantial improvement over 2G, which relied on narrowband signaling. The leap to 4G in the 2010s, with technologies like LTE, brought unprecedented improvements in data speeds and network capacity, enabling services like HD video streaming and high-speed internet access. The OFDM signaling technique used in 4G allows for the simultaneous transmission of several information symbols by overlapping them in time while preserving orthogonality in the

frequency domain. This is achieved by allocating these symbols to precisely spaced subcarriers. This is realized by mapping these symbols onto a grid of precisely spaced subcarrier frequencies.

The ongoing rollout of 5G represents a paradigm shift, aiming not only to enhance mobile broadband experiences but also to enable new use cases like massive IoT (Internet of Things) and critical communications. 5G is built to offer faster data speeds, decreased latency, and increased capacity by utilizing cutting-edge technologies such as massive MIMO and network slicing.

As we look towards 6G, the demands for connectivity in highly dynamic environments (like high-speed trains and autonomous vehicles) are expected to rise. This is where OTFS modulation stands out. OTFS modulation excels by presenting a new method to overcome challenges faced by current modulation techniques in high-mobility environments. By effectively leveraging DD domain, OTFS aims to become a pivotal element in evolving future wireless communications.

This evolutionary path reflects a continuous quest for higher speeds, efficiency, and adaptability in mobile communications, setting the stage for OTFS as an innovative solution in the ever-growing and demanding landscape of wireless technology.

1.2 High mobility wireless channels

The environment within a stationary wireless channel remains static as neither the transmitting nor receiving devices move. The signal transmitted from the antenna encounters multiple reflections from surrounding objects, arriving at the receiver via various paths. Receivers must

detect the original transmitted information from the composite signals, which may experience fading due to the incoherent addition caused by varying path lengths.

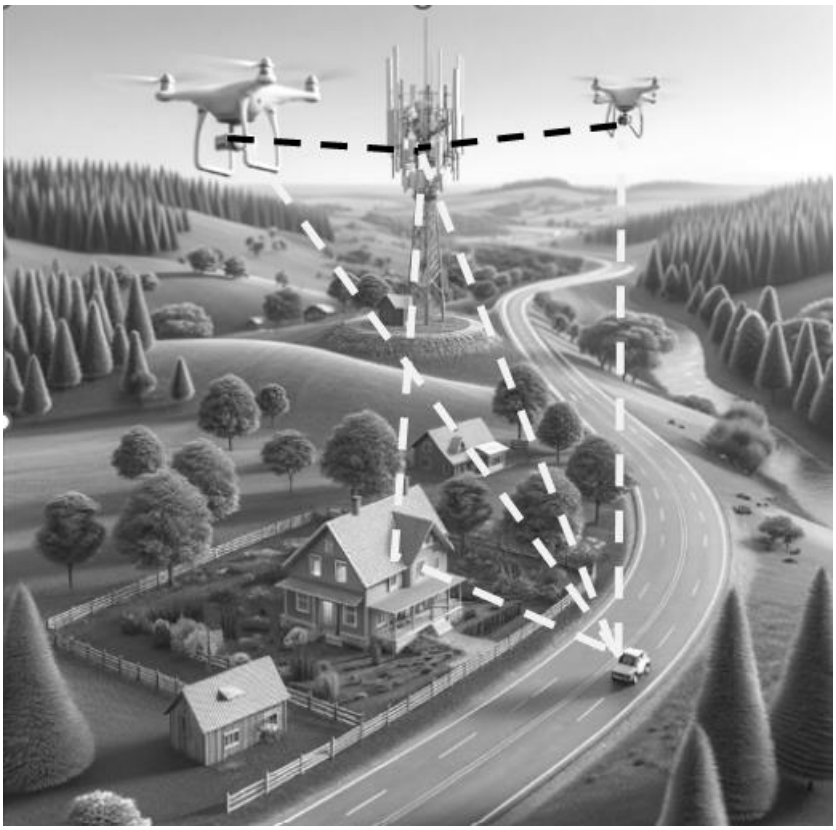


Figure 1.1: Illustration of high-mobility channel

With the advancing applications of wireless networks, scenarios involving high mobility are becoming increasingly common. Consider the rise of high-speed trains, autonomous vehicles, and planes, all of which travel at high velocities demanding high data rates. In such a high-mobility wireless channel, transmitters, receivers, and multiple reflective objects are all in motion, each moving at different velocities and directions, demanding high data rates. The signal in each path undergoes distinct Doppler shifts f_d relative to the carrier frequency f_c with $e^{j2\pi f_d t}$ [1] composite of nonlinearly altered due to these multiple Doppler shifts and time delays.

1.3 Waveforms for high-mobility wireless channels

Let $s(t)$ be the transmitted signal. It consists of the sequence a_n, which comprises data symbols drawn from a set A and is combined over time or frequency. Then the $s(t)$ is expressed as,

$$s(t) = \sum_n a_n \psi_n(t) \tag{1.1}$$

Let $\psi_{TX} = \{\psi_n(t)\}$ denote a set of orthonormal signals such that

$$\langle \psi_n(t), \psi_m(t) \rangle = \int \psi_n(t) \psi_m(t)^* dt = \delta_{m,n} = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases} \tag{1.2}$$

Here, the operator $(\cdot)^*$ signifies the complex conjugate. Considering the need for clarity, in the subsequent discussion, we will omit the noise component that is usually present in the system model [1].

Assuming the transmission of $s(t)$ occurs over an ideal linear channel that does not introduce distortion, the received signal $r(t)$ is given by attenuating $s(t)$ by g_0 and delaying by t_0 denoted as $r(t) = g_0 s(t - t_0)$. For the purpose of this discussion, and without loss of generality, $g_0 = 1$ and $t_0 = 0$, leading to $r(t) = s(t)$[1]. The channel's linearity ensures that all the basis signals $\psi_n(t)$ remain unchanged upon reception. The set of basis signals received will be denoted as ψ_{RX}.

To detect an information symbol a_{n_0}, the received signal s(t) is be projected onto its corresponding basis signal $\psi_{n_0}(t)$. This basis signal was used for multiplexing a_{n_0}.

$$\langle s(t), \psi_{n_0}(t) \rangle = \sum_n a_n \langle \psi_n(t), \psi_{n_0}(t) \rangle = a_{n_0} \tag{1.3}$$

This method is effective due to the channel's linearity and the orthogonal nature of the basis functions. With $\psi_{RX} = \psi_{TX}$, it indicates that the received basis signals are identical to the transmitted ones.

Consider an LTI channel characterized by $h(t)$ as the impulse response. The basis functions s(t) transmitted through this LTI channel are convoluted with $h(t)$, resulting in $\phi_n(t) = h(t) * \psi_n(t)$, thus experiencing distortion. Since the channel is linear, the signal received at the receiver is,

$$r(t) = \sum_n a_n \phi_n(t) \tag{1.4}$$

The set of received basis signals $\psi_{RX} = \{\phi_n(t)\}$ is no longer orthonormal due to the convolution with h(t), meaning that projecting r(t) onto $\psi_{n_0}(t)$ will not directly recover symbol a_{n_0} yielding,

$$r_{n_0} = \langle s(t), \psi_{n_0}(t) \rangle = a_{n_0} \langle \phi_{n_0}(t), \psi_{n_0}(t) \rangle + \sum_{n \neq n_0} a_n \langle \phi_n(t), \psi_{n_0}(t) \rangle \tag{1.5}$$

This implies that the sample r_{n_0} not only contains the desired symbol a_{n_0}, modified because of the angle's cosine between $\phi_{n_0}(t)$ and $\psi_{n_0}(t)$. This modification occurs because the projection of one basis function onto another involves a cosine factor, representing the alignment between the two functions. It also includes interference from all other symbols a_n giving rise to ISI. This ISI on the received signal is removed using a channel equalizer. For channels causing minimal distortion, the received symbol $r_{n_0} \approx a_{n_0}$, suggests that interactions between elements of ψ_{TX} and ψ_{RX} are nearly null for any pair n_0 and $n(n \neq n_0)$, while the scalar product $\langle \phi_{n_0}(t), \psi_{n_0}(t) \approx 1 \rangle$. Under these conditions, ISI can be considered minor additive noise, allowing for symbol-by-symbol detection to extract a_{n_0} from r_{n_0}. Conversely,

when severe distortions cause significant ISI, affecting many symbols a_n for $n \neq n_0$, treating ISI as small additional noise is insufficient. Advanced equalization methods, like maximum likelihood sequence estimation, must be deployed. This approach evaluates all potential symbol combinations to identify the most probable transmitted symbol a_{n_0}. Here, a_n contributes to ISI. As the number of these symbols increases, the complexity of these equalizers grows exponentially.

At the receiver side, the ISI component vanishes if it adheres to the following biorthogonal properties $\langle \phi_n(t), \psi_m(t) \rangle = \delta_{n,m}$. So when the ISI component is significantly less than 1, ψ_{TX} and ψ_{RX} become quasi-biorthogonal, which is precisely the role of an equalizer: to reestablish biorthogonality, or at least quasi-biorthogonality, between ψ_{TX} and ψ_{RX}.

The orthonormality property of ψ_{TX}, according to Parseval's identity, can also be represented using frequency domain signals. These signals are best described by the Fourier transforms $\hat{\psi}_n(f) = F\{\psi_n(t)\}$ for all n. Consequently, the ISI component and the cosine component remains the same. Their values do not change even when working with frequency domain signals.

$$r_{n_0} = a_{n_0} \cos\left(\angle\left(\hat{\phi}_{n_0}(f), \hat{\psi}_{n_0}(f)\right)\right) + \underbrace{\sum_{n \neq n_0} a_n \langle \hat{\phi}_n(f), \hat{\psi}_{n_0}(f) \rangle}_{\text{ISI}} \qquad 1.6$$

Channel bandwidth and latency considerations are the two key design constraints that any practical communication system must address to operate effectively. Due to these requirements, the signals used in communication systems are finite in duration and approximately bandlimited. [1].

The selection of transmit basis signals is further refined by their interaction with the communication channel. Specifically, the received basis signal $\phi_n(t)$ results from the transformation of the transmit basis signal $\psi_n(t)$ by the channel.

$$\phi_n(t) = \sum_{i=1}^{P} h_i \psi_i(t)\,, \tag{1.7}$$

In this scenario, the transformation results in a sum where each term is weighted by a weighting factor h_i, involving a restricted number P of basis functions. Consequently, the ISI is limited to these P terms, which can simplify the equalization process. A channel is considered sparse when P is much smaller than the total number of the transmit basis signals ψ_{TX}, suggesting that the orthogonality of ψ_{TX} remains largely unaffected.

The chosen transmit basis in OTFS provides a clear depiction of high-mobility channels characterized by P paths. This representation shows a sparse two-dimensional signal composed merely of P terms. It highlights the efficiency of OTFS in managing the complexities associated with high-mobility scenarios.

$$h(\tau, \nu) = \sum_{i=1}^{P} h_i \tilde{\psi}_{l_i,k_i}(\tau, \nu) \tag{1.8}$$

Where l_i denotes the delay index and $k_i$denotes the Doppler shift index, the ith path ranges from 1 to P. Utilizing this basis effectively reduces terms contributing to ISI, thus, allowing for a comparatively straightforward iterative detection method [1].

1.4 Organization of the paper

The first chapter is about the evolution of the Mobile Communication System and the ongoing quest for a new communication technology suitable for 6G's highly dynamic environments. It also covers the topic of waveforms for high mobility channel conditions, including a discussion

on orthogonality condition. In the second chapter, various wireless channels are reviewed in detail. Moreover, the transition from TD to TF, and then to DD domain is thoroughly examined. The third chapter offers a concise introduction to OFDM highlighting its limitations in high Doppler conditions and proposes OTFS modulation as the path forward. Next, in the fourth chapter, OTFS is introduced in detail. This chapter covers the theoretical foundation of OTFS, its mathematical modeling, and the principles behind its robustness in high mobility scenarios. Finally, in the fifth chapter, we analyze the BER curve and the performance of OTFS vs. OFDM under different scenarios.

Chapter 2

Wireless channel

The mobile radio channel inherently limits the performance of wireless communication systems. The route for signal transmission from sender to receiver can range from a straightforward line of sight to paths significantly blocked by urban structures, natural landscapes, and vegetation. Unlike wired channels' stationary and predictable nature, radio channels are notably erratic and challenging to analyze with simplicity.

The propagation of electromagnetic waves is affected by several factors such as reflection, diffraction, and scattering. In urban environments, where cellular radio systems predominantly operate, a direct LoS does not exist between transmitter and receiver, further deteriorating the performance. Moreover, the presence of buildings adds to diffraction loss in these environments.

2.1 Path loss

As the signal is transmitted from the transmitter, its strength decreases, and this reduction in strength continues as the distance increases. This phenomenon is called path loss. Wireless channels experience significant variations in received power or the intensity of the electromagnetic field at the receiver's location. Path loss, large-scale fading and small-scale fading each contribute to this fluctuation in the received power.

In path loss, there is a decrease in power of electromagnetic waves proportional to the distance they travel. The attenuation factor can be modeled as d^{β} [15] where d is the distance traveled by the wave and β is the path loss exponent. Also, the path loss is calculated as $PL = 10\beta log_{10}(d)$ and is measured in decibels.

2.2 Large scale fading

The average attenuation of signal power, or path loss, experienced by a signal as it propagates over a long distance is defined as large-scale fading. The strength of a signal gradually reduces as the signal travels. Barriers such as thick vegetation or block can block or diminish the signal, resulting in shadowing and absorption losses. [3,4]. Experimental data observed over such distances has shown that power levels fluctuate around a mean value with these fluctuations having a relatively long period [5].

Large-scale fading has a spatially consistent effect on received power over areas with dimensions ranging from 10 to 100 times the wavelength λ_c, indicating a large relative scale. Over the years, different models have been developed to describe large-scale fading. Extensive data analysis has demonstrated that large-scale fading can be effectively modeled as a random variable following a lognormal distribution around the mean, which allows for the estimation of path loss as a function of distance [6].

2.3 Small scale fading

In contrast to pathloss and large-scale fading which are constant over larger distances and time periods, small-scale fading refers to the rapid fluctuation in a strength of a radio signal's amplitude over short periods or distances. Fading occurs when multiple copies of the same signal arrive at the receiver with slight timing or phase differences. These copies interfere with each other, causing large fluctuations in signal strength. Multipath propagation leads to small-scale fading, characterized by quick signal strength variations, random frequency shifts due to differential Doppler effects, and echoes from propagation delays. Factors such as the environment's multipath characteristics, the mobile's velocity, the speed of nearby objects, and the signal's bandwidth significantly influence these rapid signal changes in wireless communication channels. [7]

Types of small-scale fading

Transmitted signals experience various fading types based on how their parameters, such as bandwidth and symbol period, align with the channel's characteristics, like root mean square (RMS) delay and Doppler spread.

2.3.1 Flat fading channel

When a signal experiences almost the same channel response across all frequency components throughout the transmission bandwidth, it is described as flat fading. In flat fading, the channel's effect is uniform across the entire bandwidth of the signal, meaning that all frequency components of the signal experience the same amount of fading. This type of channel does not exhibit time or frequency dispersion. This allows the received signal to be characterized as the transmitted signal scaled by a specific coefficient, assuming there is no noise. However, due to the multiple propagation paths of the transmitted signal, the channel's gain experienced by a signal in each propagation path varies. As a result, the strength of the received signal fluctuates.

Flat fading occurs when,

$$B_s \ll B_c \tag{2.1}$$

and

$$T_s \gg \sigma_\tau \tag{2.2}$$

where T_s and B_s denotes the symbol period (reciprocal bandwidth) and bandwidth of the transmitted signal. B_c and σ_τ denote the coherence bandwidth and RMS delay spread of the channel respectively [7].

Flat fading channels prevalently occur in indoor settings that have a strong line-of-sight (LOS) component and minimal multipath effects. In such channels, the transmitter and the receiver tend to be stationary or moving slowly. As a result, delays from different paths are nearly identical.

2.3.2 Frequency selective/time dispersive channels

In frequency-dispersive channels, various attenuated and delayed copies of the transmitted waveform are received. This leads to signal distortion due to frequency selective fading caused by symbol time dispersion. This occurs when the channel's multipath delay spread exceeds the inverse of the transmitted signal's bandwidth, resulting in inter-symbol interference. Frequency domain analysis shows that some frequency components are amplified more than others, making the channel frequency selective. Such channels, common in large indoor or outdoor environments with significant multipath effects, exhibit large delays across different paths and typically involve slow-moving or stationary transmitters and receivers. For mobile communications, models like the 2-ray Rayleigh fading model [11] are used to study frequency-selective small-scale fading statistically.

Frequency selective fading occurs when,

$$B_s > B_c \tag{2.3}$$

And

$$T_s < \sigma_\tau \tag{2.4}$$

In this type of fading, the bandwidth (B_s) of the transmitted signal's spectrum $S(f)$ exceeds the channel's coherence bandwidth B_c [7].

2.3.3 Time selective/ frequency dispersive channels

Whenever there is any relative movement between the transmitter and receivers, frequency shifts occur in the transmitted electromagnetic wave within a frequency dispersive channel.

This phenomenon is called the Doppler effect. This effect can result in significant signal distortions based on the signal bandwidth and relative velocity.

In time-selective channels, the channel impulse response varies rapidly within the duration of a symbol. This indicates that the channel's coherence time is less than the duration of the transmitted signal's symbol. This leads to signal distortion, known as frequency dispersion or time-selective fading, which is caused by Doppler spreading.

Time selective fading occurs when,

$$T_s > T_c \tag{2.5}$$

And

$$B_s < B_D \tag{2.6}$$

Where T_s is the symbol period and T_c is the coherence time of the channel and B_D is the doppler spread [7].

2.4 Doubly selective channels

A channel exhibits double selectivity, i.e., it is time and frequency dispersive relative to the transmitted signal, when it is affected by multipath and Doppler effects. [3,4]. This phenomenon occurs due to signal transmission amidst significant multipath effects with notable delay and Doppler shifts. Such channels, often encountered in outdoor settings with strong multipath components, feature large excess delays and involve transmitters and receivers moving at high speeds.

Delay/frequency and Doppler/time diversity are new degrees of freedom introduced by doubly selective channels. These diversities emerge from the inherent limitations of real-life channels and signals in terms of time duration and bandwidth. Bello[8] leveraged these constraints to develop useful canonical channel models using sampling theorems and power series

expansions. Specifically, Orthogonal Time Frequency Space (OTFS) waveforms can fully exploit this time-frequency diversity.

2.5 Deterministic representation

In this section we aim to mathematically represent wireless channels according to their specific operating conditions. We aim to mathematically represent wireless channels in the complex baseband domain, specifically examining LTV channels with P resolvable paths. Each path characterized by the complex fading coefficient $h_{p,}$ time delay τ_p, and Doppler frequency ν_p for each path p, where $1 \leq p \leq P$.

2.5.1 Characterizing LTV Channels in the TD domain

Consider a static wireless channel with a base-station transmitter, a mobile receiver, and a stationary reflector like a building. In this scenario, since everything is static, the transmitted signal is unaffected by the Doppler shift. However, the direct and reflected paths cause the signal $s(t)$ to reach the mobile receiver at two different times due to varying propagation delays. As a result, r_1 represents the delay of a direct route taken by the signal traveling from base station to the mobile receiver, while the reflected signal travels a longer path, combining distance of $r_2 + r_3$. Assuming $h_1$and $h_2$be the complex gain (attenuation) of the direct and the reflected paths, the principle of superposition allows the received signal $r(t)$ to be expressed as,

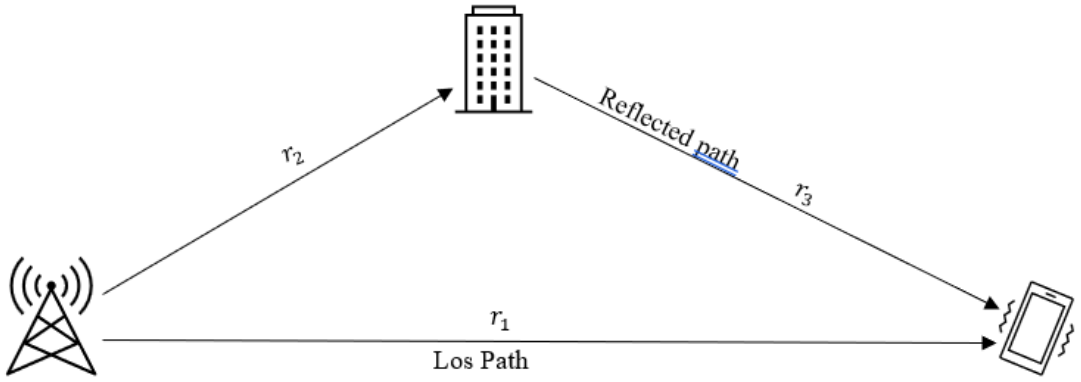
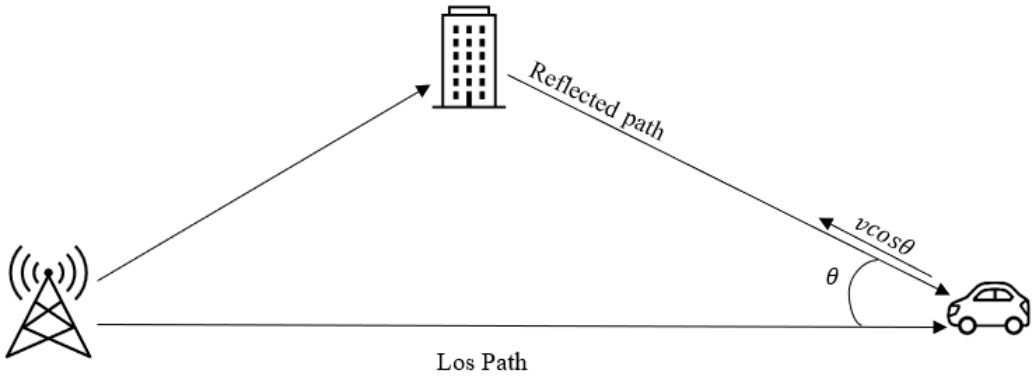


Figure 2.1: Static channel with direct LoS signal and a reflected path from building

$$r(t) = h_1 s(t - \tau_1) + h_2 s(t - \tau_2), \qquad 2.7$$

Where the delay for the LoS is given by $\tau_1 = r_1/c$, and for the reflected path, the delay is $\tau_2 = (r_2 + r_3)/c$, and $c = 299792458$ m/s is the speed of the light. The time difference between the longest and the shortest paths is $\tau_2 - \tau_1$ and is called delay spread. This measure indicates the variance in arrival times of the signal at the receiver caused by different propagation paths.



2. 2: Dynamic channel with a direct line-of-sight signal and a reflected path from a building

Now let's consider a scenario where mobile receiver is moving with a relative velocity v towards the base station. Also, let B be the bandwidth of $s(t)$. It is assumed to be significantly

smaller than the carrier frequency f_c, such that $f_c >> B$. The Doppler shift resulting from a relative motion is $v/c * f_c$. Under these conditions, the received signal is composed of the delayed and Doppler-shifted versions of the transmitted signal [1].

$$r(t) = h_1 e^{j2\pi\nu_1(t-\tau_1)} s(t-\tau_1) + h_2 e^{j2\pi\nu_2(t-\tau_2)} s(t-\tau_2) \quad 2.8$$

The attenuation for the LoS path is given by

$$h(t,\tau_1) = h_1 e^{j2\pi\nu_1(t-\tau_1)} \quad 2.9$$

where $h_1\ and\ \tau_1$ are the complex gain and the propagation delay for LoS path, and $\nu_1 = \frac{v}{c} f_c$ is the Doppler shift for the LoS path.

The attenuation for the reflected path is expressed as,

$$h(t,\tau_2) = h_2 e^{j2\pi\nu_2(t-\tau_2)} \quad 2.10$$

where h_2 represents the complex gain of the reflected path, τ_2 indicates the propagation delay for the reflected path, and $\nu_2 = \frac{vcos\theta}{c} f_c$ denotes the Doppler shift for the reflected path, θ being the angle of incidence.

Doppler spread is given by, $|\nu_2 - \nu_1|$ which indicates the range of frequency shift. It affects the signal over time.

$$h(t,\tau_i) = h_i e^{j2\pi\nu_i(t-\tau_i)} \quad 2.11$$

The overall time-varying attenuation is then the sum of these two components, reflecting the combined effects of path loss, propagation delays, and Doppler shifts on the signal as it gets transmitted to the moving receiver.

The relationship between input-output for a two-path model described in the above equation can be extended to a system with *P* propagation paths as follows,

$$r(t) = \sum_{i=1}^{P} h_i s(t-\tau_i) e^{j2\pi\nu_i(t-\tau_i)} \quad 2.12$$

Where h_i is the complex attenuation factor, τ_i is delay of the $i-th$ path, and ν_i is the Doppler frequency for i^{th} path. Equation (2.12) formulates the effect of P discrete point scatters on signal transmission [1]. This approach illustrates that the LTV channel time-domain response can be characterized by effectively capturing the dynamics of signal propagation through such a channel [8].

$$h_{TD}(t,\tau) = \sum_{i=1}^{P} h_i e^{j2\pi\nu_i t}\delta(\tau-\tau_i) = \sum_{i=1}^{P} h_i e^{j2\pi\nu_i(t-\tau_i)} \quad 2.13$$

The relationship between the transmitted signal s(t) and the received signal r(t) is given by,

$$r(t) = \int_{-\infty}^{\infty} s(t-\tau) h_{TD}(t,\tau) d\tau \quad 2.14$$

Based on (2.14), it is observed that the maximum distinct responses by the TD channel in the delay domain is P, indicating a limited number of separable paths. In addition, across the time domain, the channel responds to all components related to the Doppler frequency.

2.5.2 Characterizing LTV Channels in the TF domain

The TF channel response is achieved when FT is applied to the delay domain of TD function as indicated in equation (2.13) [9].

$$h_{TF}(t,f) = \int_{-\infty}^{\infty} h_{TD}(t,\tau) e^{-jt\pi f\tau} d\tau = \sum_{p=1}^{P} h_p e^{j2\pi\nu_p(t-\tau_i)} e^{-j2\pi\tau_p f} \quad 2.15$$

The TF domain channel representation, largely due to OFDM, has become the most known channel in wireless communications.

2.5.3 Characterizing LTV Channels in the DD domain

The DD channel response is achieved when FT is applied to the time domain of TD function as indicated in equation (2.13) [9].

$$h_{DD}(\tau,\nu) = \int_{-\infty}^{\infty} h_{TD}(t,\tau)e^{-j2\pi t\nu}dt = \sum_{p=1}^{P} h_p\delta(\tau-\tau_p)\delta(\nu-\nu_p) \qquad 2.16$$

Also, the response in the DD domain is derived by applying the SFFT to the TF channel [1]. This provides a concise analysis farmwork for systems such as OTFS modulation.

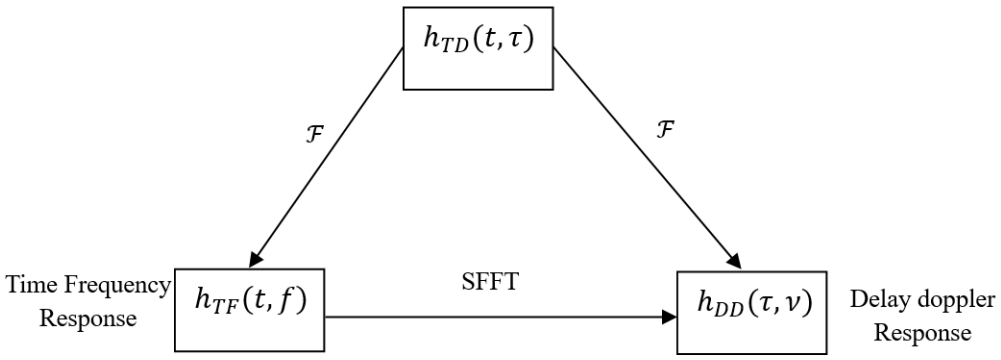


Figure 2.3: Transformation between various channel domains

The SFFT converts the dense TF domain channel responses into sparse DD domain channel responses by applying a FT to the time domain and an IFT to the frequency domain [9]. It also ensures path responses are separable, meaning there is no interference between different paths. Additionally, the channel response in the DD domain is compact, providing a concise representation of the channel that is confined to the area determined by the maximum time delay and Doppler frequency.

The received signal in a LTV channel from (2.14) can be represented as,

$$r(t) = \int_{-\infty}^{\infty} s(t-\tau) h_{TD}(t,\tau) d\tau \quad (Linear\ time\ variant\ channel)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_{DD}(\tau,\nu) s(t-\tau) e^{j2\pi\nu t} d\tau d\nu \quad (Delay\ Doppler\ channel) \qquad 2.17$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_{TF}(t,f) S(f) e^{j2\pi f t} df dt \quad (Time\ Frequency\ channel) \qquad 2.18$$

The relationship between delay doppler channel $h_{DD}(\tau,\nu)$ and Time Frequency channel $h_{TF}(t,f)$ is given by,

$$h_{DD}(\tau,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_{TF}(t,f) e^{-j2\pi(\nu t - f\tau)} dt df \qquad 2.19$$

$$h_{TF}(t,f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_{DD}(\tau,\nu) e^{j2\pi(\nu t - f\tau)} d\nu d\tau \qquad 2.20$$

The DD channel and the TF channel constitute a pair of symplectic Fourier transform pairs, meaning that the relationship between two domains is characterized by the application of symplectic Fourier transforms to convert signals from one domain to the other and vice versa [10].

$$h_{TF}(t,f) \underset{\text{ISFFT}}{\overset{\text{SFFT}}{\rightleftarrows}} h_{DD}(\tau,\nu) \qquad 2.21$$

Symplectic Fourier transforms are essentially dual Fourier transforms that utilize symplectic inner products in the exponent. Within the framework of OTFS modulation and demodulation, this concept [10] is implemented as,

$$e^{-j2\pi(\nu t - f\tau)} = e^{-j2\pi\nu t} . e^{+j2\pi f\tau} \qquad 2.22$$

$$e^{j2\pi(\nu t - f\tau)} = e^{+j2\pi\nu t} . e^{-j2\pi f\tau} \qquad 2.23$$

The discrete SFFT and its ISFFT based on the principles of Fourier transforms are defined for the periodized version $X_p[m,n]$ of $X[m,n]$ with period (M,N) as,

$$x_p[k,l] = SFFT\big(X_p[m,n]\big)$$

$$= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_p[m,n] e^{-j2\pi\left(\frac{mk}{M} - \frac{nl}{N}\right)} \qquad 2.24$$

$$X_p[m,n] = ISFFT(x_p[k,l])$$

2.6 Summary

In this chapter, we explored the properties of wireless channels and the various fading conditions affecting them. We then delved into the deterministic representation of the channel, detailing the transition from TD to TF and then to DD domain. Additionally, the concept of ISFFT was briefly introduced towards the end of the discussion.

Chapter 3

OFDM waveform limitation and path forward

3.1 Introduction

OFDM is a fundamental modulation technology widely recognized for its efficiency in modern digital communications. It addresses the growing demand for high data rate transmissions required by various applications like broadband internet, digital television, and cellular networks. OFDM operates by dividing the entire channel into many narrow band subchannels. These are then transmitted in parallel, which not only maintains high data rate transmissions but also combats the challenges of multipath propagation and frequency-selective fading. The efficiency of OFDM is further enhanced by increasing the symbol duration, allowing each subcarrier in the OFDM signal to have a longer period. This strategic adjustment significantly reduces the relative impact of multipath delay spread, thereby mitigating inter-symbol interference (ISI). Consequently, OFDM facilitates a robust and high-speed data transmission crucial for meeting today's digital demands and paving the way for future advancements.

OFDM divides high-rate data streams into N parallel streams, which are subsequently modulated onto N distinct carriers (subcarriers). Each subcarrier's symbol duration is extended by a factor of N [6], allowing the receiver to distinguish signals from different subcarriers. To maintain orthogonality, conventional FDMA would separate subcarriers by a significant margin, thus wasting valuable spectrum which is shown in the 3.1. OFDM achieves much closer subcarrier spacing without sacrificing orthogonality, assigning subcarriers at frequencies

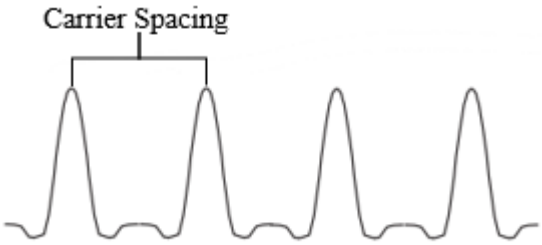


Figure 3.1: Illustration of an FDMA system with large gaps between consecutive subcarriers to avoid interference.

$f_n = nB/N$ where $n = 1,2,3 \ldots, N$. and B is the total bandwidth. Modulation is then performed using a rectangular pulse.

The essence of OFDM lies in maintaining the orthogonality of the carriers. Two signals are considered orthogonal if their product integrates to zero over a specific period T. This principle is expressed as,

$$\int_0^T cos\,(2\pi n f_0 t)\, cos\,(2\pi m f_0 t)\, dt = 0\ (n \neq m) \qquad 3.1$$

Here, n and m are distinct integers, f_0 signifies the fundamental/ base frequency and is set to 1/T to optimize performance. This definition guarantees that despite the overlap in the spectra of modulated carriers-induced by the time-domain rectangular pulses, there is no interference between data streams of any two subcarriers.

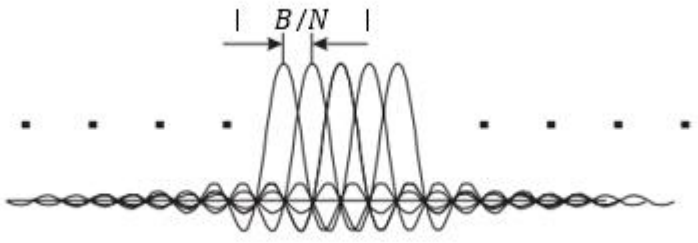


Figure 3. 2: Illustration of an OFDM system where the individual subcarriers are spaced by 1/T Hz

3.1.1 Implementation of OFDM transceivers

OFDM can be realized through both analog and digital methodologies. Beginning with analog techniques, consider a standard communication system designed for amplitude shift keying (ASK) as shown in the figure. In this system, the input signal X_0 can take values of -1 and 1. This signal is modulated using a sinusoid $\left(m_0(t) = e^{\frac{j2\pi kt}{T}} for\ k = 0\right)$ over a period of T. The modulated signal, $s(t)$ is then transmitted through a channel and then received by a receiver. At $t = T$, the received signal is subsequently processed by a matched filter, comprising oscillators, integrators, and filters-components which are the fundamental of analog electronics.

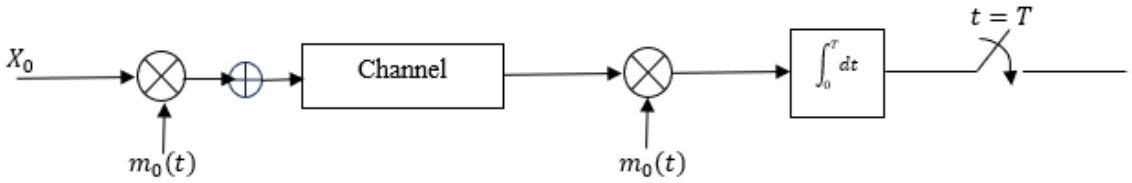


Figure 3.3: Realization of a single carrier ASK system with analog components

Let's consider an additional parallel orthogonal channel X_1 with the carrier frequency $\left(m_1(t) = e^{\frac{j2\pi kt}{T}} for\ k = 1\right)$ as shown in the Fig. 3.4. Before transmission, $X_0 m_0(t)$ and $X_1 m_1(t)$ are summed. The combined signal is represented as,

$$s(t) = X_0 m_0(t) + X_1 m_1(t) \qquad 3.2$$

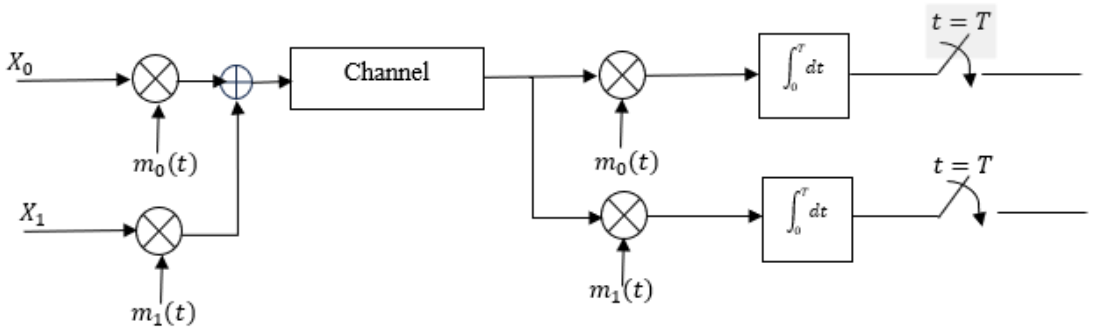


Figure 3.4: Realization of a single carrier ASK system with analog components

Orthogonality Principle in Practice

Upon receiving the signal $s(t)$, it is fed into an integrator over a symbol period T. The following equation shows how the orthogonality principle can distinguish between $x_0(t)$ and $x_1(t)$.

$$\begin{aligned} X_0'(t) &= \int_0^T s(t)m_0(t)dt \\ &= \int_0^T \{X_0 m_0(t) + X_1 m_1(t)\} m_0(t)dt \\ &= X_0 \int_0^T {m_0}^2(t)dt + X_1 \int_0^T \underbrace{m_0(t)m_1(t)}_{\text{zero}} dt \\ &= X_0 \int_0^T {m_0}^2(t)\,dt \end{aligned} \qquad 3.3$$

Generalization to DFT

Expanding the inputs to encompass $X_0, X_1, \ldots, X_{N-1}$, we utilize sinusoids expressed as $e^{\frac{j2\pi kt}{T}}$, where the integer selection of k guarantees that the frequency elements maintain orthogonality. This critical orthogonality among frequency components ensures their distinct

separability upon reception, despite concurrent transmission across the same channel. The extended figure is shown in figure 3.5.

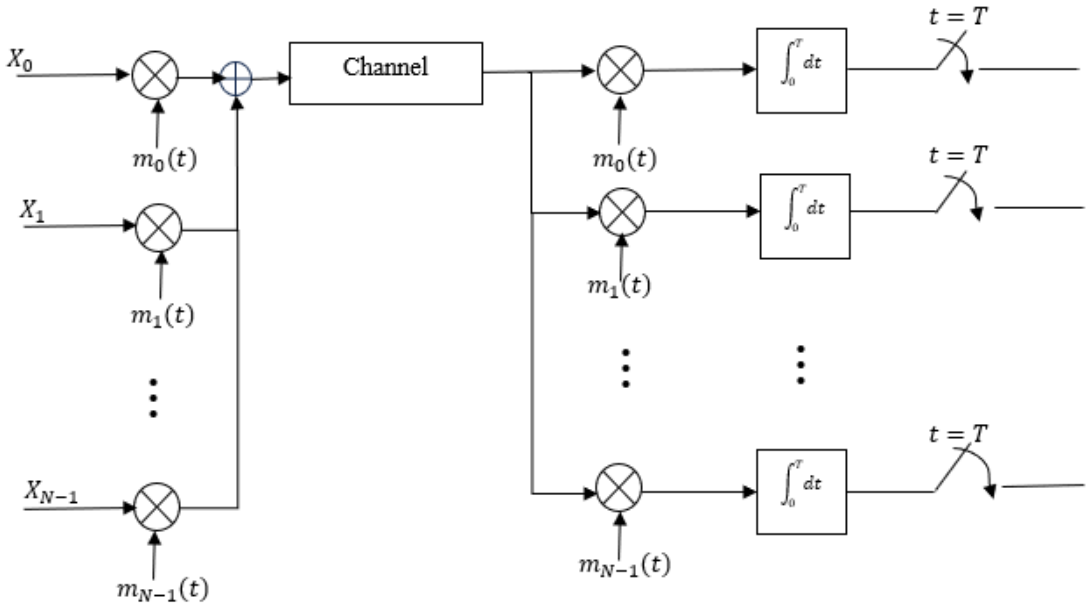


Figure 3.5: Illustration of an OFDM system with N sub-carriers using analog components

Thus, the encoding process for N sub-carriers can be generalized as,

$$x(t) = \sum_{k=0}^{N-1} X_k e^{\frac{j2\pi kt}{T}} \qquad 3.4$$

It can be seen that this is exactly DFT. Instead of doing analog electronics for implementation for OFDM, we can replace all of this by IDFT. Then $X_0, X_1, \dots, X_{N-1}$ can be viewed as being FD components of signal $s(t)$ which is being sent over the channel. The DFT implementation of OFDM is shown in the figure.

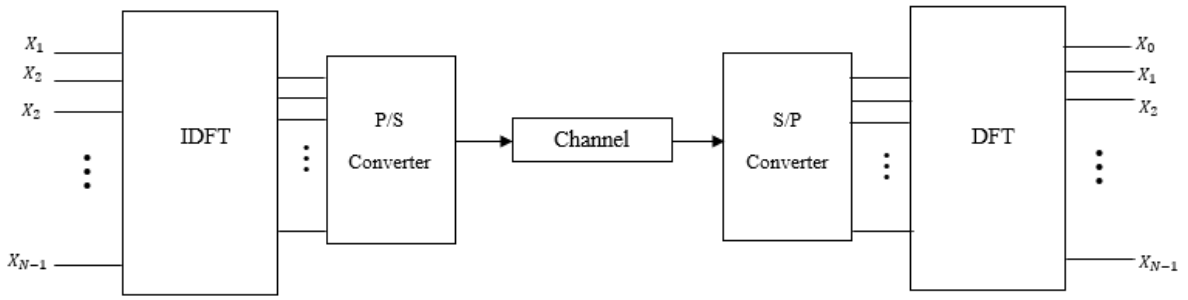


Figure 3.6: An OFDM system implemented digitally using the DFT

The DFT-based implementation of OFDM involves organizing the transmitted data into blocks of N symbols, each processed through an Inverse Fast Fourier Transform (IFFT) before transmission. This sequence of N values is transmitted serially, necessitating a parallel-to-serial (P/S) conversion post-IFFT. The processor is reversed at the receiver. The signal is sampled into N-sample blocks, transformed from serial-to-parallel (S/P), and converted back to frequency domain using a FFT operation. This digital approach simplifies integration into circuits compared to the analog method. Analog OFDM, requiring multiple Local Oscillators (LOs) with minimal phase noise and drift to maintain subcarrier orthogonality, proves impractical. The efficacy and cost-efficiency of OFDM stem from this streamlined digital implementation.

3.2 Impact of Doppler Spreading on OFDM performance

In high-mobility scenarios like vehicular networks or high-speed trains, OFDM's performance is notably degraded due to the loss of orthogonality between sub-carriers because of the Doppler effect, which leads to ICI which corrupts signal quality. This issue is compounded in typical vehicular environments, which are doubly selective or time-varying multipath channels influenced by multipath propagation and frequency shifts. Urban areas, with their multiple obstacles, create complex propagation paths, each contributing unique delays and shifts. Consequently, understanding these environments requires a two-dimensional representation of the wireless channel to accurately depict the interplay of delays and Doppler shifts. Addressing

the challenges posed by the Doppler effect and multipath propagation is crucial for improving OFDM performance in high mobility scenarios.

The channel variations over time can be approximated using a Taylor series expansion [11] around a point t_0. The first-order Taylor series is given by,

$$\alpha_k(t) \approx \alpha_k(t_0) + \alpha_k'(t_0)(t - t_0) \quad 3.5$$

This assumes that the channel state doesn't change too quickly over the symbol duration T, so we can use just the first derivative (the rate of change) to approximate it.

The impulse response the k-th subchannel at a certain time t is a function of channel variation α_k(t) and delta function $\delta(t)$. This represents how an instantaneous signal (an impulse) would be affected by the channel at time t.

$$c_k(\tau; t) = \alpha_k(t_0)\delta(\tau) + (t - t_0)\alpha_k'(t_0)\delta(\tau) \quad 3.6$$

The transmitted baseband signal s(t) is expressed as,

$$s(t)\frac{1}{\sqrt{T}}\sum_{k=0}^{N-1} s_k e^{j2\pi f_k t}, \qquad 0 \le t \le T \quad 3.7$$

The received baseband signal r(t) with additive noise may be expressed as,

$$r(t) = \frac{1}{\sqrt{T}}\sum_{k=0}^{N-1} \alpha_k(t) s_k e^{j2\pi f_k t} + n(t) \quad 3.8$$

Here $n(t)$ represents the additive noise characterized as a complex Gaussian noise exhibiting uniform spectral distribution across signal's bandwidth. This implies that the noise maintains a constant power spectral density, denoted as N_0watts/Hz. Through the application of the Taylor series expansion to $\alpha_k(t)$, we obtain

$$r(t) = \frac{1}{\sqrt{T}} \sum_{k=0}^{N-1} \alpha_k(t_0) s_k e^{j2\pi f_k t} + \frac{1}{\sqrt{T}} \sum_{k=0}^{N-1} (t - t_0) \alpha_k'(t_0) s_k e^{j2\pi f_k t} + n(t) \quad 3.9$$

The signal received within a given duration is processed via a series of correlators arranged in parallel, with each correlator specifically aligned to one of the N subcarriers. The i^{th} correlator output is,

$$\hat{s}_i = \frac{1}{\sqrt{T}} \int_0^T r(t) e^{-j2\pi f_i t} dt \quad 3.10$$

From 3.9 and 3.10, we get [19],

$$\hat{s}_i = \alpha_i(t_0) s_i + \frac{1}{2\pi j} \sum_{k=0\, k\neq i}^{N-1} \frac{\alpha_k'(t_0) s_k}{k - i} + n_i \quad 3.11$$

The mean square value of the desired signal component is,

$$\begin{aligned} S &= E[|\alpha_i(t_0) s_i|^2] \\ &= E[|\alpha_i(t_0)|^2] E[|s_i|^2] = 1 \end{aligned} \quad 3.12$$

Where the average channel gain is normalized to unity and the power withing the ICI term is expressed as [19],

$$I = E\left[\left|\frac{1}{2\pi j} \sum_{k=0\, k\neq i}^{N-1} \frac{\alpha_k'(t_0) s_k}{k - i}\right|^2\right]$$

$$\begin{aligned} &= \left(\frac{T}{2\pi}\right)^2 \sum_{k=0\, k\neq i}^{N-1} \sum_{l=0\, l\neq i}^{N-1} \frac{1}{(k-i)(l-i)} E[\alpha_k'(t_0) s_k (\alpha_l'(t_0) s_l)^*] \\ &+ \left(\frac{T}{2\pi}\right)^2 \sum_{k=0\, k\neq i}^{N-1} \frac{1}{(k-i)^2} E[|\alpha_k'(t_0) s_k|^2] \end{aligned} \quad 3.13$$

The pair $\left(\alpha_k'(t_0), \alpha_l'(t_0)\right)$ is statistically independent of (s_k, s_l). The first term of the right-hand side of the equation is zero. The power of the ICI component is [19],

$$I = \frac{(T f_m)^2}{2} \sum_{k=0\, k\neq i}^{N-1} \frac{2\varepsilon_s}{(k-i)^2} \quad 3.14$$

The signal to interference ratio (S/I) ration can be expressed as,

$$\frac{S}{I} = \frac{1}{\frac{(Tf_m)^2}{2}\sum_{k=0\,k\neq i}^{N-1}\frac{1}{(k-i)^2}} \tag{3.15}$$

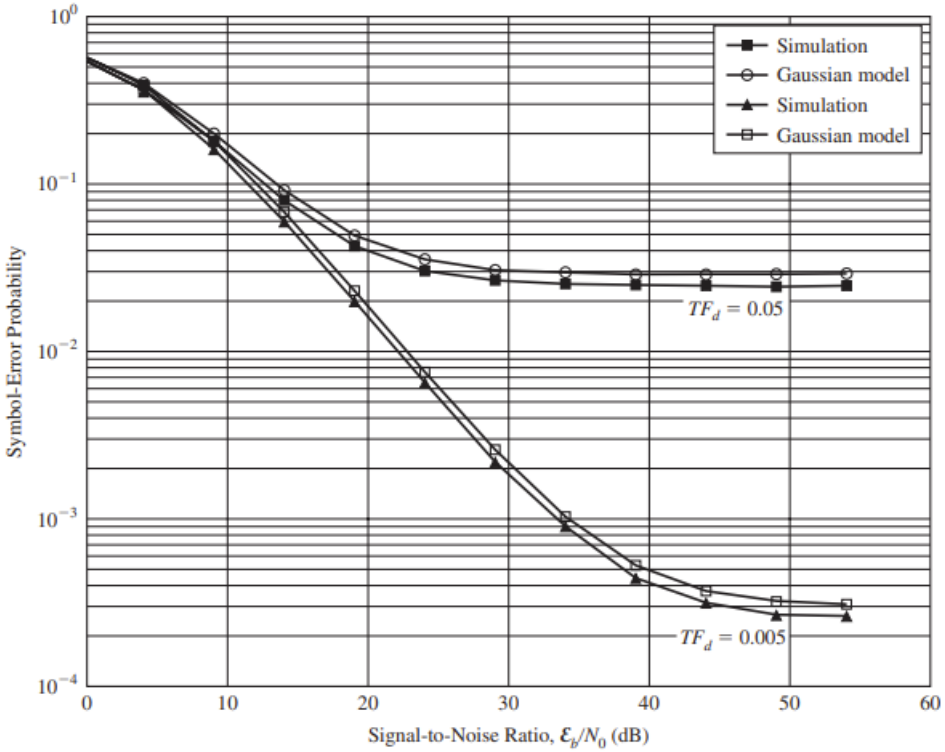


Figure 3.6: This plot shows the symbol error probability and demonstrates that ICI significantly impairs the performance of an OFDM system with 256 subcarriers and 16-QAM [19]

The symbol error probability for an OFDM system with 256 subcarriers and 16-QAM is illustrated in Figure 3.6. The error probability is determined both analytically, using the Gaussian model for ICI, and using Monte Carlo simulation. The result show that ICI significantly deteriorates the performance of the OFDM system.

3.3 OTFS: A novel approach to modulation in high-mobility scenarios

To overcome the challenges presented by high mobility environments, a novel modulation framework called Orthogonal Time Frequency Space (OTFS) has been proposed [2]. OTFS is designed to efficiently manage channel variations caused by high mobility. Unlike OFDM,

which transmits data on a time-frequency grid, OTFS operates in the DD domain. This means OTFS considers the channel's response over a wide range of potential delays (due to multipath) and Doppler shifts (due to mobility). This provides a two-dimensional representation of the channel's behavior over delay and doppler.

OTFS maintains orthogonality between transmitted symbols despite rapid changes in the channel, thereby significantly reducing ICI and improving the reliability of data transmission. As a result, OTFS is a promising candidate for the future of wireless communication, especially those that need to support high mobility, as it has been shown to outperform OFDM in conditions such as high mobility.

3.4 Summary

Here, we provided a brief introduction to OFDM and discussed the critical importance of maintaining orthogonality among carriers. Then, we examined the challenges OFDM faces under high Doppler conditions and presented OTFS as an innovative solution to address these issues in high-mobility environments.

Chapter 4

Orthogonal time frequency space (OTFS)

4.1 Introduction

Let's examine the basic principles of signal processing, which primarily hinges on two forms of signal representations: time and frequency, before we discuss OTFS. Signals are represented as series of delta functions in time and complex exponentials in frequency, with the ability to switch between these using Fourier transform. The Heisenberg uncertainty principle demonstrates that it is impossible to localize determine both the time and frequency of a signal simultaneously. But there exists a category of signals that appear to be localized in both dimensions simultaneously, a characteristic crucial for technologies like delay-Doppler Radar and wireless communications. These signals are defined within a two-dimensional delay-Doppler plane, characterized by variables of delay and Doppler.

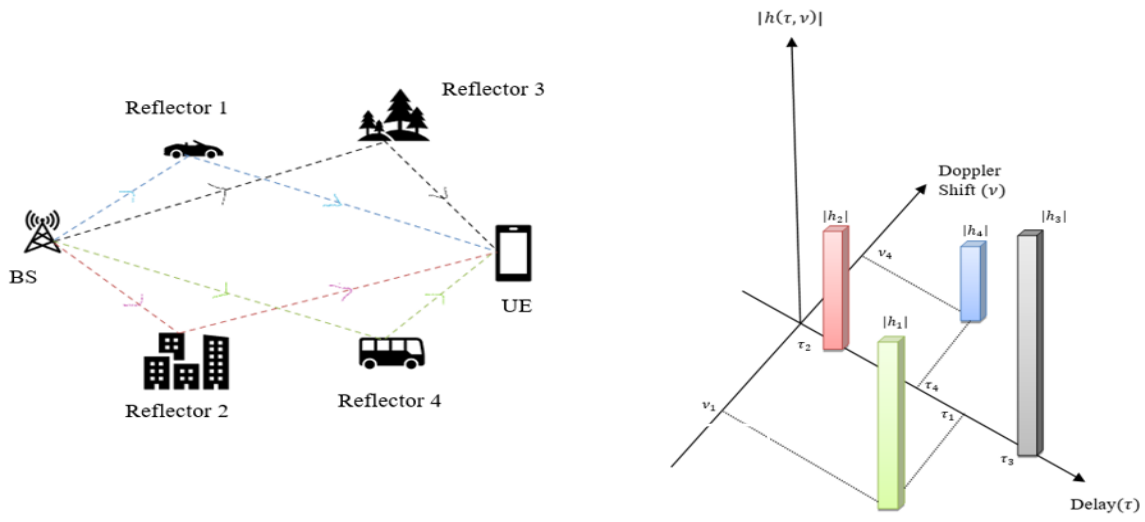


Figure 4.1: Illustration of a channel with 4 reflectors in the delay Doppler domain

The figure shows an urban area with buildings and vehicles, wireless signals between a transmitter and receiver mainly through a few key paths with different colored paths for each reflector type in a city. The signal reflection off various surfaces like buses, cars, and buildings creates these paths. Examining the channel's response in the DD domain reveals a sparse structure, highlighting the specific delays and Doppler shifts from groups of reflectors. This sparsity and geometry of the wireless environment are efficiently captured in the delay-Doppler representation, indicating that the channel changes more slowly than in the time-frequency domain. [12]

In this context, wireless channels are characterized as having "double spread", incorporating both DD shift and complex gain for each transmission path. The influence of the channel on a transmitted signal, x(t), is described by the DD spreading function $h(\tau.\upsilon)$. Here, $\tau \in R$ and $\nu \in R$ denote the delay and Doppler variables, respectively as illustrated in the figure.

$$h(\tau,\nu) = \sum_{i=1}^{4} h_i \delta(\tau - \tau_i)\delta(\nu - \nu_i) \tag{4.1}$$

The impulse function known as the Dirac-delta is represented by δ . The signal in the time domain, received by the user equipment (UE) without any noise, can be expressed as follows,

$$y(t) = \iint h(\tau,\nu)x(t-\tau)e^{j2\pi\nu(t-\tau)}d\tau d\nu \tag{4.2}$$

In fig 4.1,

$$y(t) = \sum_{i=1}^{4} h_i x(t-\tau_i)e^{j2\pi\nu_i(t-\tau_i)}, \tag{4.3}$$

This represents the total signal copies that are received via the four distinct paths.

Delay-Doppler variables $(\tau.\upsilon)$ are widely used in radar technology to distinguish moving targets by analyzing their unique delay and Doppler characteristics, and in the field of

communication, DD variables can be used to characterize channels through a combination of time and frequency components. However, these variables can also represent information signals in a way that matches the description of the DD channel. This method of signal representation is more complex mathematically, necessitating the use of quasi-periodic functions which is discussed in the coming section [13]. For this approach, we select specific delay (τ_r) and Doppler period (ν_r) that meet the criterion $\tau_r . \nu_r = 1$, essentially creating a unit area box. A DD signal ($\phi(\tau, \nu)$), must adhere to a quasi-periodic condition which ensures that the signal can be consistently identified over this unit area, based on its unique delay and Doppler shifts.

The transition between time and frequency domains is facilitated by the Fourier transform, which offers a well-established mathematical tool for analyzing signals in these domains. Similarly, Zak transform facilitates the transition between the DD domain and the time or frequency domains. Zak transforms denoted as Z_t for time domain and Z_f for frequency domain serve as crucial tools in this context. Zak transforms realization are shown in the 4.4 and 4.5.

$$Z_t(\phi) = \int_0^{\nu_r} e^{j2\pi t\nu} \phi(t, \nu) d\nu \tag{4.4}$$

$$Z_f(\phi) = \int_0^{\tau_r} e^{-j2\pi t\nu} \phi(\tau, f) d\tau \tag{4.5}$$

In simple terms, converting to the time domain via the Zak transform involves using the inverse Fourier transform over a Doppler period. Conversely, to get to the frequency domain requires the use of Fourier transform over a delay period.

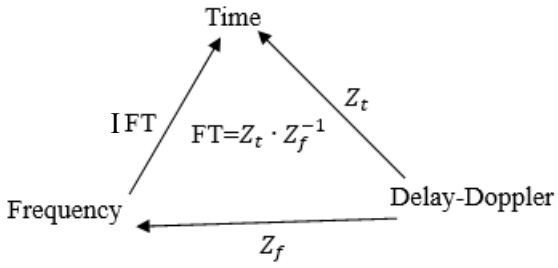


Figure 4.2: Transformation between the various representations of the underlying channel response

Data is transmitted using a quasi-periodic pulse in the DD domain in OTFS. The inverse Zak transform is used to convert a signal from the DD domain to the time domain signal. In OTFS, information is carried by a DD domain quasi-periodic pulse. The concept of delay-Doppler (DD) domain pulse involves two critical periods: the delay period ($\tau_p \in R_{\geq 0}$) and the Doppler periods, ($\nu_p \in R_{\geq 0}$) which must be reciprocal ($\nu_p = \frac{1}{\tau_p}$). Here, the Zak transform establishes a link between time domain signals and certain quasi-periodic signals in the DD domain. The definition of the Zak transforms for a time domain signal $x(t)$ is as follows,

$$\begin{aligned} x_{dd}(\tau,\nu) &= Z_t\big(s(t)\big) \\ &\triangleq \sqrt{\tau_p}\sum_{k=-\infty}^{\infty} x\big(\tau + k\tau_p\big)e^{-j2\pi\nu k\tau_p} \end{aligned} \quad 4.6$$

From 1. For any $n, m \in Z, x_{dd}(\tau,\nu)$ satisfies

$$\begin{aligned} x_{dd}\big(\tau + n\tau_p, \nu + m\nu_p\big) &= \sqrt{\tau_p}\sum_{k=-\infty}^{\infty} x\big(\tau + (k+n)\tau_p\big)e^{-j2\pi(\nu+m\nu_p)k\tau_p} \\ &= \sqrt{\tau_p}\sum_{k'=-\infty}^{\infty} x\big(\tau + k'\tau_p\big)e^{-j2\pi\nu k'\tau_p}e^{j2\pi n\nu\tau_p} \\ &= e^{j2\pi n\nu\tau_p}x_{dd}(\tau,\nu), n, m \in Z \end{aligned} \quad 4.7$$

The quasi-periodic condition 4.7 states that a quasi-periodic pulse is localized at the origin $(\tau,\nu) = (\tau_0,\nu_0)$ within a rectangular area.

$$D_0 \triangleq \{0 \leq \tau < \tau_p, 0 \leq \nu < \nu_p\} \quad 4.8$$

D_0 being the fundamental period along with its quasi periodicity enables the pulse to appear occur at integer translations $(\tau, \nu) = (\tau_o + n\tau_p, \nu_0 + m\nu_p)$, where $m, n \in Z$. The pulse appears in such way that when shifted by integer multiple of τ_p, there is a phase change in the pulse whereas when shifted by integer multiple of ν_p, no phase change in the pulse occurs. These pulses are an infinite series, repeating themselves at consistent intervals along the axes of the time delay and Doppler frequency. Despite having a certain width in both dimensions, indicating that each pulse is spread out over time and frequency. Despite the overall regularity, the pulses are not perfectly uniform in their repetition. To analyze these repeating pulses and their structure more deeply, techniques such as the inverse time Zak transform are employed, providing insights into the signal's composition in the time domain.

$$x(t) = Z_t^{-1}(x_{dd}(\tau, \nu)) \triangleq \sqrt{\tau_p} \int_0^{\nu_p} x_{dd}(t, \nu) d\nu \quad 4.9$$

Here each pulse within a sequence is separated from other pulses by delay period τ_p and extends over 1/B time duration. These pulses are generated at times given by $t = n\tau_p + \tau_0, n \in Z,$. Here, τ_0 denotes the delay coordinate associated with the DD domain pulse. Furthermore, the entire sequence of pulses is modulated by a sinusoidal signal with a frequency ν_0, which corresponds to the Doppler coordinate in the fundamental DD domain.

Subsequently, we utilize the inverse Zak transform $Z_f^{-1}$to convert the pulse, shifting it from the DD domain to the FD.

$$X(f) = Z_f^{-1}(x_{dd}(\tau, \nu)) \triangleq \sqrt{\nu_p} \int_0^{\tau_p} x_{dd}(\tau, f) e^{-j2\pi f\tau} d\tau \quad 4.10$$

In this context, $X(f)$ denotes a series of pulses with a bandwidth of B. Each pulse in the sequence has a bandwidth of $1/T$ and is spaced apart by the Doppler period ν_p. The pulses appear at frequencies given by $f = m\nu_p + \nu_0, m \in Z$, where m is an integer and ν_0 represents the Doppler coordinate of the primary DD domain pulse. The entire sequence is further

modulated by a sinusoidal function in the frequency domain $e^{-j2\pi f\tau_0}$ with τ_0 being the delay coordinate of the DD pulse.

In summary, the transformation to the time domain via the Zak transform is achieved using the inverse Fourier transform along the Doppler period. Conversely, it is transformed to the frequency domain by applying the Fourier transform along the delay period. It is crucial to maintain the quasi-periodicity condition for the Zak transform to ensure a one-to-one mapping between functions on a one-dimensional line and those on a two-dimensional plane. Without this condition, a signal on the line would have multiple possible representations in the DD domain.

Next, we will briefly review Zak transform and then we'll jump to multicarrier interpretation of OTFS.

4.2 Zak transform

Zak transform (ZT) operates by converting a single-variable function into a two-variable function.

ZT is expressed as,

$$Z_T[x(t)](\tau,\nu) = \sqrt{T}\sum_{n=-\infty}^{\infty} x(\tau+nT)e^{-j2\pi nT\nu}, -\infty < \tau,\nu < \infty \qquad 4.11$$

Here, x(t) is a complex, continuous-time function and T is a positive constant.

For any given delay or time shift within the range $-\infty < \tau,\nu < \infty$, equation 4.11 represents the discrete-time Fourier transform (DTFT) of the sampled DT discrete-time signal $x_\tau[n] = x(\tau+nT)$. This signal is obtained by sampling x(t) at intervals of T seconds, beginning from τ. It shows that Zak transform exhibits periodicity in the variable ν, with the period $\Delta f = 1/T$, such that T_s equal to T.

For $\nu = 0$ (no Doppler),

$$Z_T[x(t)](\tau, \nu) = \sqrt{T} \sum_{n=-\infty}^{\infty} x(\tau + nT),\ -\infty < \tau, \nu < \infty \tag{4.12}$$

The Zak transform exhibits periodicity in τ with period T mirroring the periodic replicas of the signal $x(t)$, and projects the signal $x(t)$ onto Zak basis functions, defined by delay τ and Doppler shift ν, like the Fourier transformation.

$$Z_T[x(t)](\tau, \nu) = \langle x(t), \phi_{\tau,\nu}(t) \rangle = \int_{-\infty}^{\infty} x(t) \phi_{\tau,\nu}(t)^*\, dt \tag{4.13}$$

Where $(\cdot)^*$denotes the complex conjugation,

$$\phi_{\tau,\nu}(t) = \sqrt{T} \sum_{n=-\infty}^{\infty} \delta(t - \tau - nT) e^{j2\pi(t-\tau)\nu} = \sqrt{T} \sum_{n=-\infty}^{\infty} \delta(t - \tau - nT) e^{j2\pi nT\nu} \tag{4.14}$$

These Zak basis functions are called pulsones as coined by its inventor Hadani. These pulsones consists of periodic pulse trains that are time-shifted by τ and modulated with a complex sine wave oscillating at frequency ν, combining pulse and tone characteristics.

4.2.1 Properties of the Zak transform

4.2.1.1 Quasi-periodicity

The ZT is quasi-periodic along the delay axis with a period T, and it maintains periodicity along the Doppler axis with a period of $\Delta f = \frac{1}{T}$.

$$Z_T[x(t)](\tau, \nu + \Delta f) = Z_T[x(t)](\tau, \nu), along\ the\ Doppler\ axis \tag{4.15}$$
$$Z_T[x(t + \tau_0)](\tau, \nu) = Z_T[x(t)](\tau + \tau_0, \nu), along\ the\ Delay\ axis$$

For $\tau_0 = mT$, m being an integer multiple of T

$$Z_T[x(t + mT)](\tau, \nu) = e^{j2\pi mT\nu} Z_T[x(t)](\tau, \nu) \tag{4.16}$$

4.2.1.2 Linearity

$$Z_T[ax(t) + by(t)](\tau, \nu) = aZ_T[x(t)](\tau, \nu) + bZ_T[y(t)](\tau, \nu) \tag{4.17}$$

Modulation by a complex sine wave with frequency ν_0

$$Z_T\left[e^{j2\pi\nu_0 t}x(t)\right](\tau,\nu) = e^{j2\pi\nu_0\tau}Z_T[x(t)](\tau,\nu-\nu_0) \quad 4.18$$

This shows that modulation shifts the Zak transform's Doppler axis by ν_0.

If $\nu_0 = m\Delta f$=m/T, where m is an integer multiple of Δf, then

$$Z_T\left[e^{j2\pi m\Delta f t}x(t)\right](\tau,\nu) = Z_T[x(t)](\tau,\nu) \quad 4.19$$

This implies that ν_0 being an integer multiple of Δf, the modulation does not change the Zak transform, indicating a periodic nature in the Doppler domain.

For a combined shift by τ_0 and modulation using a complex sine wave at frequency ν_0

$$Z_T\left[e^{j2\pi\nu_0 t}x(t-\tau_0)\right](\tau,\nu) = e^{j2\pi\nu_0(\tau-\tau_0)}Z_T[x(t)](\tau-\tau_0,\nu-\nu_0) \quad 4.20$$

This demonstrates how both translation (shift it time) and modulation (shift in frequency) simultaneously affect the Zak transform, altering both delay and Doppler domains.

4.2.1.3 Conjugation

$$Z_T[x^*(t)](\tau,\nu) = Z_T^*[x(t)](-\tau,-\nu) \quad 4.21$$

This indicates that taking the complex conjugate of the signal before applying the ZT, and then applying the ZT to the original signal and taking the conjugate of the result, are equivalent operations, except that in the latter case, the signs of both the delay (τ) and the Doppler shift (ν) are reversed.

4.2.1.4 Symmetry

$$Z_T[x(t)](\tau,\nu) = Z_T^*[x(t)](-\tau,-\nu), if\ x(t)\ is\ even, \quad 4.22$$

$$Z_T[x(t)](\tau,\nu) = -Z_T^*[x(t)](-\tau,-\nu), if\ x(t)\ is\ odd, \quad 4.23$$

In this case, for an odd function x(t), the Zak transform equals the negative of the complex conjugate of its transform, also evaluated at the negated delay and Doppler shift.

4.2.1.5 Product in time

$$y(t) = h(t)x(t) \quad 4.24$$

Then

$$Z_T[y(t)](\tau,\nu) = \sqrt{T}\int_0^{\Delta f} Z_T[h(t)](\tau,u)Z_T[x(t)](\tau,u-\nu)du \quad 4.25$$

In essence, this demonstrates that the ZT of the product of two signals results from convolving their individual ZT in the Doppler domain. This convolution is integrated over the interval from 0 to Δf, which represents the Doppler frequency resolution.

4.2.1.6 Convolution in time

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\theta)x(t-\theta)d\theta \quad 4.26$$

Then

$$Z_T[y(t)](\tau,\nu) = \frac{1}{\sqrt{T}}\int_0^{T} Z_T[h(t)](\theta,\nu)Z_T[x(t)](\tau-\theta,\nu)d\theta \quad 4.27$$

This given formula represents a convolution in the Doppler domain for a specific delay τ. So, for any fixed delay, the convolution of the ZT of the individual signals with respect to their Doppler frequencies is equivalent to the ZT of the product of the two-time domain signals.

Proof:

$$\begin{aligned}
Z_T[y(t)](\tau,\nu) &= \sqrt{T}\sum_n y(\tau+nT)e^{-j2\pi nT\nu} \\
&= \sqrt{T}\int_{-\infty}^{\infty} h(\theta)\sum_n x(\tau+nT-\theta)e^{-j2\pi nT\nu}d\theta \\
&= \int_{-\infty}^{\infty} h(\theta)Z_T[x(t)](\tau-\theta,\nu)d\theta \\
&= \sum_n \int_{nT}^{(n+1)T} h(\theta)Z_T[x(t)](\tau-\theta,\nu)d\theta \\
&= \sum_n \int_0^{T} h(\theta+nT)Z_T[x(t)](\tau-\theta \\
&- nT,\nu)d\theta \\
&= \int_0^{T}\sum_n h(\theta+nT)e^{-j2\pi nT\nu}Z_T[x(t)](\tau \\
&- \theta,\nu)d\theta, from 4.27 \\
&= \frac{1}{\sqrt{T}}\int_0^{T} Z_T[h(t)](\theta,\nu)Z_T[x(t)](\tau-\theta,\nu)d\theta \quad 4.28
\end{aligned}$$

ZT is fully described by the values within a specific rectangular area because of its quasi-periodic characteristics.

$$R = \left\{\tau \in [0, T), \nu \in \left[0, \frac{1}{t}\right)\right\} \quad 4.29$$

The fundamental region is denoted by R, and the Zak Transform restricted to this region is $\underline{Z}_T[x(t)](\tau, \nu)$, where $\underline{Z}_T[x(t)](\tau, \nu) = 0$ for $(\tau, \nu) \notin R$.

Now,

$$\begin{aligned} Z_T[x(t)](\tau, \nu) &= \sum_n \sum_m [x(t)](\tau - nT, \nu - m\Delta f) e^{j2\pi nT(\nu - m\Delta f)} \\ &= \sum_n \sum_m \bar{Z}_T[x(t)](\tau - nT, \nu - m\Delta f) e^{j2\pi nT\nu} \end{aligned} \quad 4.30$$

4.2.1.7 The inverse Zak Transform (IZT)

The time domain signal x(t) can be recovered from its ZT representation using the IZT.

$$x(t) = \sqrt{T} \int_0^{\Delta f} Z_T[x(t)](\tau, \nu) d\nu \, , -\infty < \tau < \infty \quad 4.31$$

To understand the equation, we use equation 4.30 and apply the properties defined in equation 4.16 and 4.19. This approach explains how the inverse Zak Transform leverages specific characteristics and relationship of the Zak Transform to accurately reconstruct the original time-domain signal $x(t)$,

$$x(t) = \sqrt{T} \sum_{n=-\infty}^{\infty} \int_0^{\Delta f} \bar{Z}_T[x(t)](\tau + nT, \nu) e^{j2\pi nT\nu} d\nu = \sum_{n=-\infty}^{\infty} x(\tau + nT) \quad 4.32$$

4.3 Multicarrier interpretation of OTFS modulation

This approach allows OTFS to be added as a straightforward pre-processing step to OFDM. The core idea rests on the principle of Fourier duality, which links a grid in the DD domain to a corresponding grid in the TF domain, ensuring a smooth transition between these two representations.

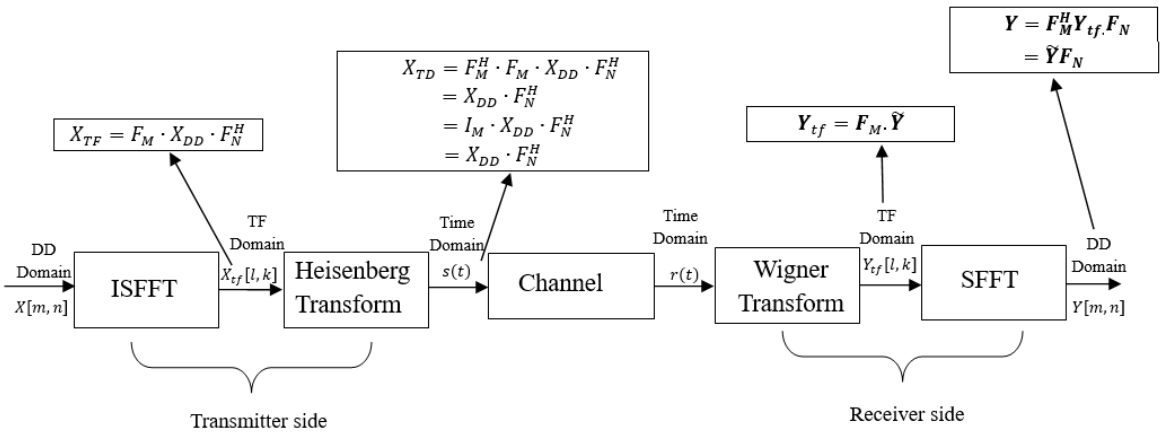


Figure 4.3: OTFS system Implementation

We consider an OTFS system designed to function over a high-mobility channel that includes P distinct paths. It operates with a bandwidth B, and is designed to handle a maximum delay spread τ_{max}, and maximum Doppler shift ν_{max}. An OTFS frame is divided into N blocks, each containing M samples, making up NM total samples. Each block lasts for a duration T, and the entire frame spans a total of $T_f = NT$. Each block in the OTFS frame utilizes the full bandwidth B, and the spectrum of each individual sample within a block is $\Delta f = \frac{1}{T}$. According to [1], the sampling rate for a continuous time OTFS signal is equal to $f_s = B = \frac{1}{T_s}$, where T_s is the sampling duration. Given that $B = M\Delta f = \frac{1}{T_s}$, this results in $T_s = \frac{1}{M\Delta f} = \frac{T}{M}$. Consequently, the entire frame duration can be calculated $T_f = NT = NMT_s$ representing the total time required to transmit one OTFS frame.

Every T second, we perform M-point DFT on each block, producing spectrum samples spaced at $\Delta f = \frac{1}{T}$. By compiling the spectra from all N blocks, each with bandwidth $B = M\Delta f$, we map out the OTFS frame in the discrete TF domain, depicted as an $M \times N$ array of points, as illustrated in the left side of the figure 4.3.

$$\Lambda = \{(l\Delta f, kT), l = 0, \cdots, M-1, k = 0, \cdots, N-1\}, M, N > 0 \qquad 4.33$$

At the points defined by Λ, the discrete TF samples are located and are organized into the matrix $X_{tf}[l,k], l = 0,\cdots,M-1, k = 0,\cdots,N-1$, where each column holds spectrum samples from each block. These discrete TF samples can be transformed into the DD domain using 2D SFFT, resulting in an $M \times N$ array of points as depicted in the corresponding figure 4.3 on the right.

$$\Gamma = \left\{\left(\frac{m}{\mathrm{M\Delta f}}, \frac{n}{NT}\right), m = 0,\cdots,M-1, n = 0,\cdots,N-1\right\}, \quad 4.34$$

The resolutions for path delays and Doppler shifts in the system are $\frac{1}{\mathrm{M\Delta f}}$ and $\frac{1}{NT}$, respectively. This means that if two paths have the same Doppler shift but differ in propagation by less than $\frac{1}{\mathrm{M\Delta f}}$, the receiver cannot distinguish between them. Similarly, paths with the same propagation delay but differing in Doppler shift by $\frac{1}{NT}$ are also indistinguishable to the receiver.

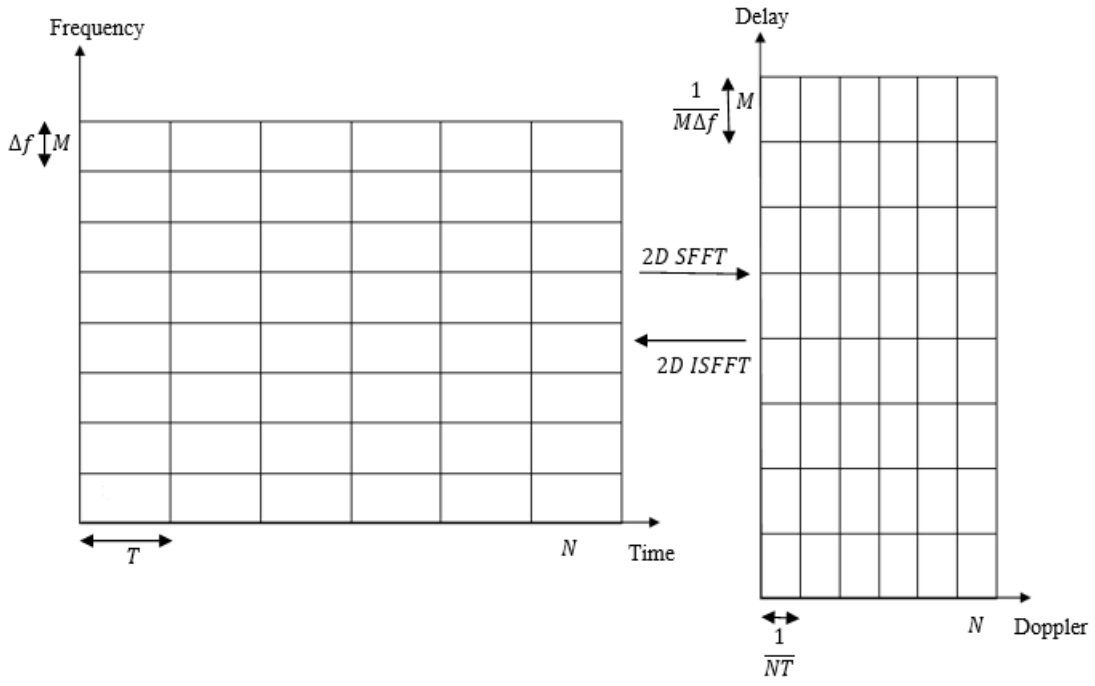


Figure 4.4: Transformation between the time-frequency and delay-Doppler domain using SFFT

The relationship between two grids is captured using the Symplectic finite Fourier Transform (SFFT), which is two-dimensional in nature. The inverse of this transform, the ISFFT, converts an $N \times M$ delay-doppler matrix $X[m,n]$ into a corresponding time-frequency matrix $X_{tf}[l,k]$ as,

$$X_{tf}[l,k] = \frac{1}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m-0}^{M-1} X[m,n] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)} \quad 4.35$$

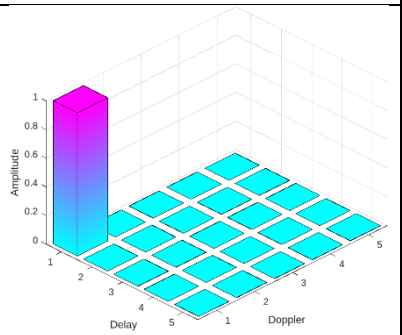
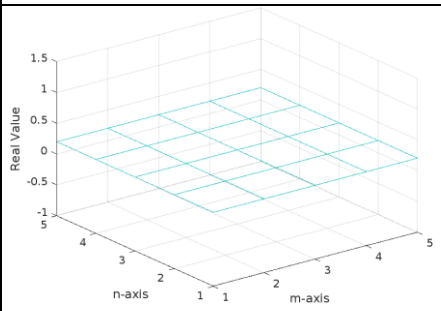
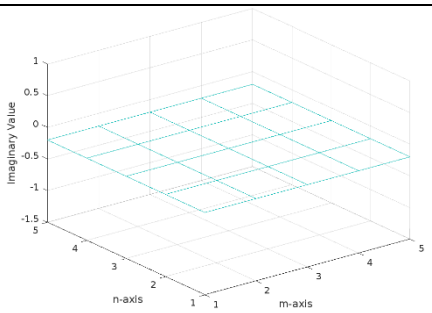
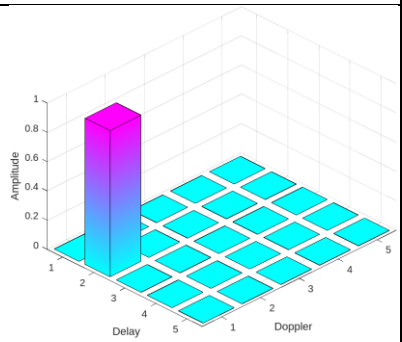
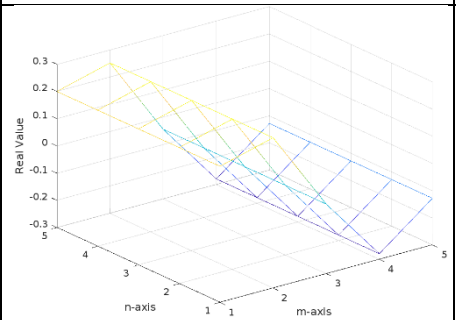
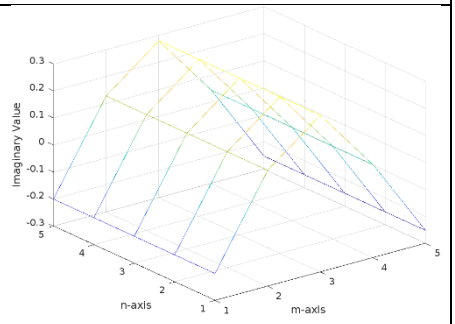
Considering the indices $l = 0, \cdots, M-1, k = 0, \cdots, N-1$, where $X_{tf} \in \mathbb{C}^{M \times N}$ represents the matrix of transmitted samples in the TF domain. The ISFFT performs a 2D transformation, where an M-point DFT is applied to each column of X, followed by an N-point IDFT on each row.

Next, a designated transmit waveform $g_{tx}(t)$ is used by the TF modulator to convert the 2D samples $X_{tf}[l,k]$ into a continuous-time waveform s(t). This process is called Heisenberg transform.

$$s(t) = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} X_{tf}[l,k] g_{tx}(t - kT) e^{j2\pi l \Delta f (t-kT)} \quad 4.36$$

The multicarrier view of OTFS suggests that you can take a delay-Doppler matrix (size $N \times M$) and map it to the time domain in two steps. First, the matrix is converted to a TF grid using the SFFT. Then, an IFFT is performed on the columns of the current matrix to transform it into the time domain as a series of M multicarrier symbols. This method shows that OTFS can act as an additional layer of pre-processing and post-processing over a standard multicarrier system, like the ones used in OFDM. Essentially, OTFS integrates seamlessly with existing frameworks, enhancing their capability to handle fluctuations in both time and frequency domains.

The multicarrier interpretation views OTFS as a method where each DD symbol is spread across a two-dimensional sequence on a time-frequency grid. This approach is like two-dimensional CDMA, using orthogonal 2D complex exponential codewords. According to figure 4.5, when we draw an analogy between OTFS and 2D CDMA, the codewords are orthogonal 2D complex exponentials. In the DD domain, each symbol is spread across the entire time-frequency (TF) plane before transmission. This dispersion means that every DD domain symbol experience the entire channel fluctuations within the TF channel, offering a chance for achieving full channel diversity. Furthermore, the periodic variations of symbol values in the TF domain, depending on the values of k and l, suggest the transition from the DD domain to the TF domain functions similarly to CDMA.

	DD domain	TF domain (Real Part)	TF domain (Imaginary part)
K=1 L=1			
K=2 L=1			

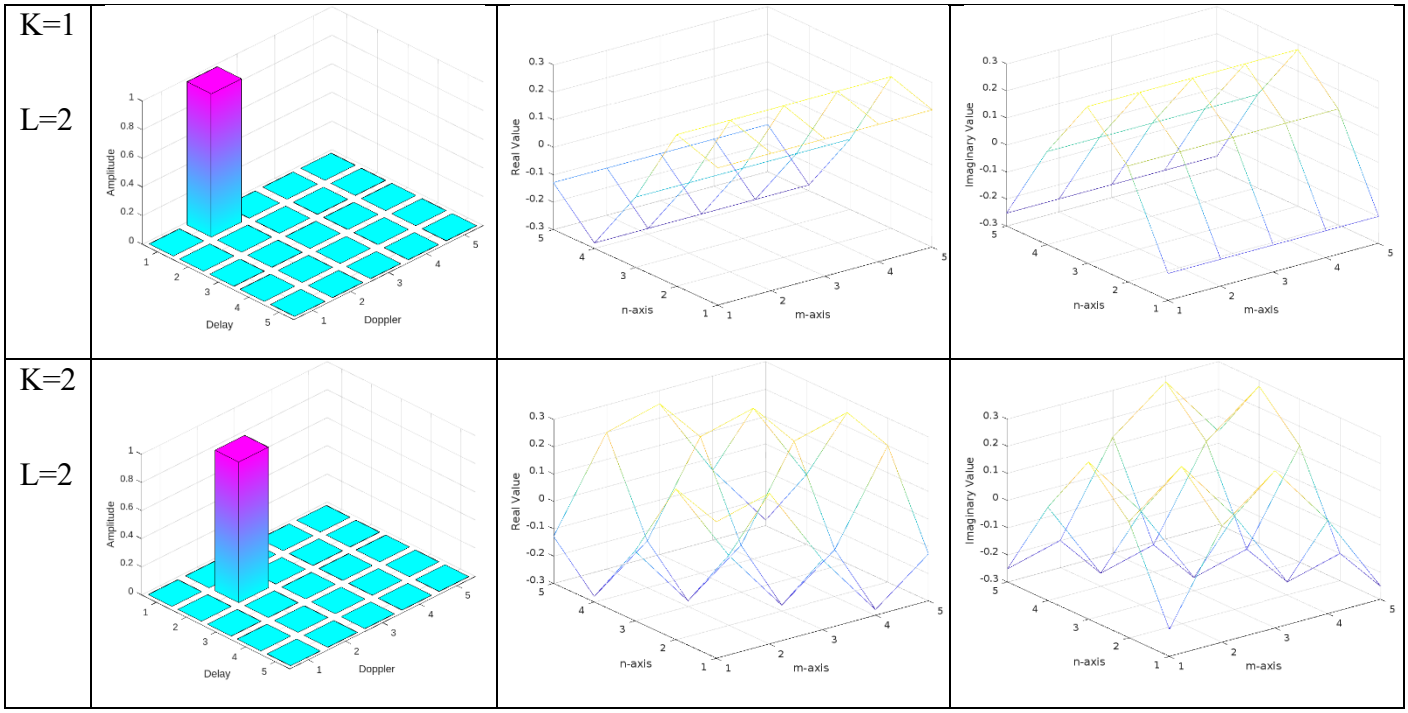
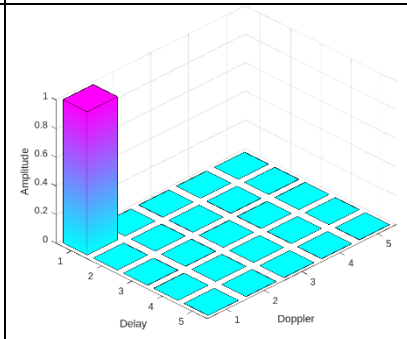
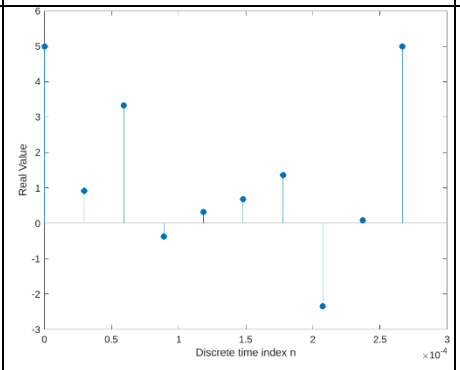
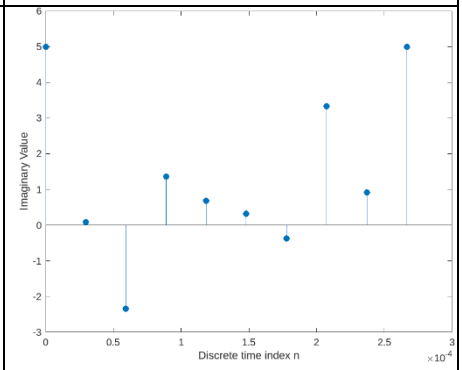


Figure 4.5: A visual representation of symbols in the DD domain and their equivalent OTFS symbols in the TF domain.

The placement of the information symbols on the DD grid significantly influences the behavior in the Time Domain. The sample values of OTFS signals in the time domain exhibit periodic changes that are directly associated with the Doppler index of the corresponding symbol in the DD domain.

There is a distinct link between the placement of symbols in the DD domain and their corresponding behavior in the time domain. Small changes in the delay indices of the information symbol in the DD domain result in corresponding time shifts in the time domain.

	DD domain	Time domain (Real Part)	Time domain (Imaginary part)
K=1 L=1			

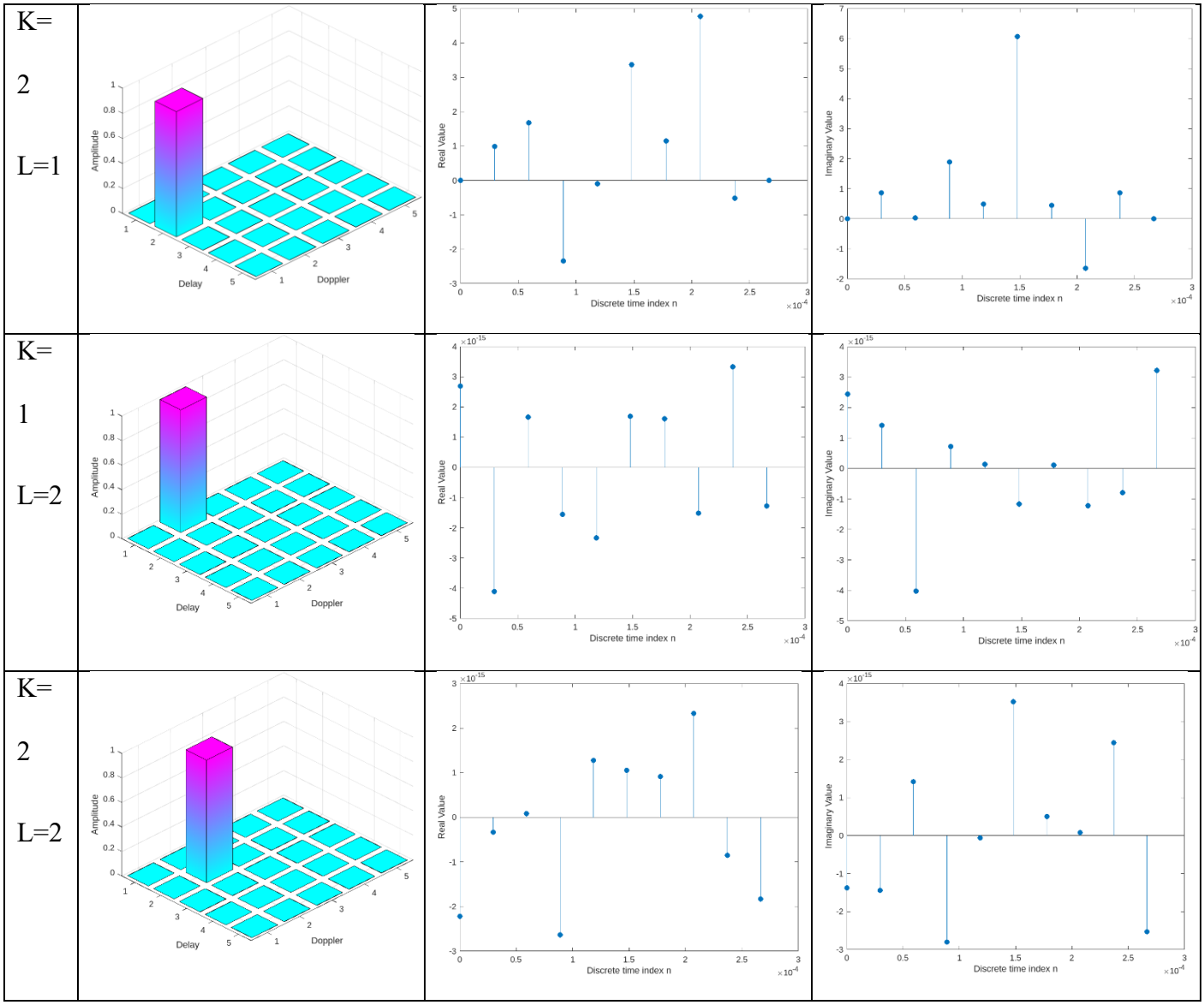


Figure 4.6: A visual representation of symbols in the DD domain and their equivalent OTFS symbols in the Time domain.

The OTFS signal in the time domain acts similarly to TDMA with localized pulses, exhibits global properties similar to OFDM due to its orthogonality in the time-frequency domain, and utilizes two-dimensional spreading in the delay-Doppler domain, much like CDMA.

4.4 OTFS modulation: Transmitter

The grid has MN information symbols $x[k,l]$ across M delay and N Doppler points, spaced by $\Delta\tau$ and $\Delta\nu$ respectively, totalling MN points in the fundamental rectangular region. A

localized pulse is placed at the specific grid point $(m\Delta\nu, n\Delta\tau)$ within the fundamental domain's boundaries, repeating quasi-periodically across the delay-Doppler plane. These symbols are typically QAM and are converted from the DD to the time domain using the ISFFT transform

$$X_{tf}[l,k] = \frac{1}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m-0}^{M-1} X[m,n] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)} \quad 4.37$$

where $n = 0, \dots, N-1, m = 0, \dots, M-1$.The TF modulator transforms the samples $X_{tf}[l,k]$ into a continuous time waveform $s(t)$ by using the transmit waveform $g_{tx}(t)$ as,

$$s(t) = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} X_{tf}[l,k] g_{tx}(t-kT) e^{j2\pi l\Delta f(t-kT)} \quad 4.39$$

,

4.5 High-mobility channel distortion

Gaussian white noise $u[t]$ is added to the transmitted signal after it passes through the channel $h(\tau,\nu)$ to obtain the received signal $r(t)$. After the noise is removed, the received signal $r(t)$ is expressed as,

$$\begin{aligned} r(t) &= \iint h(\tau,\nu) s(t-\tau) e^{j2\pi\nu(t-\tau)} d\tau d\nu \\ &= \int g(\tau,t) s(t-\tau) d\tau \end{aligned} \quad 4.40$$

Here, the transmitted signal is denoted by s(t), the delay-Doppler DD response over a time-varying channel is denoted by $h(\tau,\nu)$, and the delay time DT response of the channel is denoted by $g(\tau,t)$,

$$g(\tau,t) = \int_{\nu} h(\tau,\nu) e^{j2\pi\nu(t-\tau)} d\nu \quad 4.41$$

At the receiver, the signal received is discretized and then, sampled at time $t = qT_s = qT/M$ for $q = 0, \dots, MN-1$, and $\tau = lT_s = lT/M$ for $l = 0, \dots, M-1$. Following discretization 4.2 transforms to,

$$r[q] = \sum_{l} g^{s}[l,q]s[q-l] \tag{4.42}$$

From [1] the discrete delay-time response $g^{s}[l,q]$ is,

$$g^{s}[l,q] = \sum_{l\in L}\left(\sum_{k\in K_l} v_l(k) z^{k(q-l)}\right) sinc(l-\ell) \tag{4.43}$$

Where $z = e^{\frac{j2\pi}{NM}}$, and $v_\ell(k)$ indicates the Doppler response for a delay shift $\ell t/M$. The set L includes unique normalized delay shifts l within the channel. The set K_ℓ contains normalized Doppler shifts k associated with paths having the same delay shift $\ell T/M$.

Now if a signal $s(t)$ reaches receivers via P distinct paths, each path $(i-th\ path\ , i = 1, \dots, P)$, in a multipath channel is characterized by a channel gain g_i, delay shift τ_i, and Doppler shift v_i, the Doppler response at a specific delay shift $l = l_i = \tau_i/M\Delta f$ is determined as

$$v_l(k) = \begin{cases} g_i, & if\ l = l_i = \dfrac{\tau_i}{M\Delta f}\ and\ k = k_i = \dfrac{v_i}{NT} \\ 0, & otherwise \end{cases} \tag{4.44}$$

For integer normalized delay shifts l_i and the Doppler shifts k_i, the $sinc$ function becomes a unit pulse at l_i resulting in a discrete delay-time multipath channel response.

$$\mathrm{g}^{s}[l,q] = \sum_{i=1}^{P} v_{l_i}(k_i) z^{k_i(q-l)}\delta[l-l_i] \ = \sum_{i=1}^{P} g_i z^{k_i(q-l)}\delta[l-l_i] \tag{4.45}$$

Here, $l_{max} < M$. The delay-Doppler response is considered constant for an OTFS frame duration, implying fixed channel parameters g_i, τ_i, and v_i over $T_f = NT$.

4.6 OTFS modulation: Receiver

The signal r(t) received at the receiver is processed using a matched filter, which calculates the cross-ambiguity function $A_{g_{rx,r}}(f,t)$

$$Y(f,t) = A_{g_{rx,r}}(f,t) \triangleq \int r(t') g_{rx}^{*}(t'-t) e^{-j2\pi f(t'-t)} dt' \tag{4.46}$$

And then, sampling $Y(f,t)$ at the grid points Λ generates the received samples matrix in the TF domain, denoted as $Y_{tf}[l,k] = Y(f,t)$

Now the Wigner transform is used to translate a signal $r(t)$ in time domain into a symbol $Y[m,n]$. This translating is achieved by deriving cross-ambiguity function, which is done by applying a matched filter $g_{rx}^{*}(t)$ with respect to the transmit pulse $g_{tx}(t)$.

$$Y(t,f) = \int g_{rx}^{*}(t'-t)r(t')e^{-j2\pi f(t'-t)}dt' \tag{4.47}$$

The cross-ambiguity function $Y(f,t)$ of the received signal within the TF domain is sampled at time intervals $t = nT$ and frequency intervals $f = m\Delta f$. This process generates the $Y[n,m]$ in the TF domain corresponding to the output of the matched filter.

$$Y_{tf}[l,k] = Y(f,t)|_{f=l\Delta f,\ t=kT} \tag{4.48}$$

The ISFFT is then applied to convert the symbols $Y[n,m]$ into the DD domain following the Wigner transform resulting in,

$$y[m,n] = \frac{1}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} Y_{tf}[l,k] e^{-j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)} \tag{4.49}$$

4.7 Matrix formulation for OTFS

4.7.1 Transmitter

Here, several complex mathematical relationships of OTFS, involving modulation, demodulation, and input-output relationship across various domains, are simplified into a matrix form. By utilizing two normalized DFT matrices, F_M and F_N, with dimensions $M \times M$ and $N \times N$ respectively, the SFFT is defined. This enables the transformation of the DD symbol matrix X_{DD} to the TF domain matrix X_{TF}.

$$X_{TF} = F_M \cdot X_{DD} \cdot F_N^H \tag{4.50}$$

Here, the M-point Fourier transform FT is denoted by F_M and the inverse Fourier transform IFT by F_N^H . Then, the M-point IFFT is used to convert the TF domain matrix to the delay-time DT domain samples matrix as,

$$\begin{aligned} X_{TD} &= F_M^H \cdot F_M \cdot X_{DD} \cdot F_N^H \\ &= I_M \cdot X_{DD} \cdot F_N^H \\ &= X_{DD} \cdot F_N^H \end{aligned} \quad 4.51$$

In this context, I_m denotes $M \times M$ identity matrix. The final mirrors an IDZT. After pulse shaping the delay-time matrix X_{TD} at the transmitter and applying row-wise vectorization, a time domain vector of length NM is obtained as,

$$s = vec(G_{tx} \cdot X_{TD}) \in C^{NM \times 1} \quad 4.52$$

Here, the diagonal matrix G_{tx} has the samples of $g_{tx}(t)$ entries,

$$G_{tx} = diag\left[g_{tx}(0), g_{tx}\left(\frac{T}{M}\right), \cdots, g_{tx}\left(\frac{(M-1)T}{M}\right)\right] \in C^{M \times M} \quad 4.53$$

And if a rectangular matrix is used, G_{tx} becomes an identity matrix of size $M \times M$.

$$G_{tx} = I_M \quad 4.54$$

4.7.2 Time domain input-output relation

With the noise term added, the output $r[q]$ becomes,

$$r[q] = \sum_l g^s[l,q]s[q-l] + w[q], \;\; q = 0, \dots, NM-1 \quad 4.55$$

Here if $q < l, s[q-l] = 0$, it implies that the delay l is larger than the current time index q, as it is not possible as it would mean the receiver is getting a signal from the future, hence output is set to zero for these terms. $g^s[l,q]$ is the time-invariant impulse response as seen from 4.3

The relationship between the vectorized input and output can be expressed as follows,

$$r = G.s + w \quad 4.56$$

Here, G belongs to the complex $NM \times NM$ space, s represents the transmitted vector, and w is the time domain AWGN vector, which belongs to complex $NM \times 1$ space.

The discrete-time delay response of each block, which experiences no delay ($l = 0$), is positioned in the diagonal elements of a matrix. The responses of blocks that experience a delay are placed in the sub-diagonal elements, immediately to the left of the first diagonal element.

Table 4.1 shows the structure of a channel matrix designed to represent a communication channel with three distinct delay paths ($l = 0,1,2, where\ l_{max} = 2$) and $N = 8$ blocks in a system. Within this matrix, $s_n \in C^{M\times 1}$ and $r_n \in C^{M\times 1}$ symbolize the transmitted and received time vectors, respectively. Each vector consists of M elements, where M refers to the number of symbols or data points in each block. This matrix arrangement illustrates that any received block is subjected to interference from the immediately preceding block, denoted as the $n-1th$ block.

$$r_n = G_{0,n}.s_n + G_{1,n}.s_{n-1} + G_{2,n}.s_{n-2} + w_n, \quad 1 \le n \le N-1 \tag{4.57}$$

For n=0,

$$r_0 = G_{0,0}.s_0 + w_0, \tag{4.58}$$

r_1	$G_{0,0}$								s_1
r_2	$G_{0,0}$	$G_{0,1}$							s_2
r_3	$G_{2,2}$	$G_{1,2}$	$G_{0,2}$						s_3
r_4		$G_{2,3}$	$G_{1,3}$	$G_{0,3}$					s_4

r_5			$G_{2,4}$	$G_{1,4}$	$G_{0,4}$				s_5
r_6				$G_{2,5}$	$G_{1,5}$	$G_{0,5}$			s_6
r_7					$G_{2,6}$	$G_{1,6}$	$G_{0,6}$		s_7
r_8						$G_{2,7}$	$G_{1,7}$	$G_{0,7}$	s_8

Table 4.1: Matrix G, which illustrates the channel characteristics, includes three delay paths along the sub diagonals and is divided into M-by-M subcarrier segments.

The channel matrix's arrangement highlights the impact of the channel's delay paths on the signal transmission. The main diagonal of the matrix will represent the direct channel influence (delay $l = 0$) on each block, indicating the path with the least delay. The first sub-diagonal denotes the effect of the delay path ($l = 1$) on the signal. Similarly, the second sub-diagonal reflects the impact of the delay path corresponding to ($l = 2$).

4.7.3 Transmitter

At the receiver, the incoming NM sample vectors are rearranged as

$$r = \begin{bmatrix} r_0 \\ \vdots \\ r_{N-1} \end{bmatrix} = \{r[q], q = 0, \cdots, NM - 1\}, \tag{4.59}$$

Each component of r_n is a member of $\mathbb{C}^{M\times 1}$, for $n = 0, \ldots, N - 1$. Subsequently, the vector r is transformed into the delay-time matrix $\widetilde{\mathrm{Y}} \in \mathbb{C}^{M\times 1}$. Then the receiver pulse shaping matrix G_{rx} is then applied to the received vector r to obtain the matrix $\widetilde{\mathrm{Y}}$.

$$\widetilde{\mathrm{Y}} = G_{rx} \cdot \left(vec_{M,N}^{-1}(r)\right), \tag{4.60}$$

The operation $vec_{M,N}^{-1}(r)$ converts the complex vector $r \in \mathbb{C}^{MN\times 1}$ into a $M \times N$ matrix. The matrix G_{rx} is a diagonal matrix. It is defined as,

$$G_{rx} = diag\left[g_{rx}(0), g_{rx}\left(\frac{T}{M}\right), \cdots, g_{rx}\left(\frac{(M-1)T}{M}\right)\right] \in \mathbb{C}^{M\times M} \quad 4.61$$

When using rectangular pulse shaping waveforms and $G_{rx} = I_M$ $(the\ identity\ matrix)$ the relationship simplifies to,

$$\widetilde{\mathbf{Y}} = vec_{M,N}^{-1}(r) \ \ or\ r = vec\left(\widetilde{\mathbf{Y}}\right) \quad 4.62$$

Next, the received samples matrix in the TF domain, $Y_{tf} \in \mathbb{C}^{M\times 1}$, is generated by applying an M-point DFT to the DT delay-time samples, which can be expressed as, Following this, the time-frequency domain received samples matrix $Y_{tf} \in \mathbb{C}^{M\times N}$ is produced by performing M-point DFT on the delay-time samples, formulated as

$$\boldsymbol{Y}_{tf} = \boldsymbol{F}_M.\widetilde{\boldsymbol{Y}} \quad 4.63$$

Following this, delay-Doppler domain symbols are retrieved by applying an SFFT. Then, the SFFT and the Wigner transform are applied sequentially and summarized by the equation below.

$$\boldsymbol{Y} = \boldsymbol{F}_M^H \boldsymbol{Y}_{tf}.\boldsymbol{F}_N \quad 4.64$$

CHAPTER 5

Performance analysis of OTFS and OFDM in high mobility and static conditions

5.1 Introduction

In this chapter, we delve into the comprehensive simulation and analysis of OTFS modulation compared to OFDM. As the demand for high-speed, reliable communication continues to grow, particularly in environments with high mobility, there is a critical need to evaluate and understand the performance of advanced modulation schemes like OTFS. This chapter provides a detailed examination of how OTFS performs under various conditions, highlighting its potential as a future-proof technology for beyond 5G networks.

5.2 Objectives

The primary objectives of this chapter are:

- **Performance comparison under varying time slots N:**

  To analyze and compare the BER performance of OTFS and OFDM under varying Doppler shift conditions, reflecting high mobility scenarios.

- **Impact of subcarrier numbers**

  To analyze how varying the number of subcarriers affects the BER performance of both OTFS and OFDM, providing insights into the scalability and flexibility of these modulation schemes.

- **Performance in static channel conditions**

  To assess the BER performance of OTFS and OFDM in static channel conditions, including AWGN, Rayleigh, and Rician channels, thereby understanding their robustness in different propagation environments.

- **BER performance with varying speeds**

To investigate the BER performance of OTFS and OFDM at different mobility speeds while maintaining constant transmission power, demonstrating OTFS's capability to maintain reliable communication in high-speed scenarios.

- **Futureproofing beyond 5G**

  To establish OTFS as a viable and superior alternative to OFDM for beyond 5G networks, particularly in high mobility conditions, due to its enhanced robustness and efficiency.

These objectives aim to showcase the potential of OTFS to revolutionize wireless communication, providing a more reliable and efficient modulation technique in high mobility and diverse channel conditions. Through detailed simulations and rigorous analysis, this chapter seeks to substantiate OTFS as the modulation scheme of choice for the future of wireless communication.

5.3 Simulation environment and tools

The simulations were conducted using MATLAB, a powerful tool for numerical computing and algorithm development. MATLAB's robust set of functions and toolboxes facilitated the implementation and analysis of both OTFS and OFDM modulation schemes under various conditions.

5.3.1 Simulation parameters

The following parameters were used in the simulations:

- General Simulation Parameters:
  - Number of Frames: 1000 (Monte Carlo iterations)
  - Modulation Type: BPSK, QPSK, and 16QAM
  - Sub-carrier Spacing: 15 kHz
  - Number of Sub-carriers: 64

- Carrier Frequency (f_c): 2 GHz
- Transmission Power: From 5 dBm to 40 dB
- Noise Variance: 0 dBm (directly defined noise variance)

➢ OTFS-Specific Parameters:

- Number of Time Symbols: 32
- CP Length :16

➢ OFDM-Specific Parameters:

- CP Length: 0.25 * Number of Sub-carriers

➢ Channel Conditions:

- Channel Types: AWGN, Rayleigh, Rician, Pedestrian, Vehicular, Train [1]
- User Equipment Speed: Variable to simulate different mobility conditions

5.3.2 Derived parameters

➢ Bit and Symbol Calculations:

- Derived based on modulation type, number of sub-carriers, symbols, and frames.

➢ Bandwidth (BW):

- Calculated as the product of the number of sub-carriers and sub-carrier spacing.

➢ Noise Calculations:

- Adjusted based on noise type and bandwidth.

These parameters and configurations ensure a comprehensive simulation environment for comparing OTFS and OFDM under various conditions, particularly focusing on high mobility scenarios.

5.4 Simulation and analysis

In this section, we present the results of our simulations and provide an in-depth analysis of the performance of OTFS compared to OFDM. The simulations were carried out under

various conditions to comprehensively evaluate the robustness and efficiency of OTFS, especially in high mobility scenarios. The analysis is divided into four parts based on different simulation conditions: varying time slots N, varying subcarrier numbers, static channel conditions, and varying speeds with constant transmission power.

5.4.1 Performance comparison under varying time slots N

The objective of this section is to analyze and compare the BER performance of OTFS modulation and OFDM under different Doppler shift conditions. These conditions reflect high mobility scenarios that are typical in environments such as trains and high-speed vehicles.

Simulation Setup:

- Number of frames: 1000
- Modulation type: BPSK, QPSK, 16 QAM
- Number of sub-carriers: 64
- Number of time symbols for OTFS: [8, 16, 32, 64]
- Channel model
    - Channel Type: Train Scenario
    - User Equipment Speed: 1000 Km/h
- Train scenario channel parameters
    - Delays: [0, 50, 120, 200, 230, 500, 1600, 2300, 5000] ns
    - Power Delay Profile: [-1, -1, -1, 0, 0, 0, -3, -5, -7] dB

5.4.1.1 Analysis

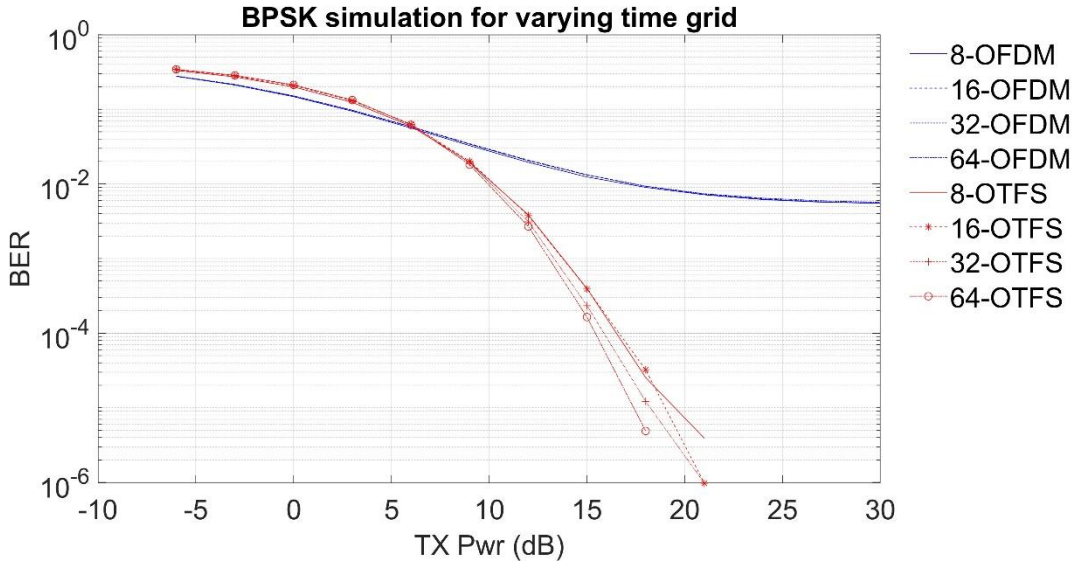


Figure 5.1: BPSK simulation for varying values of N

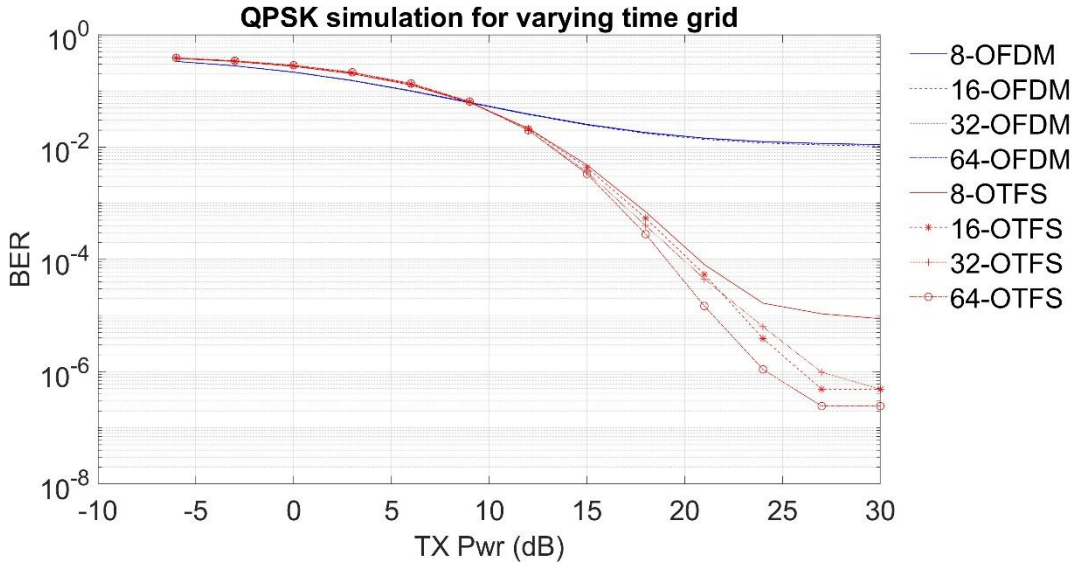
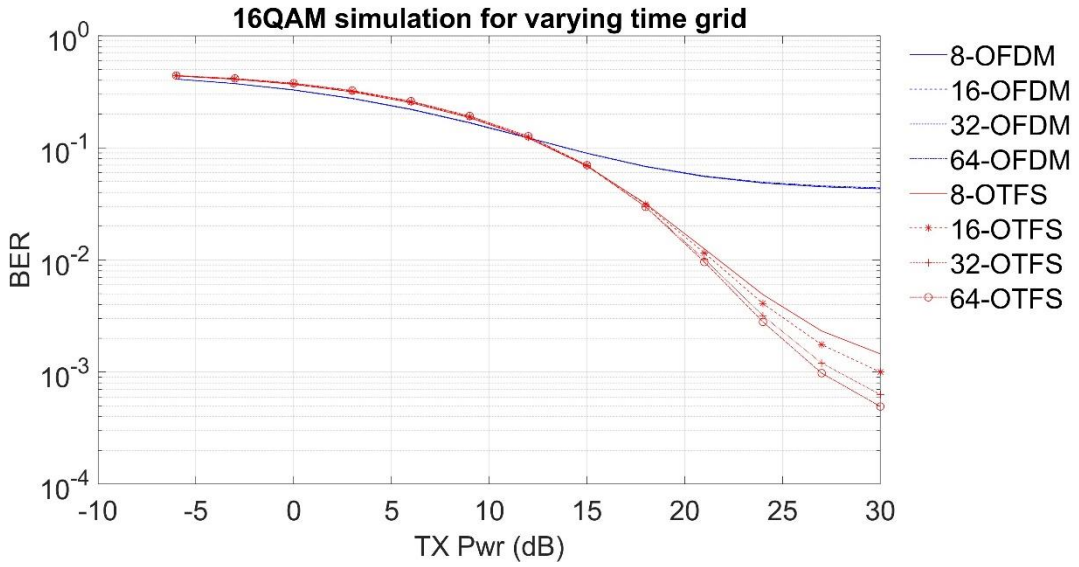


Figure 5.2: QPSK simulation for varying values of N



5.3: BPSK simulation for varying values of N

The provided plots highlight the BER performance of BPSK, QPSK, and 16 QAM modulation schemes with OTFS and OFDM under different values of N and transmission power levels. For each modulation scheme, OTFS consistently outperforms OFDM, especially as the transmission power increases.

- **Steeper decline in BER with increasing TX power**

  For OTFS, the BER decreases significantly more rapidly with increasing transmission power compared to OFDM. This is evident across all modulation schemes. For instance, in the BPSK modulation at 0 dB, the BER for 64-OTFS is approximately 10^{-1}, but it drops sharply to almost 10^{-6} at $20\ dB$. This steep is much more pronounced than in OFDM, which only drops from 10^{-1} to 10^{-2} over the same range.

- **Performance at low TX power**

  At low transmission power levels, the curves show that OFDM performs slightly better than OTFS. However, if we examine the y-axis values in this region, it becomes evident that the difference in BER performance is minimal. This observation is crucial because these differences occur at BER values greater than 0.1. For any practical

communication system, a BER value above 0.1 is considered very high and unacceptable for uncoded BER performance. In real-world systems, the target is typically an uncoded BER on the order of 10^{-4} or lower. Therefore, the minor advantage of OFDM at low transmission power levels is not significant from a practical standpoint. The more meaningful observation is the significant outperformance of OTFS over OFDM at moderate to high transmission power levels. As the TX power increases, OTFS shows a much steeper decline in BER, achieving substantially lower error rates compared to OFDM. This makes OTFS a more robust and reliable choice for real systems that operate in the low BER regime.

- **Modulation scheme independence**

  OTFS consistently outperforms OFDM across all modulation schemes (BPSK, QPSK, 16QAM). This consistency underscores the flexibility and reliability of OTFS in various modulation environments. For instance, in the 16 QAM simulation, at 30 dB, 64-OTFS achieves a BER of approximately 10^{-4}, whereas 64 OFDM remains around 10^{-1}.

- **Sub Carrier configuration benefits**

  Additionally, the number of time grids has a significant impact on OTFS performance. As the number of time grid increases, the BER for OTFS decreases more sharply compared to OFDM. For instance, in the QPSK simulation, 64-OTFS achieves a BER around 10^{-6} at $30\ dB$, a marked improvement over 8-OTFS, which has a BER around 10^{-4}. This pattern holds true across all modulation schemes, indicating that OTFS effectively leverages higher time grids to enhance its robustness and efficiency in communication. These findings suggest that OTFS, with its superior performance and ability to exploit time-frequency resources, is well suited for next-generation communication systems requiring high reliability and low error rates.

5.4.1.2 Discussion

- The results clearly indicate that OTFS provides better BER performance compared to OFDM across different values of N and modulation schemes (BPSK, QPSK, 16QAM).
- One important finding is that the number of time grids has a significant impact on OTFS performance. As the number of time grids increases, the BER for OTFS decreases more sharply compared to OFDM.
- The ability of OTFS to maintain low BER in high mobility conditions makes it a promising candidate for future high-speed communication systems.
- The improved BER performance of OTFS can be attributed to its ability to better manage the time-frequency diversity in the channel, making it more resilient to the impairments that typically degrade OFDM performance.

5.4.2 Impact of subcarrier numbers

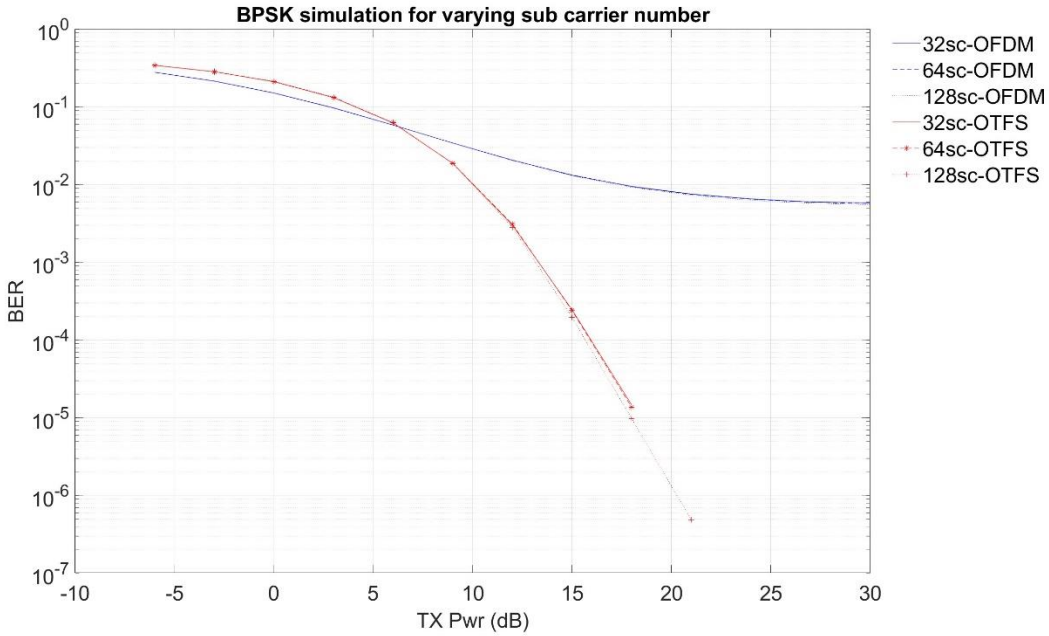
This section evaluates how varying the number of subcarriers affects the BER performance of OTFS modulation and OFDM. This analysis aims to provide insights into the scalability and flexibility of these modulation schemes in handling different subcarrier configurations.

Simulation parameters:

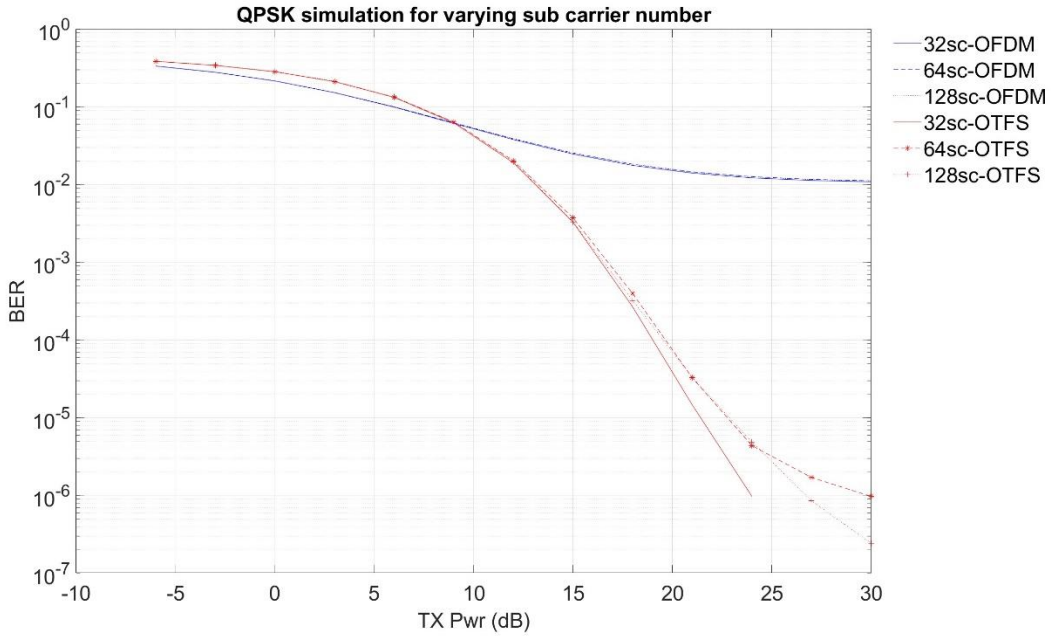
- Modulation types: BPSK, QPSK, and 16QAM
- Number of sub-carriers: 32, 64, and 128
- Carrier frequency: 2 GHz
- Transmission power: From -10 dB to 30 dB
- Channel model
    - Channel Type: Train Scenario
    - User Equipment Speed: 1000 Km/h
- Train scenario channel parameters

- Delays: [0, 50, 120, 200, 230, 500, 1600, 2300, 5000] ns
- Power Delay Profile: [-1, -1, -1, 0, 0, 0, -3, -5, -7] dB

5.4.2.1 Analysis



5.4: BPSK simulation for varying subcarrier numbers



5.5: BPSK simulation for varying subcarrier numbers

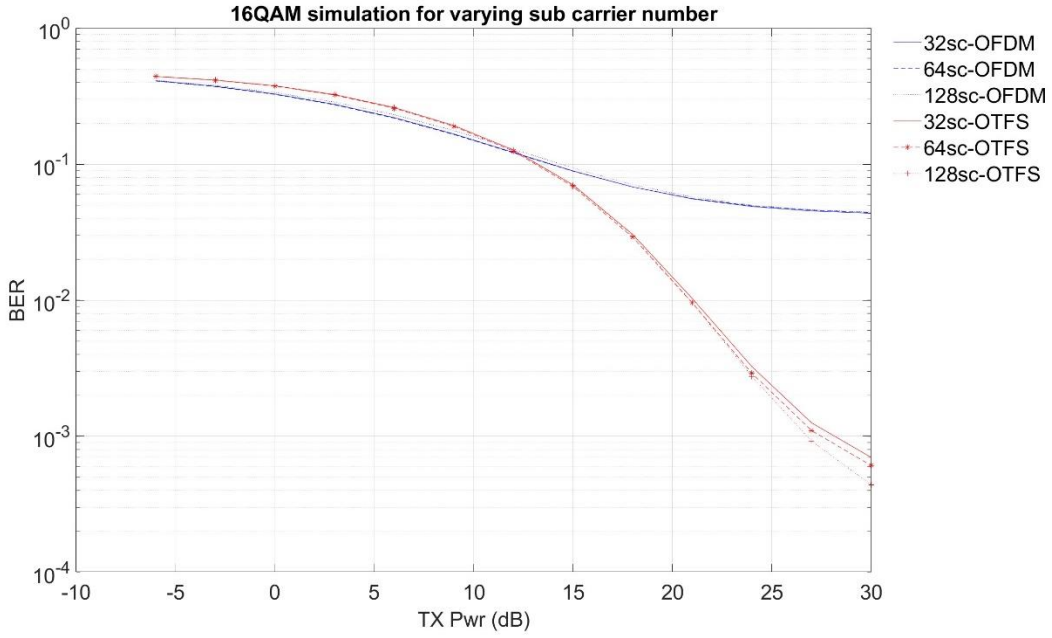


Figure 5.6: 16QAM simulation for varying subcarrier numbers

The provided plot showcases the BER performance of different modulation schemes – BPSK, QPSK, and 16 QAM under varying subcarrier numbers for both OTFS and OFDM. The trend across all three plots indicates that as the number of subcarriers increases, the BER for OTFS consistently decreases, outperforming OFDM. For example, with 128 subcarriers in the QPSK,

OTFS achieves a BER approximately 10^{-3} at $17\ dB$ and 10^{-6} at $30\ dB$, while OFDM maintains higher BER values around 0.1 and 0.01 at $17\ dB$ and 0.01 at $30dB$. Even at low transmission power levels, where OFDM shows slightly better performance, the difference is minimal and occurs at BER values greater than 0.1, which are not practical for real systems. In practical scenarios, systems typically operate at BER values on the order of 10^{-4} or lower. These results underscore OTFS's superior performance across different subcarrier configurations, making it robust and reliable candidate for modern communication systems that require low BER and high efficiency.

5.4.2.2 Discussion

- The results clearly indicate that OTFS provides better BER performance compared to OFDM across varying subcarriers number (BPSK, QPSK, 16QAM)
- The scalability and flexibility of OTFS in handling higher subcarrier counts make it a promising candidate for high-speed communication systems.
- The improved BER performance of OTFS can be attributed to better managing the time-frequency diversity in the channel, making it more resilient to the impairments that typically degrade OFDM performance.

5.4.3 Performance in static channel conditions

In this section we assess and compare the BER of OTFS and OFDM in static channel conditions. The static channels considered include Additive White Gaussian Noise (AWGN), Rayleigh, and Rician Channels.

Parameters set up:

- Number of frames: 1000
- Transmission power: [-6dB - 30dB]
- Modulation types: BPSK, QPSK, 16QAM

- Carrier frequency: 2 GHz
- Sub-carrier spacing: 15 kHz
- Channel type (Static): AWGN, Rayleigh, Rician
- Channel parameters for AWGN channel
    - Number of taps: 1
- Channel parameters for Rayleigh channel
    - Number of taps: 5
- Channel parameters for Rician channel
    - Rician K-factor: 2
    - Number of taps: 5

5.4.3.1 Analysis

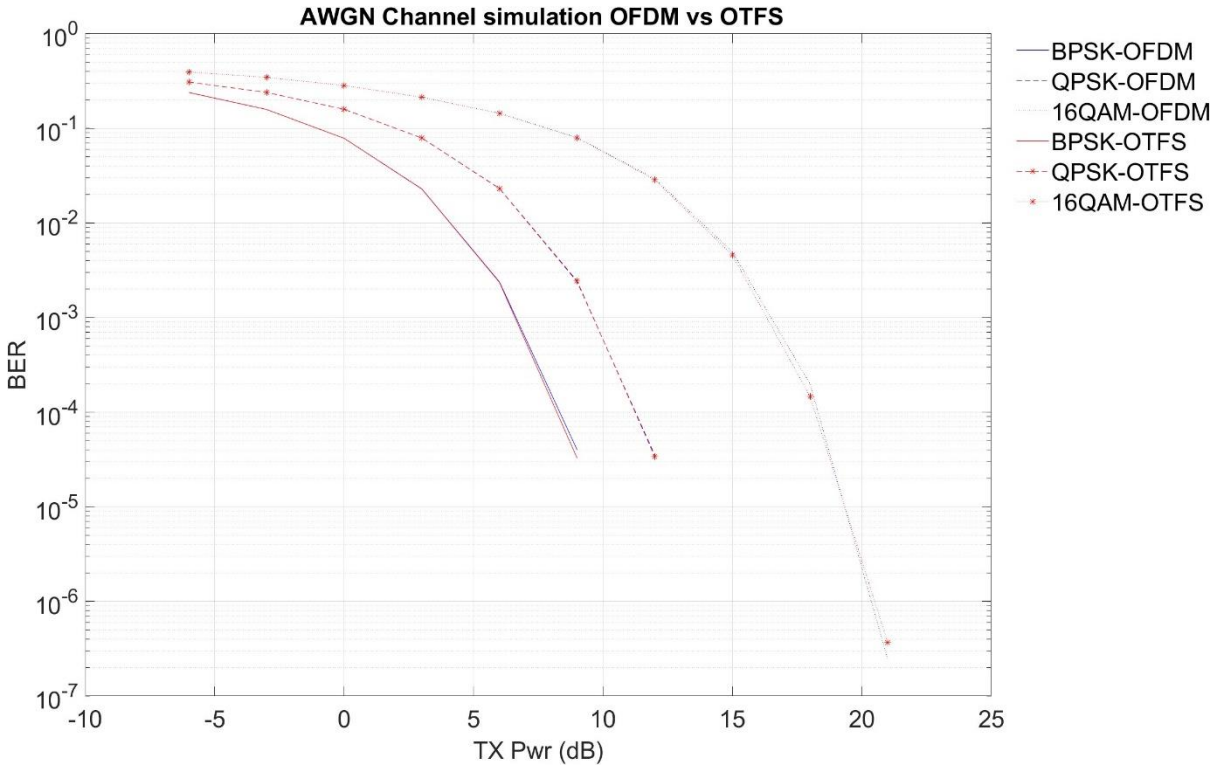


Figure 5.7: AWGN channel simulation OFDM vs OTFS

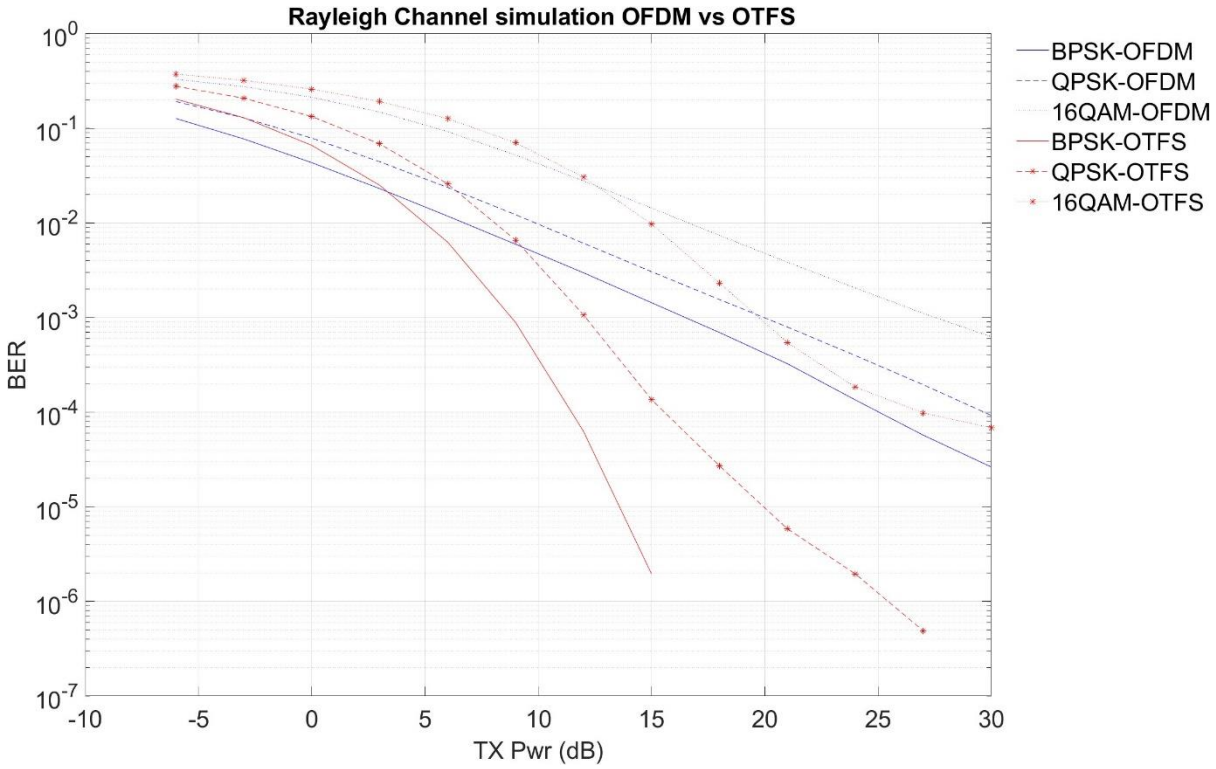


Figure 5.8: Rayleigh Channel simulation for OFDM vs OTFS

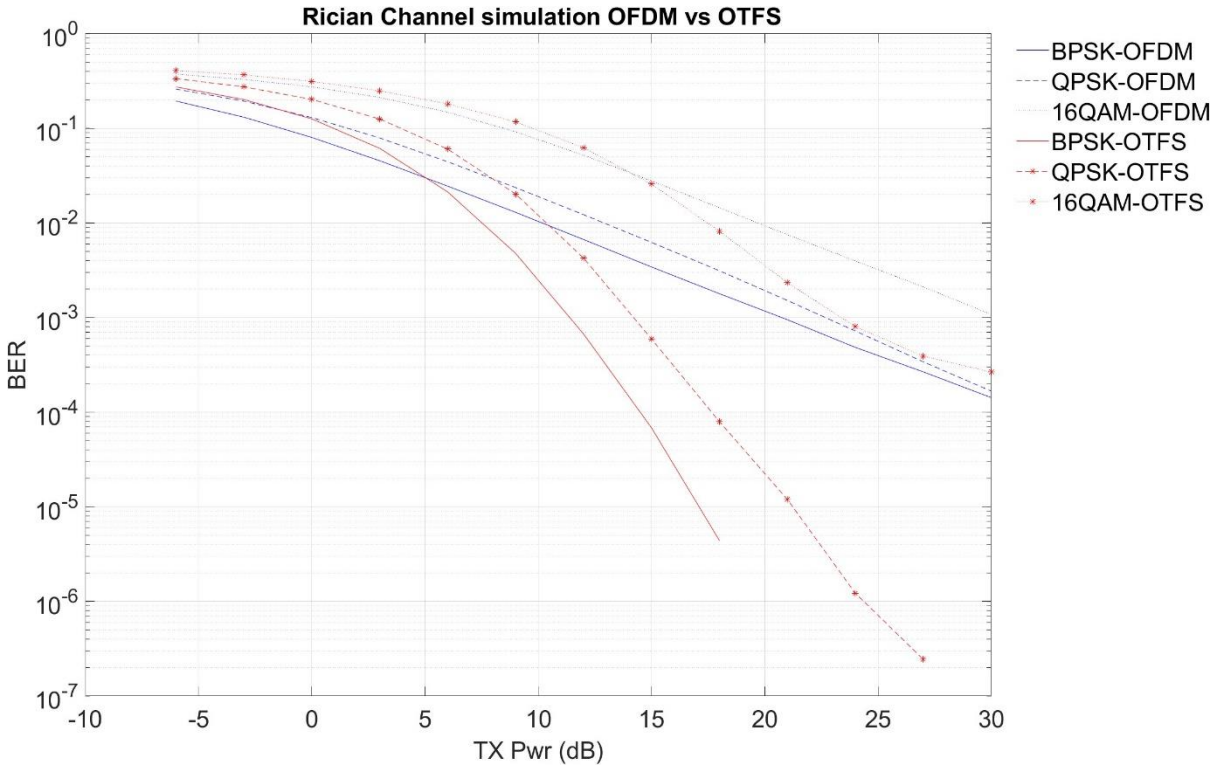


Figure 5.9: Rician Channel simulation OFDM vs OTFS

In the AWGN channel simulation, OTFS and OFDM curves follow similar trends.

In the Rayleigh channel, the performance gap widens. BPSK-OTFS achieves BER greater than 10^{-5} at $15\ dB$, while BPSK-OFDM lags in between 0.01 and 0.001. QPSK-OTFS reaches 10^{-5} at $20\ dB$, in contrast to QPSK-OFDM's 10^{-3}. The 16 QAM results are particularly striking with OTFS attaining BER greater than 10^{-4} at $20\ dB$, compared to OFDM's 10^{-3}.

The Rician channel also shows OTFS superiority. BPSK-OTFS hits 10^{-4} at $10\ dB$, whereas BPSK-OFDM is at 10^{-3}. For QPSK, OTFS achieves 10^{-5} at $20\ dB$, outperforming OFDM's 10^{-4}. Lastly, 16QAM-OTFS achieves almost 10^{-4} at $25\ dB$, significantly better than OFDM's 10^{-3}. These results illustrate OTFS's robust performance in static channel conditions across different modulation schemes.

5.4.3.2 Discussion

- OTFS operates in the delay-Doppler domain, providing enhanced resilience against time and frequency selectivity, which is beneficial in static channel conditions. This improved performance of OTFS in static channels can be attributed to the delay-Doppler domain.
- The significant difference in BER performance, especially in Rayleigh and Rician channels suggests that OTFS can exploit full diversity gains offered by these multipath channels, leading to better performance in terms of BER compared to OFDM.
- OTFS has superior mechanisms for managing inter-symbol and inter-carrier interference, especially in multipath environments, resulting in lower BER for the same transmission power.

The results from the Rayleigh and Rician channel simulations which are representative of multipath environments show a clear performance advantage for OTFS. This is likely due to OTFS's ability to exploit the multipath effects more effectively than OFDM, leading to lower BERs.

5.4.4 Performance of varying speeds with constant transmission power

This section investigates the BER performance of OTFS modulation and OFDM at different mobility speeds while maintaining constant transmission power. This analysis aims to demonstrate OTFS's capability to maintain reliable communication in high-speed scenarios, highlighting its potential advantages over OFDM in dynamic environments.

Parameters Set up:

- Modulation types: QPSK, and 16QAM
- Number of sub-carriers: 64
- Number of time symbols for OTFS: [16, 32, 64]
- Carrier frequency: 2 GHz
- Transmission power: 30 dB
- Channel model
    - Channel type: Train Scenario
    - User equipment speed: $[0 - 1000]$ Km/h
- Train scenario channel parameters
    - Delays: $[0, 50, 120, 200, 230, 500, 1600, 2300, 5000]$ ns
    - Power delay profile: $[-1, -1, -1, 0, 0, 0, -3, -5, -7] dB$

5.4.4.1 Analysis

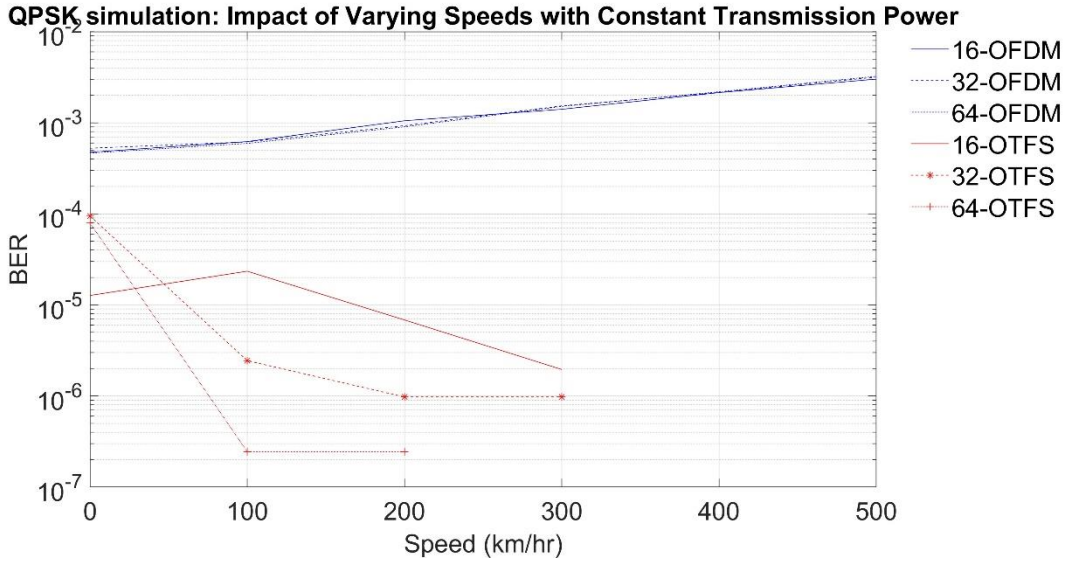


Figure 5.10: QPSK simulation: Impact of Varying Speeds with Constant Transmission Power

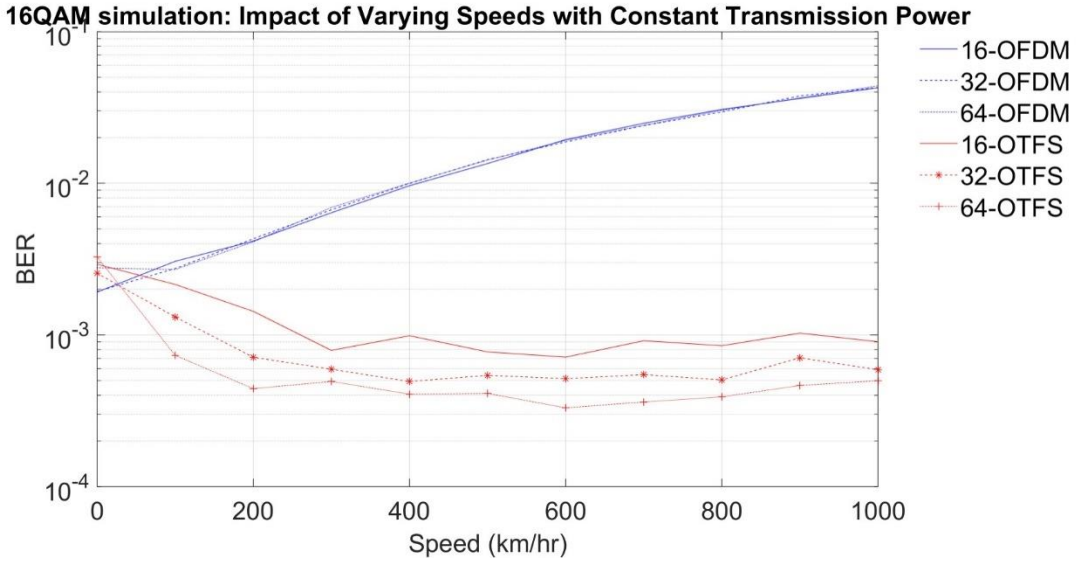


Figure 5.11: 16QAM: Impact of Varying Speeds with Constant Transmission Power

From the analysis of the impact of varying speeds with constant transmission power for both 16QAM and QPSK simulations, a crucial observation is that the performance of OTFS improves as the number of Doppler bins (time grid) increases. Specifically:

- **Enhanced performance with increased time slots N**: As the number of N increases from 16 to 32 to 64, the BER for OTFS consistently decreases, indicating better

performance. This trend is evident across all speed ranges, including low, medium, and high speeds.

- **Low and medium speed ranges**: In the lower speed range ($0 - 100\ km/hr$), the BER performance for OFDM (16QAM) shows a slight increase, with all remaining within 10^{-3} range, while the OTFS datasets demonstrate better performance, with BER decreasing and stabilizing below 10^{-3}. In the medium speed range ($100 - 500 km/hr$), the BER for all OFDM schemes gradually increases, with 16-OFDM and 32-OFDM reaching around 10^{-2} by $500 km/hr$, whereas the OTFS datasets maintain a lower and more stable BER, indicating superior performance compared to OFDM in both lower and medium speed conditions.
- **High-speed performance**: At high speeds (500-1000 km/h for 16QAM and 300-500 km/h for QPSK), the performance advantage of OTFS with increased Doppler bins becomes even more pronounced. The BER remains stable and low, highlighting the robustness of OTFS in high-mobility environments.

5.4.4.2 Discussions

Resilience to Mobility: OTFS shows superior resilience to varying speeds compared to OFDM. The lower and more stable BER for OTFS across all speed ranges indicates better performance in high-mobility scenarios.

- Interference management: The improved performance of OTFS, especially at higher speeds, suggests better handling of Doppler effects and inter-carrier interference, which are more pronounced in mobile environments.
- Performance consistency: The consistency of OTFS's performance across different speeds and subcarrier configurations highlights its robustness and reliability compared to OFDM, which shows a significant degradation in BER as speed increases.

- Adaptability to high-speed scenarios: The lower BER at higher speeds (up to 1000 km/h for 16QAM and 500 km/h for QPSK) indicates that OTFS is well-suited for high-speed communication scenarios, such as high-speed trains or vehicular networks, where OFDM may struggle to maintain performance.

5.4.5 Constellation diagram: OTFS vs OFDM

This section analyzes and compares the constellation diagrams for QPSK and BPSK modulations using OTFS and OFDM. The purpose is to highlight the advantages of OTFS over OFDM in terms of noise resilience and signal integrity, particularly in challenging high-mobility and multipath conditions, thereby making a strong case for the adoption of OTFS in modern wireless communication systems.

The constellation diagrams for BPSK and QPSK modulations reveal that OTFS offers superior

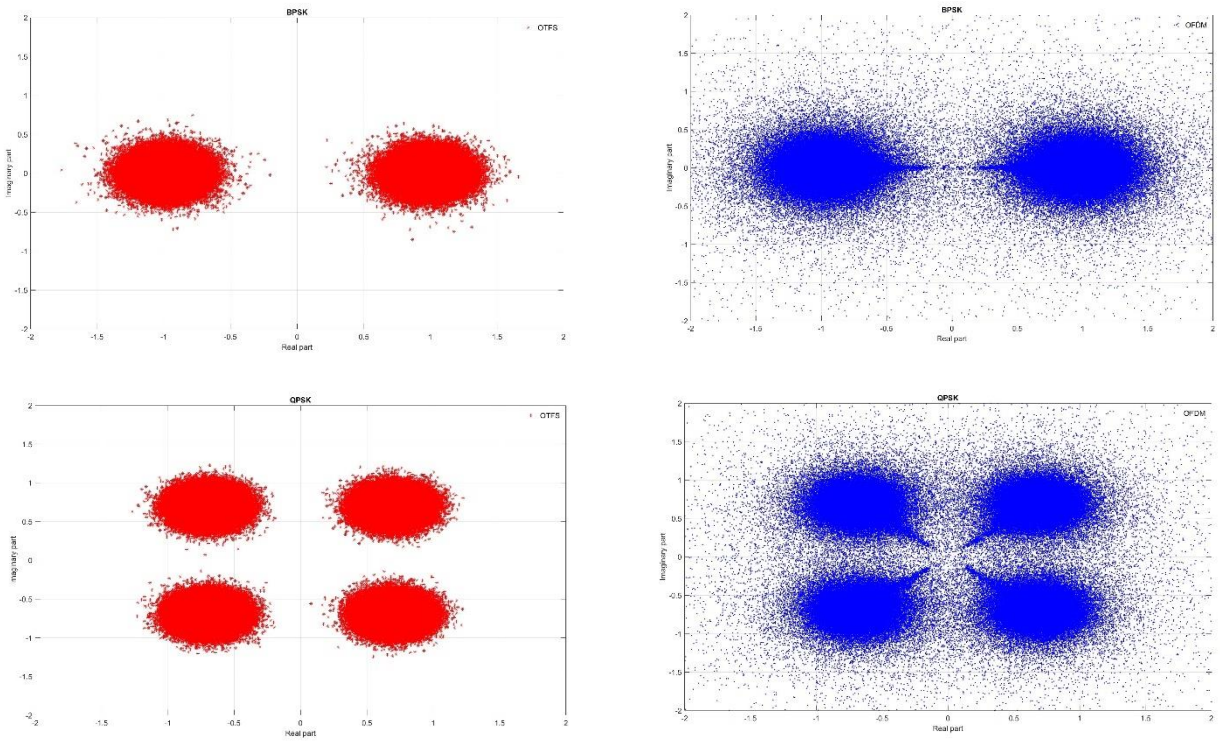


Figure 5.13: Constellation diagram OTFS vs OFDM

performance compared to OFDM. In BPSK modulation, OTFS maintains closer alignment with the ideal points $(\pm 1, 0)$, while OFDM shows significant dispersion into the neighboring region. Similarly, for the QPSK diagrams, OTFS shows a tight clustering of points around the

ideal positions $(\pm 1, \pm 1)$, indicating high resilience to noise and minimal distortion. Overall, the symbols in the OTFS plot are more compact and less scattered demonstrating better noise resilience and signal integrity, making it a more robust choice for high-mobility and multipath environments. In contrast, the OFDM diagram exhibits a more dispersed spread of points into the neighboring region due to the high mobility of the channel, reflecting higher susceptibility to noise and channel impairments.

5.5 Summary and conclusion

5.5.1 Summary

In this chapter, we conducted a comprehensive simulation and analysis to compare the performance of Orthogonal Time Frequency Space (OTFS) modulation with Orthogonal Frequency Division Multiplexing (OFDM). The simulations were designed to evaluate their performance under various conditions, including varying time grids (Doppler bins), varying subcarrier numbers, static channel conditions, and varying speeds with constant transmission power.

5.5.2 Conclusion

The simulation results demonstrate that OTFS offers significant advantages over OFDM in various scenarios, particularly under high mobility conditions and with varying subcarrier numbers. The ability of OTFS to leverage time-frequency diversity more effectively results in lower BER and enhanced robustness across different channel conditions.

- BER performance of OTFS vs OFDM under varying values of N
    - OTFS consistently outperformed OFDM for all values of N and modulation schemes (BPSK, QPSK, 16QAM).

- The value of N significantly impacts OTFS performance. As the number of time grids increases, the BER for OTFS decreases more sharply compared to OFDM, demonstrating OTFS's ability to utilize time-frequency diversity effectively.

- BER Performance under varying subcarrier numbers:
  - OTFS showed superior BER performance compared to OFDM for different subcarrier numbers and modulation schemes.
  - The scalability and flexibility of OTFS in handling higher subcarrier counts were evident, making it a robust choice for high-speed communication systems.
- Performance in static channel conditions (AWGN, Rayleigh, and Rician):
  - OTFS consistently provided better BER performance than OFDM in AWGN, Rayleigh, and Rician channels.
  - The enhanced robustness of OTFS in different channel conditions makes it a promising candidate for future wireless communication systems.
- BER performance with varying speeds and constant transmission power:
  - OTFS maintained superior BER performance compared to OFDM at different mobility speeds.
- The effectiveness of OTFS in high mobility conditions underscores its potential for reliable communication in dynamic environments.

Key takeaways:

- High mobility scenarios: OTFS outperforms OFDM in high Doppler environments, making it suitable for applications involving high mobility.
- Scalability: The performance of OTFS improves significantly with an increasing number of subcarriers and time grids, highlighting its scalability for future communication systems.

- Channel robustness: OTFS shows better resilience in AWGN, Rayleigh, and Rician channels, indicating its potential for diverse communication environments.
- Future potential: The superior performance of OTFS under various conditions positions it as a strong candidate for beyond 5G networks, where reliability and efficiency are paramount.

The findings from this chapter strongly support the adoption of OTFS as a modulation scheme for future wireless communication systems, offering better performance and reliability than traditional OFDM. This comprehensive analysis underscores the potential of OTFS to meet the increasing demands for high-speed, reliable communication in the evolving landscape of wireless technology.

Appendix A

A.1 Main file

```
clc;
clear all;
% close all;
rng(100);
% rng('shuffle');
%% Generic independent parameters
% sim
numFrames = 100;                % no. of monte carlo iterations      % ##
numOTFS_Sym_perFrame = 1;      % no of OTFS symbols per frame -- OFDM
determined based on this
IndPLot = 1;
fontSize = 14;
lineWidth = 3;
% Pwr
PwrTX_dB = [5 : 5 : 40];        % ##
% modulation parameters
modType = 'BPSK';       % BPSK || QPSK || 16QAM % ##
% spectral parameters
fc = 2e9;
deltaF = 20e3;
num_subCarriers = 64;
% noise parameters
noiseType = 0;      % 0 == directly define noise variance || 1 == define
based on BW
NoiseVar_dB = 00;
noiseFloor_dB_Hz = -204;
noiseFigure_dB = 5;
%% Generic derived parameters
% Pwr
PwrTX = 10.^(PwrTX_dB ./ 10);
% modulation
vecConstellation = {'BPSK' , 'QPSK' , '16QAM'};
vecM = [2 , 4 , 16]; % number of points in each constellation type
vecBits = log2(vecM);% Number of bits each symbol in each constellation
encodes
modInd = find( strcmp( vecConstellation , modType ) == 1 );
numSymbols_Constellation = vecM( modInd );
numBitsPerSymbol = vecBits( modInd );
% spectral parameters
T = 1 / deltaF;
BW = num_subCarriers * deltaF;
% noise parameters
if noiseType == 0
elseif noiseType == 1
    NoiseVar_dB = noiseFloor_dB_Hz + noiseFigure_dB + 10*log10( BW );  %
variance in dB Watts
else
    error('Define correct noise type!');
end
NoiseVar = 1 * 10.^( NoiseVar_dB / 10 ); % Watts
%% OTFS parameters
```

```
num_timeSym_OTFS = 32;       % ##
CP_length_OTFS = 16;
CP_reducedOrNot_OTFS = 0;
CP_zeroOrNot_OTFS = 0;
%% OFDM parameters
CP_length_OFDM = 0.25 * num_subCarriers;
CP_reducedOrNot_OFDM = 0;
CP_zeroOrNot_OFDM = 0;
%% derived OTFS and OFDM param
% Bits
numBitsPerFrame = numBitsPerSymbol * num_subCarriers * num_timeSym_OTFS *
numOTFS_Sym_perFrame;
numSymsPerFrame = numBitsPerFrame / numBitsPerSymbol;
numOFDM_Sym_perFrame = num_timeSym_OTFS * numOTFS_Sym_perFrame;
numBitsPerOFDMSym = numBitsPerSymbol * num_subCarriers;
numBitsPerOTFSSym = numBitsPerSymbol * num_subCarriers * num_timeSym_OTFS;
%% Channel parameters
channelType = 2;            % 0 == AWGN || 1 == Rayleigh || 2 == Rician || 3
== Vehicular || 4 == Pedestrian || 5 == Train  % ##
unitPwr_channelGen = 1;
UE_speed = 0;               % km per hour % ##
channelPropagationMode = 0; % 0 == convolution || 1 == channel matrix (time
variant channels)
%% Varaibles for saving
err_OTFS_perFrame = zeros( numel( PwrTX_dB ) , numFrames );
err_OFDM_perFrame = zeros( numel( PwrTX_dB ) , numFrames );
%% Main Loop
for indFrame = 1:numFrames
    % get channel realization
    lenOFDM = (num_subCarriers + CP_length_OFDM) * numOFDM_Sym_perFrame;
    if CP_reducedOrNot_OTFS         % reduced
        lenOTFS = (num_subCarriers * num_timeSym_OTFS) + CP_length_OTFS *
numOTFS_Sym_perFrame;
    elseif ~CP_reducedOrNot_OTFS    % regular
        lenOTFS = (num_subCarriers + CP_length_OTFS) * num_timeSym_OTFS *
numOTFS_Sym_perFrame;
    end
    if lenOFDM >= lenOTFS
        [channelStruct] = genChannelv2( channelType , deltaF , T , 1 ,
num_subCarriers , UE_speed , fc , CP_length_OFDM , numOFDM_Sym_perFrame ,
channelPropagationMode , CP_reducedOrNot_OFDM , CP_zeroOrNot_OFDM );
    else
        [channelStruct] = genChannelv2( channelType , deltaF , T ,
num_timeSym_OTFS , num_subCarriers , UE_speed , fc , CP_length_OTFS ,
numOTFS_Sym_perFrame , channelPropagationMode , CP_reducedOrNot_OTFS ,
CP_zeroOrNot_OTFS );
    end
    % Generate bits, channel and noise realization bits and mapping
    bitsTX = randi([0,1] , 1 , numBitsPerFrame)';
% bit generation
    symbolsTX_unitEnergy = qammod(bitsTX , numSymbols_Constellation , 'gray' ,
'InputType' , 'bit' , 'UnitAveragePower' , true);  % symbol mapping
    % noise
    noise = sqrt( NoiseVar/2 ) * ( randn( round(numSymsPerFrame * (1 +
max(CP_length_OFDM/num_subCarriers , CP_length_OTFS) )) , 1  ) + 1i*randn(
```

```
round(numSymsPerFrame * (1 + max(CP_length_OFDM/num_subCarriers ,
CP_length_OTFS) )) , 1 ) ); % noise generation
    for indPwr = 1:numel( PwrTX_dB )

        % Generate bits, channel and noise realization bits and mapping
        % bitsTX = randi([0,1] , 1 , numBitsPerFrame)';
% bit generation
        % symbolsTX = sqrt( PwrTX(indPwr) ) * qammod(bitsTX ,
numSymbols_Constellation , 'gray' , 'InputType' , 'bit' , 'UnitAveragePower' ,
true);  % symbol mapping
        symbolsTX = sqrt( PwrTX(indPwr) ) * symbolsTX_unitEnergy;
        % % noise
        % noise = sqrt( NoiseVar/2 ) * ( randn( round(numSymsPerFrame * (1 +
max(CP_length_OFDM/num_subCarriers , CP_length_OTFS) )) , 1  ) + 1i*randn(
round(numSymsPerFrame * (1 + max(CP_length_OFDM/num_subCarriers ,
CP_length_OTFS) )) , 1 ) ); % noise generation

        % OFDM
        [bitsRX_OFDM , symbolsRX_OFDM] = simOFDMv2(numOFDM_Sym_perFrame ,
num_subCarriers , symbolsTX , noise , channelType , CP_length_OFDM ,
numSymbols_Constellation , fc , deltaF , T , UE_speed ,
channelPropagationMode, CP_reducedOrNot_OFDM , CP_zeroOrNot_OFDM , 0*NoiseVar
, channelStruct );
        err_OFDM_perFrame( indPwr , indFrame ) = sum( bitsRX_OFDM ~= bitsTX )
/ numBitsPerFrame;

        % OTFS with freq domain single tap equalization
        % [bitsRX_OTFS , symbolsRX_OTFS] = simOTFSv2(numOTFS_Sym_perFrame ,
num_timeSym_OTFS , num_subCarriers , symbolsTX , noise , channelType ,
CP_length_OTFS , numSymbols_Constellation , fc , deltaF , T , UE_speed ,
channelPropagationMode , CP_reducedOrNot_OTFS , CP_zeroOrNot_OTFS , 0*NoiseVar
, channelStruct );
        % err_OTFS_perFrame( indPwr , indFrame ) = sum( bitsRX_OTFS ~= bitsTX
) / numBitsPerFrame;

        % OTFS with time domain channel reciprocity based equalization
        [bitsRX_OTFS , symbolsRX_OTFS] = simOTFSv3(numOTFS_Sym_perFrame ,
num_timeSym_OTFS , num_subCarriers , symbolsTX , noise , channelType ,
CP_length_OTFS , numSymbols_Constellation , fc , deltaF , T , UE_speed ,
channelPropagationMode , CP_reducedOrNot_OTFS , CP_zeroOrNot_OTFS , 0*NoiseVar
, channelStruct );
        err_OTFS_perFrame( indPwr , indFrame ) = sum( bitsRX_OTFS ~= bitsTX )
/ numBitsPerFrame;

        1;
    end
    1;
end
%% Final calculations
% OFDM
err_OFDM = mean( err_OFDM_perFrame , 2);
disp(err_OFDM);
% OTFS
err_OTFS = mean( err_OTFS_perFrame , 2);
disp(err_OTFS);
%% Final plots
```

```
figure;
semilogy( PwrTX_dB , err_OFDM , 'LineWidth', lineWidth , 'Color' , 'b' );
hold on;
semilogy( PwrTX_dB , err_OTFS , 'LineWidth', lineWidth , 'Color' , 'r',
'LineStyle','--' );
legend_vecType = {'OFDM' , 'OTFS'};
title( modType );
xlabel( 'TX Pwr (dB)' );
ylabel( 'BER' );
set( findall( gcf , '-property' , 'FontSize' ) , 'FontSize' , fontSize );
set( gcf , 'color' , [1 1 1] );
legend( legend_vecType , 'FontSize' , fontSize , 'Location' , 'northeast' ,
'Box' , 'off' );
grid on;
```

A.2 OTFS simulation

```
function [bitsRX , vecSymbolsRX] = simOTFSv2(numOTFS_Sym_perFrame ,
num_timeSym_OTFS , num_subCarriers , symbolsTX , noise , channelType ,
CP_length , numSymbols , fc , deltaF , T , UE_speed , channelPropagationMode,
CP_reducedOrNot , CP_zeroOrNot , NoiseVar , channelStruct )
% channelPropagationMode : 0 == through conv suitable for time invariant || 1
== for time variant channels through channel matrix
% channel
[channelStruct] = Wrapper_getDTBChannelv2(channelStruct , deltaF , T ,
num_timeSym_OTFS , num_subCarriers , UE_speed , fc , CP_length ,
numOTFS_Sym_perFrame , CP_reducedOrNot , channelType );
[channelStruct] = genChannelConvMatv2( channelType , deltaF , T ,
num_timeSym_OTFS , num_subCarriers , UE_speed , fc , CP_length ,
numOTFS_Sym_perFrame , channelPropagationMode , CP_reducedOrNot , CP_zeroOrNot
, channelStruct );
if channelPropagationMode == 0  % time invariant channel
    if channelStruct.timeVariant == 1
        channelPropagationMode = 1;
        warning(" Changing propagation mode to matrix because channel was time
variant! ");
    end
elseif channelPropagationMode == 1
else
    error( 'Enter valid propagation mode!' );
end
if channelPropagationMode == 0
    channelTaps_FD = fft( channelStruct.channelTaps , num_subCarriers ).' /
sqrt(num_subCarriers);          % FD channel for equalization
end
%
bitsRX = [];
vecSymbolsRX = [];
if CP_reducedOrNot          % reduced CP
    len_symOTFS = num_subCarriers*num_timeSym_OTFS + CP_length;
elseif ~CP_reducedOrNot     % after every TD symbol
    len_symOTFS = num_timeSym_OTFS * (num_subCarriers + CP_length);
end
NxM = num_timeSym_OTFS * num_subCarriers;
```

```
dftmtx_timeFreq = dftmtx(num_timeSym_OTFS * num_subCarriers) /
sqrt(num_timeSym_OTFS * num_subCarriers);
dftmtx_timeDim = dftmtx(num_timeSym_OTFS) / sqrt(num_timeSym_OTFS);
dftmtx_freqDim = dftmtx(num_subCarriers) / sqrt(num_subCarriers);
pulseShapingMat_TX = eye( num_subCarriers );
pulseShapingMat_RX = eye( num_subCarriers );
numChannelTaps = ( channelStruct.numTaps ) ;
convSpillOver = zeros( numChannelTaps-1 , 1 );
for indOTFS = 1:numOTFS_Sym_perFrame
    % OTFS modulation
    currSyms = reshape( symbolsTX( (indOTFS-1)*(num_subCarriers *
num_timeSym_OTFS) + 1 : indOTFS * (num_subCarriers * num_timeSym_OTFS) ) ,
num_subCarriers , num_timeSym_OTFS );
    currOTFSSymbol_timeFreq = dftmtx_freqDim * ( currSyms ) * dftmtx_timeDim';
% ISFFT
    currOTFSSymbol_delayTime = dftmtx_freqDim' * ( currOTFSSymbol_timeFreq );
% IDZT
    currOTFSSymbol_delayTime = pulseShapingMat_TX * currOTFSSymbol_delayTime;
    % Channel proapgation
    if channelPropagationMode == 0      % through convolution
        if CP_reducedOrNot              % reduced CP
            currOTFSSymbol = reshape( currOTFSSymbol_delayTime , [] , 1 );
% parallel to serial
            currOTFSSymbol = [(currOTFSSymbol(end-
CP_length+1:end)).^(CP_zeroOrNot == 0) ; currOTFSSymbol];     % CP
        elseif ~CP_reducedOrNot         % every TD symbol CP
            currOTFSSymbol = reshape( [(currOTFSSymbol_delayTime(end-
CP_length+1:end , :)).^(CP_zeroOrNot == 0) ; currOTFSSymbol_delayTime] , [] ,
1 );                % parallel to serial
        end
        currOTFSSymbol_RX = conv( currOTFSSymbol , channelStruct.channelTaps
);                                  % convolution
        % spill over
        currOTFSSymbol_RX( 1:numChannelTaps-1 ) = currOTFSSymbol_RX(
1:numChannelTaps-1 ) + convSpillOver;      % previous spill over
        convSpillOver = currOTFSSymbol_RX( end-numChannelTaps+2:end );
% next spill over
        % noise + ISI removal for next symbol
        currOTFSSymbol_RX = currOTFSSymbol_RX(1 : len_symOTFS) + noise(
(indOTFS-1)*(len_symOTFS) + 1 : indOTFS * (len_symOTFS) );
        % discard CP
        if CP_reducedOrNot              % reduced CP
            currOTFSSymbol_RX = currOTFSSymbol_RX( CP_length+1 : end );
            currOTFSSymbol_RX = reshape( [currOTFSSymbol_RX] , num_subCarriers
, num_timeSym_OTFS );                   % parallel to serial reverse
        elseif ~CP_reducedOrNot         % per TD CP
            currOTFSSymbol_RX = reshape( [currOTFSSymbol_RX] ,
num_subCarriers+CP_length , num_timeSym_OTFS );          % parallel to serial
reverse
            currOTFSSymbol_RX = currOTFSSymbol_RX( CP_length + 1 : end , : );
        end
    elseif channelPropagationMode == 1  % through convolution matrix for time
variance
        currOTFSSymbol = reshape( [ currOTFSSymbol_delayTime ] , [] , 1 );
% parallel to serial
```

```
        currOTFSSymbol_RX = sum( channelStruct.channelTaps_convMat( (indOTFS-
1)*(NxM ) + 1 : indOTFS * (NxM)  ...
                                                                , (indOTFS-
1)*(NxM ) + 1 : indOTFS * (NxM) ) .* [currOTFSSymbol].' , 2 );

        if CP_reducedOrNot
            currNoise = noise( (indOTFS-1)*(len_symOTFS) + 1 + CP_length :
indOTFS * (len_symOTFS) );
        elseif ~CP_reducedOrNot
            currNoise = reshape( noise( (indOTFS-1)*(len_symOTFS) + 1 :
indOTFS * (len_symOTFS) ) , num_subCarriers+CP_length , num_timeSym_OTFS );
            currNoise = reshape( currNoise(CP_length + 1:end , :) , [] , 1);
        end
        currOTFSSymbol_RX = currOTFSSymbol_RX + currNoise;   % noise
        currOTFSSymbol_RX = reshape( [currOTFSSymbol_RX] , num_subCarriers ,
num_timeSym_OTFS );        % parallel to serial reverse
    end
    % OTFS demodulation
    if CP_reducedOrNot              % reduced CP
        F = dftmtx( NxM ) / sqrt( NxM );
        currOTFSSymbol_RX = F * currOTFSSymbol_RX(:);
        currChannelEst = channelStruct.timeFreqMat(: , indOTFS);
    elseif ~CP_reducedOrNot         % per time
        F = dftmtx( num_subCarriers ) / sqrt( num_subCarriers );
        currOTFSSymbol_RX = F * currOTFSSymbol_RX;
        currChannelEst = channelStruct.timeFreqMat(: , : , indOTFS);
    end
    currOTFSSymbol_RX = ( conj( currChannelEst ) .* currOTFSSymbol_RX ) ./ ( (
conj( currChannelEst ) ) .* ( currChannelEst ) + NoiseVar );
    currOTFSSymbol_RX = F' * currOTFSSymbol_RX;
    % OTFS equalization
    currOTFSSymbol_RX_delayTime = pulseShapingMat_RX * reshape(
currOTFSSymbol_RX , num_subCarriers , num_timeSym_OTFS );       % reshape and
pulse shape
    currOTFSSymbol_RX_timeFreq = dftmtx_freqDim * (
currOTFSSymbol_RX_delayTime );                                          %
DZT
    currOTFSSymbol_delayDopp = dftmtx_freqDim' * ( currOTFSSymbol_RX_timeFreq
) * dftmtx_timeDim;                                 % SFFT
    currOTFSSymbol_delayDopp = currOTFSSymbol_delayDopp / ( norm(
currOTFSSymbol_delayDopp , 'fro' ) ) * sqrt(numel(currOTFSSymbol_delayDopp));
% rescaling to unit power
    currOTFSSymbol_delayDopp = currOTFSSymbol_delayDopp(:);
    bitsRX = [bitsRX; qamdemod(currOTFSSymbol_delayDopp , numSymbols ,  'gray'
, 'OutputType' , 'bit' , 'UnitAveragePower' , true)];                     %
demodulation
    vecSymbolsRX = [vecSymbolsRX; currOTFSSymbol_delayDopp];
end
end
```

A.3 OFDM simulation

```
function [bitsRX , vecSymbolsRX] = simOFDMv2(numOFDM_Sym_perFrame ,
num_subCarriers , symbolsTX , noise , channelType , CP_length ,
```

```
numSymbols_Constellation , fc , deltaF , T , UE_speed , channelPropagationMode
, CP_reducedOrNot , CP_zeroOrNot , NoiseVar , channelStruct )
% channelPropagationMode : 0 == through conv suitable for time invariant || 1
== for time variant channels through channel matrix
% channel
[channelStruct] = Wrapper_getDTBChannelv2(channelStruct , deltaF , T , 1 ,
num_subCarriers , UE_speed , fc , CP_length , numOFDM_Sym_perFrame ,
CP_reducedOrNot , channelType );
[channelStruct] = genChannelConvMatv2( channelType , deltaF , T , 1 ,
num_subCarriers , UE_speed , fc , CP_length , numOFDM_Sym_perFrame ,
channelPropagationMode , CP_reducedOrNot , CP_zeroOrNot , channelStruct );
if channelPropagationMode == 0  % time invariant channel
    if channelStruct.timeVariant == 1
        channelPropagationMode = 1;
        warning(" Changing propagation mode to matrix because channel was time
variant! ");
    end
elseif channelPropagationMode == 1
else
    error( 'Enter valid propagation mode!' );
end
if channelPropagationMode == 0
    channelTaps_FD = fft( channelStruct.channelTaps , num_subCarriers ).' /
sqrt(num_subCarriers);          % FD channel for equalization
end
%
bitsRX = [];
vecSymbolsRX = [];
len_symOFDM = num_subCarriers + CP_length;
numChannelTaps = ( channelStruct.numTaps ) ;
convSpillOver = zeros( numChannelTaps-1 , 1 );
for indOFDM = 1:numOFDM_Sym_perFrame
    % OFDM modulation
    currSyms = symbolsTX( (indOFDM-1)*num_subCarriers + 1 : indOFDM *
num_subCarriers );
    currOFDMSymbol = ifft( currSyms ) * sqrt(num_subCarriers);
% IFFT
    % Channel proapgation
    if channelPropagationMode == 0
        currOFDMSymbol = [ (currOFDMSymbol(end-
CP_length+1:end)).^(CP_zeroOrNot == 0); currOFDMSymbol];
% CP
        currOFDMSymbol_RX = conv( currOFDMSymbol , channelStruct.channelTaps
);                              % convolution
        % spill over
        currOFDMSymbol_RX( 1:numChannelTaps-1 ) = currOFDMSymbol_RX(
1:numChannelTaps-1 ) + convSpillOver;  % previous spill over
        convSpillOver = currOFDMSymbol_RX( end-numChannelTaps+2:end );
% next spill over
        % noise + ISI removal for next symbol
        currOFDMSymbol_RX = currOFDMSymbol_RX(1 : len_symOFDM) + noise(
(indOFDM-1)*(len_symOFDM) + 1 : indOFDM * (len_symOFDM) );
        currOFDMSymbol_RX = currOFDMSymbol_RX( CP_length+1 : end );
% discard CP
    elseif channelPropagationMode == 1
```

```
        currOFDMSymbol_RX = sum( channelStruct.channelTaps_convMat( (indOFDM-
1)*(num_subCarriers ) + 1 : indOFDM * (num_subCarriers)  ...
                                                                , (indOFDM-
1)*(num_subCarriers ) + 1 : indOFDM * (num_subCarriers) ) .*
[currOFDMSymbol].' , 2 );
        % noise
        currOFDMSymbol_RX = currOFDMSymbol_RX + noise( (indOFDM-
1)*(len_symOFDM) + 1 + CP_length : indOFDM * (len_symOFDM) );
    end
    % OFDM demodulation and equalization
    symbolsRX = fft( currOFDMSymbol_RX ) / sqrt(num_subCarriers);
% FFT
    % single tap frequency domain equalization
    if channelPropagationMode == 0
        symbolsRX = symbolsRX .* conj(channelTaps_FD) ./ ( channelTaps_FD .*
conj(channelTaps_FD) + NoiseVar );
    elseif channelPropagationMode == 1
        symbolsRX = symbolsRX .* conj( channelStruct.timeFreqMat(: , 1 ,
indOFDM) ) ./ ( channelStruct.timeFreqMat(: , 1 , indOFDM) .*
conj(channelStruct.timeFreqMat(: , 1 , indOFDM)) + NoiseVar );
    end
    symbolsRX = symbolsRX / ( sqrt(symbolsRX' * symbolsRX) ) *
sqrt(numel(symbolsRX));                                  % rescaling to unit
power
    bitsRX = [bitsRX; qamdemod(symbolsRX , numSymbols_Constellation ,  'gray'
, 'OutputType' , 'bit' , 'UnitAveragePower' , true)];   % demodulation
    vecSymbolsRX = [vecSymbolsRX; symbolsRX];
end
end
```

A.4 Channel generation

```
function [channelStruct] = genChannelv2( channelType , deltaF , T , N , M ,
UE_speed , fc , CP_length , numFrames , channelPropagationMode ,
CP_reducedOrNot , CP_zeroOrNot )

% deltaF : sub-carrier spacing
% T : time of one symbol typicall equal to 1/deltaF
% M : Num of sub-carriers
% N : Num of TD symbols
% CP_reducedOrNot : 1 == one CP for N TD symbols i.e RCP || 0 == one CP for
each of N TD symbosl
% CP_zeroOrNot : 1 == CP is zeros || 0 == CP is circular
if CP_reducedOrNot == 1         % RCP
    lenOneFrame = N*M + CP_length;
    lenTotal = numFrames*( lenOneFrame );
elseif CP_reducedOrNot == 0     % CP
    lenOneFrame = N * ( M + CP_length );
    lenTotal = numFrames * lenOneFrame;
end
if channelType == 0
    % AWGN
    channelStruct.timeVariant = 0;
    channelStruct.numTaps = 1;
```

```
    channelStruct.TapMag = 1;
    channelStruct.phase = 0;
    channelStruct.channelTaps = channelStruct.TapMag * exp( 1i *
channelStruct.phase  );
elseif channelType == 1
    % Rayleigh
    channelStruct.timeVariant = 0;
    channelStruct.numTaps = 5;
    channelStruct.channelTaps = sqrt(1/2) * (randn(1 , channelStruct.numTaps)
+ 1i*randn(1 , channelStruct.numTaps));
    channelStruct.TapMag = abs(channelStruct.channelTaps);
    channelStruct.phase = phase(channelStruct.channelTaps);
elseif channelType == 2
    % Rician
    channelStruct.timeVariant = 0;
    rician_Kfactor = 2;                % ratio of power in line-of-sight vs
NLOS
    channelStruct.numTaps = 5;
    LoSPwr = rician_Kfactor / (rician_Kfactor + 1);
    NLoSPwr = 1 / (rician_Kfactor + 1);
    channelStruct.channelTaps = sqrt(NLoSPwr/2) * (randn(1 ,
channelStruct.numTaps) + 1i*randn(1 , channelStruct.numTaps));     % NLOS
component
    channelStruct.channelTaps(1) = channelStruct.channelTaps(1) + sqrt(LoSPwr)
* exp(1i * 2 * pi * rand);                        % LoS component
    channelStruct.TapMag = abs(channelStruct.channelTaps);
    channelStruct.phase = phase(channelStruct.channelTaps);
elseif channelType == 3
    % Vehicular
    channelStruct.delays = [0 30 150 310 370 710 1090 1730 2510] * 10^(-9);
    channelStruct.pdp = [0 -1.5 -1.4 -3.6 -0.6 -9.1 -7.0 -12.0 -16.9];
    numTaps = numel( channelStruct.delays );
    channelStruct.Doppler_taps = cos(2*pi * rand(1 , numTaps));
    channelStruct.channelTaps = (sqrt(1/2) * (randn(1 , numTaps) + 1i*randn(1
, numTaps)));
elseif channelType == 4
    % Pedestrain
    channelStruct.delays = [0 30 70 90 110 190 410] * 10^(-9);
    channelStruct.pdp = [0 -1.0 -2.0 -3.0 -8.0 -17.2 -20.8];
    numTaps = numel( channelStruct.delays );
    channelStruct.Doppler_taps = cos(2*pi * rand(1 , numTaps));
    channelStruct.channelTaps = (sqrt(1/2) * (randn(1 , numTaps) + 1i*randn(1
, numTaps)));

elseif channelType == 5
    % Train
    channelStruct.delays = [0 50 120 200 230 500 1600 2300 5000] * 10^(-9);
    channelStruct.pdp = [-1 -1 -1 0 0 0  -3 -5 -7];
    numTaps = numel( channelStruct.delays );
    channelStruct.Doppler_taps = cos(2*pi * rand(1 , numTaps));
    channelStruct.channelTaps = (sqrt(1/2) * (randn(1 , numTaps) + 1i*randn(1
, numTaps)));
else
    error('Channel type not defined!');
end
end
```

A.5 Discrete time baseband channel generation

```
function [CIR , numTaps_DTB_channel] = getDTBChannelv2( PDP_dB , delays ,
deltaF , T , N , M , UE_speed , fc , CP_length , numFrames , CP_reducedOrNot ,
channelTaps , Doppler_taps )
% In v2 actual CP is not added -- its just incorporated in the channel
% convolution matrix simulaitng what would have been received had the CP
% been there.
% deltaF : sub-carrier spacing
% T : time of one symbol typicall equal to 1/deltaF
% M : Num of sub-carriers
% N : Num of TD symbols
% numFrames : number of OFDM symbols to generate the channel for
% CP_reducedOrNot : 0 == one CP for N TD symbols i.e RCP || 1 == one CP for
each of N TD symbosl
% This function geenrates the discrete time baseband channel from continuous
time channel parameters
delayResolution = 1 / (M * deltaF);
dopplerResolution = 1 / (N * T);
% delays
numTaps = numel(delays);
normalizedDelayTaps = round(delays / delayResolution);   % not accounting
fractional delay
% channel coefficients
PDP = 10.^(PDP_dB / 10);
PDP = PDP / sum(PDP);                              % normalization
channelTaps = sqrt(PDP) .* (channelTaps);
% doppler
UE_speed = UE_speed * 1000 / 3600;
dopplerFreq = (UE_speed * fc) / 299792458;
normalizedDopplerFreq = dopplerFreq / dopplerResolution;
Doppler_taps = (normalizedDopplerFreq * Doppler_taps );       % Jake's
spectrum
% DTB Channel
z = exp(1i * 2*pi / N / M);
maxDelayTap = max( normalizedDelayTaps );
if CP_reducedOrNot == 1         % RCP
    lenTotal = numFrames*( N*M + CP_length );
elseif CP_reducedOrNot == 0     % CP
    lenTotal = numFrames * N * ( M + CP_length );
end
CIR = zeros(maxDelayTap+1 , lenTotal );          % time variant ( along column
) delay (along row) channel impulse response
for indTime = 0 : lenTotal-1
    for i = 1 : numTaps
        CIR(normalizedDelayTaps(i)+1 , indTime+1) =
CIR(normalizedDelayTaps(i)+1 , indTime+1) + channelTaps(i) * z^(
Doppler_taps(i) * (indTime-normalizedDelayTaps(i)) );
    end
end
numTaps_DTB_channel = maxDelayTap+1;
end
```

A.6 Get DTB channel wrapper function

```
function [channelStruct] = Wrapper_getDTBChannelv2(channelStruct , deltaF , T
, N , M , UE_speed , fc , CP_length , numFrames , CP_reducedOrNot ,
channelType )
if channelType >= 3
    [CIR , numTaps_DTB_channel] = getDTBChannelv2(channelStruct.pdp ,
channelStruct.delays , deltaF , T , N , M , UE_speed , fc , CP_length ,
numFrames , CP_reducedOrNot , channelStruct.channelTaps ,
channelStruct.Doppler_taps );
    channelStruct.timeVariant = 1;
    channelStruct.numTaps = numTaps_DTB_channel;
    channelStruct.channelTaps = CIR;
end
end
```

A.7 Constellation diagram generation

```
clc;
clear all;
% close all;
rng(100);
% rng('shuffle');
%% Generic independent parameters
% sim
numFrames = 50;                % no. of monte carlo iterations      % ##
numOTFS_Sym_perFrame = 1;      % no of OTFS symbols per frame -- OFDM
determined based on this
IndPLot = 1;
fontSize = 14;
lineWidth = 3;
% Pwr
PwrTX_dB = [25];        % just one value for constellation diagram plotting
% modulation parameters
modType = 'QPSK';       % BPSK || QPSK || 16QAM % ##
% spectral parameters
fc = 2e9;
deltaF = 15e3;
num_subCarriers = 64;
% noise parameters
noiseType = 0;      % 0 == directly define noise variance || 1 == define
based on BW
NoiseVar_dB = 00;
noiseFloor_dB_Hz = -204;
noiseFigure_dB = 5;
%% Generic derived parameters
% Pwr
if numel(PwrTX_dB) ~= 1
    error('Please just give one value for constellation diagram plotting!');
end
PwrTX = 10.^(PwrTX_dB ./ 10);
% modulation
vecConstellation = {'BPSK' , 'QPSK' , '16QAM'};
vecM = [2 , 4 , 16]; % number of points in each constellation type
```

```
vecBits = log2(vecM);% Number of bits each symbol in each constellation
encodes
modInd = find( strcmp( vecConstellation , modType ) == 1 );
numSymbols_Constellation = vecM( modInd );
numBitsPerSymbol = vecBits( modInd );
% spectral parameters
T = 1 / deltaF;
BW = num_subCarriers * deltaF;
% noise parameters
if noiseType == 0
elseif noiseType == 1
    NoiseVar_dB = noiseFloor_dB_Hz + noiseFigure_dB + 10*log10( BW );  %
variance in dB Watts
else
    error('Define correct noise type!');
end
NoiseVar = 1 * 10.^( NoiseVar_dB / 10 ); % Watts
%% OTFS parameters
num_timeSym_OTFS = 8;       % ##
CP_length_OTFS = 16;
CP_reducedOrNot_OTFS = 0;
CP_zeroOrNot_OTFS = 0;
%% OFDM parameters
CP_length_OFDM = 0.25 * num_subCarriers;
CP_reducedOrNot_OFDM = 0;
CP_zeroOrNot_OFDM = 0;
%% derived OTFS and OFDM param
% Bits
numBitsPerFrame = numBitsPerSymbol * num_subCarriers * num_timeSym_OTFS *
numOTFS_Sym_perFrame;
numSymsPerFrame = numBitsPerFrame / numBitsPerSymbol;
numOFDM_Sym_perFrame = num_timeSym_OTFS * numOTFS_Sym_perFrame;
numBitsPerOFDMSym = numBitsPerSymbol * num_subCarriers;
numBitsPerOTFSSym = numBitsPerSymbol * num_subCarriers * num_timeSym_OTFS;
%% Channel parameters
channelType = 5;            % 0 == AWGN || 1 == Rayleigh || 2 == Rician || 3
== Vehicular || 4 == Pedestrian || 5 == Train  % ##
unitPwr_channelGen = 1;
UE_speed = 400;               % km per hour % ##
channelPropagationMode = 1; % 0 == convolution || 1 == channel matrix (time
variant channels)
%% Varaibles for saving
err_OTFS_perFrame = zeros( numel( PwrTX_dB ) , numFrames );
err_OFDM_perFrame = zeros( numel( PwrTX_dB ) , numFrames );
symbolList_OTFS_perFrame = zeros( numSymsPerFrame , numFrames );
symbolList_OFDM_perFrame = zeros( numSymsPerFrame , numFrames );
%% Main Loop
for indFrame = 1:numFrames
    % get channel realization
    lenOFDM = (num_subCarriers + CP_length_OFDM) * numOFDM_Sym_perFrame;
    if CP_reducedOrNot_OTFS         % reduced
        lenOTFS = (num_subCarriers * num_timeSym_OTFS) + CP_length_OTFS *
numOTFS_Sym_perFrame;
    elseif ~CP_reducedOrNot_OTFS    % regular
        lenOTFS = (num_subCarriers + CP_length_OTFS) * num_timeSym_OTFS *
numOTFS_Sym_perFrame;
```

```
    end
    if lenOFDM >= lenOTFS
        [channelStruct] = genChannelv2( channelType , deltaF , T , 1 ,
num_subCarriers , UE_speed , fc , CP_length_OFDM , numOFDM_Sym_perFrame ,
channelPropagationMode , CP_reducedOrNot_OFDM , CP_zeroOrNot_OFDM );
    else
        [channelStruct] = genChannelv2( channelType , deltaF , T ,
num_timeSym_OTFS , num_subCarriers , UE_speed , fc , CP_length_OTFS ,
numOTFS_Sym_perFrame , channelPropagationMode , CP_reducedOrNot_OTFS ,
CP_zeroOrNot_OTFS );
    end
    % Generate bits, channel and noise realization bits and mapping
    bitsTX = randi([0,1] , 1 , numBitsPerFrame)';
% bit generation
    symbolsTX_unitEnergy = qammod(bitsTX , numSymbols_Constellation , 'gray' ,
'InputType' , 'bit' , 'UnitAveragePower' , true);  % symbol mapping
    % noise
    noise = sqrt( NoiseVar/2 ) * ( randn( round(numSymsPerFrame * (1 +
max(CP_length_OFDM/num_subCarriers , CP_length_OTFS) )) , 1  ) + 1i*randn(
round(numSymsPerFrame * (1 + max(CP_length_OFDM/num_subCarriers ,
CP_length_OTFS) )) , 1 ) ); % noise generation
    for indPwr = 1:numel( PwrTX_dB )

        % Generate bits, channel and noise realization bits and mapping
        % bitsTX = randi([0,1] , 1 , numBitsPerFrame)';
% bit generation
        % symbolsTX = sqrt( PwrTX(indPwr) ) * qammod(bitsTX ,
numSymbols_Constellation , 'gray' , 'InputType' , 'bit' , 'UnitAveragePower' ,
true);  % symbol mapping
        symbolsTX = sqrt( PwrTX(indPwr) ) * symbolsTX_unitEnergy;
        % % noise
        % noise = sqrt( NoiseVar/2 ) * ( randn( round(numSymsPerFrame * (1 +
max(CP_length_OFDM/num_subCarriers , CP_length_OTFS) )) , 1  ) + 1i*randn(
round(numSymsPerFrame * (1 + max(CP_length_OFDM/num_subCarriers ,
CP_length_OTFS) )) , 1 ) ); % noise generation

        % OFDM
        [bitsRX_OFDM , symbolsRX_OFDM] = simOFDMv2(numOFDM_Sym_perFrame ,
num_subCarriers , symbolsTX , noise , channelType , CP_length_OFDM ,
numSymbols_Constellation , fc , deltaF , T , UE_speed ,
channelPropagationMode, CP_reducedOrNot_OFDM , CP_zeroOrNot_OFDM , 0*NoiseVar
, channelStruct );
        err_OFDM_perFrame( indPwr , indFrame ) = sum( bitsRX_OFDM ~= bitsTX )
/ numBitsPerFrame;
        symbolList_OFDM_perFrame(: , indFrame) = symbolsRX_OFDM;

        % OTFS with freq domain single tap equalization
        % [bitsRX_OTFS , symbolsRX_OTFS] = simOTFSv2(numOTFS_Sym_perFrame ,
num_timeSym_OTFS , num_subCarriers , symbolsTX , noise , channelType ,
CP_length_OTFS , numSymbols_Constellation , fc , deltaF , T , UE_speed ,
channelPropagationMode , CP_reducedOrNot_OTFS , CP_zeroOrNot_OTFS , 0*NoiseVar
, channelStruct );
        % err_OTFS_perFrame( indPwr , indFrame ) = sum( bitsRX_OTFS ~= bitsTX
) / numBitsPerFrame;

        % OTFS with time domain channel reciprocity based equalization
```

```
        [bitsRX_OTFS , symbolsRX_OTFS] = simOTFSv3(numOTFS_Sym_perFrame ,
num_timeSym_OTFS , num_subCarriers , symbolsTX , noise , channelType ,
CP_length_OTFS , numSymbols_Constellation , fc , deltaF , T , UE_speed ,
channelPropagationMode , CP_reducedOrNot_OTFS , CP_zeroOrNot_OTFS , 0*NoiseVar
, channelStruct );
        err_OTFS_perFrame( indPwr , indFrame ) = sum( bitsRX_OTFS ~= bitsTX )
/ numBitsPerFrame;
        symbolList_OTFS_perFrame(: , indFrame) = symbolsRX_OTFS;

        1;
    end
    1;
end
%% Final calculations
% OFDM
err_OFDM = mean( err_OFDM_perFrame , 2);
% OTFS
err_OTFS = mean( err_OTFS_perFrame , 2);
%% Final plots
figure;
plot( real( symbolList_OFDM_perFrame(:) ) , imag( symbolList_OFDM_perFrame(:)
) , '.b' );
legend_vecType = {'OFDM' };
title( modType );
xlabel( 'Real part' );
ylabel( 'Imaginary part' );
set( findall( gcf , '-property' , 'FontSize' ) , 'FontSize' , fontSize );
set( gcf , 'color' , [1 1 1] );
legend( legend_vecType , 'FontSize' , fontSize , 'Location' , 'northeast' ,
'Box' , 'off' );
axis([-2 2 -2 2]);
grid on;
figure;
plot( real( symbolList_OTFS_perFrame(:) ) , imag( symbolList_OTFS_perFrame(:)
) , '*r' );
legend_vecType = {'OTFS' };
title( modType );
xlabel( 'Real part' );
ylabel( 'Imaginary part' );
set( findall( gcf , '-property' , 'FontSize' ) , 'FontSize' , fontSize );
set( gcf , 'color' , [1 1 1] );
legend( legend_vecType , 'FontSize' , fontSize , 'Location' , 'northeast' ,
'Box' , 'off' );
axis([-2 2 -2 2]);
grid on;
```

References

1. Hong, Y., Thaj, T., Viterbo, E. (2022). Delay-Doppler Communications: Principles and Applications. Netherlands: Elsevier Science.

2. Hadani, R., & Monk, A. (n.d.). OTFS: A New Generation of Modulation Addressing the Challenges of 5G. Cohere Technologies.

3. Monk, A., Hadani, R., Tsatsanis, M., & Rakib, S. (n.d.). OTFS - Orthogonal Time Frequency Space: A novel modulation technique meeting 5G high mobility and massive MIMO challenges. Cohere Technologies.

4. Hadani, R., Rakib, S., Kons, S., Tsatsanis, M., Monk, A., Ibars, C., Delfeld, J., Hebron, Y., Goldsmith, A. J., Molisch, A. F., & Calderbank, R. (2018, August 1). Orthogonal Time Frequency Space Modulation. arXiv:1808.00519v1 [cs.IT].

5. Saif Khan Mohammed, Ronny Hadani, Ananthanarayanan Chockalingam, & Robert Calderbank. (2023). OTFS − A Mathematical Foundation for Communication and Radar Sensing in the Delay-Doppler Domain. arXiv:2302.08696v1 [eess.SP].

6. Wei, Z., Yuan, W., Li, S., Yuan, J., Bharatula, G., Hadani, R., & Hanzo, L. (2021, February 3). Orthogonal Time-Frequency Space Modulation: A Promising Next-Generation Waveform. arXiv:2010.03344v2 [cs.IT].

7. Bello, P. A. (1963). Characterization of Randomly Time-Variant Linear Channels. IEEE Transactions on Communications Systems, December 1963, 360-393.

8. Lampel, F., Joudeh, H., Alvarado, A., & Willems, F. M. J. (2022). Orthogonal Time Frequency Space Modulation Based on the Discrete Zak Transform. Entropy, 24(24), 1704. https://doi.org/10.3390/e24121704

9. Li, M., Liu, W., & Lei, J. (2023). A review on orthogonal time–frequency space modulation: State-of-art hotspots and challenges. Computer Networks, 224, 109597. https://doi.org/10.1016/j.comnet.2023.109597

10. Oppenheim, A. V., Willsky, A. S., & Nawab, S. H. (1996). Signals and Systems (2nd ed.). Prentice Hall.

11. Rappaport, T. S. (2024). *Wireless Communications: Principles and Practice*. United Kingdom: Cambridge University Press.

12. Ramachandran, M. K., Surabhi, G. D., & Chockalingam, A. (2020). OTFS: A New Modulation Scheme for High-Mobility Use Cases. *Journal of the Indian Institute of Science, 100(2), 315-336.*

13. Chu, T. M. C., & Zepernick, H.-J. (2023). Modulation and Detection for High Doppler Channels: An Overview on OTFS Modulation. *Technical Report*, Blekinge Institute of Technology, Department of Computer Science.

14. Wang, T. J., Proakis, J. G., Masry, E., & Zeidler, J. R. (2006). Performance degradation of OFDM systems due to Doppler spreading. *IEEE Transactions on Wireless Communications, 5(6).*

15. Li, S. (2022). Orthogonal Time Frequency Space (OTFS) Modulation for Wireless Communications (Doctoral dissertation, The University of New South Wales). Retrieved from http://hdl.handle.net/1959.4/100590

16. Molisch, A. F. (2011). *Wireless communications* (2nd ed.). John Wiley & Sons Ltd.

17. Hlawatsch, F., & Matz, G. (2011). Wireless communications over rapidly time-varying channels. Academic Press.

18. A. J. Janssen, "The Zak transform: A signal transform for sampled time-continuous signals." Philips J. Res., vol. 43, no. 1, pp. 23–69, 1988.

19. Proakis, J., Salehi, M. (2008). Digital communications. Boston: McGraw-Hill Education.

20. Li, S., Yuan, W., Wei, Z., He, R., Ai, B., Bai, B., & Yuan, J. (2021, July). A tutorial to orthogonal time frequency space modulation for future wireless communications. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. https://doi.org/10.1109/ICCCWorkshops52231.2021.9538891

21. Grami, A. (2015). Introduction to Digital communications. Netherlands: Elsevier Science.