ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Master's Degree in
Electronics and Telecommunications Engineering for Energy

DELAY-DOPPLER MODULATION AND SOFTWARE DEFINED RADIO IMPLEMENTATION

Master's Thesis in
Information Theory and Cryptography

Supervisor:
Prof. MARCO CHIANI
Co-Supervisors:
Dr. FEDERICO CLAZZER
DAVIDE BERGAMASCO

Submitted by:
MARTINA MARGARA

ACADEMIC YEAR 2024–2025

KEY WORDS

OTFS

SDR

GNU Radio

MATLAB

Python

Contents

Abstract **1**

Introduction **3**

1 The Wireless Channel **7**
1.1 Single-path channel with AWGN 8
1.1.1 Uncoded transmission in AWGN channels 9
1.1.2 M-QAM mapping 10
1.2 Multipath channels 12
1.2.1 Static multipath channels 13
1.2.2 Time-variant multipath channels 14
1.2.3 Channel representations 15

2 Multicarrier Modulations **19**
2.1 Preliminaries 19
2.1.1 The DFT and Its Properties 19
2.1.2 The Cyclic Prefix (CP) 21
2.2 Orthogonal Frequency-Division Multiplexing (OFDM) 22
2.2.1 OFDM frequency domain input–output relation 23
2.2.2 Frequency equalization 25
2.2.3 Advantages and disadvantages of OFDM 25
2.3 Orthogonal Time Frequency Space Modulation (OTFS) 25
2.3.1 Pre-coded OFDM 26
2.3.2 IDZT-based OTFS transceiver 27
2.3.3 Discrete-time input-output relations 30
2.3.4 Delay-Doppler input-output relations 32
2.3.5 Zero Padded OTFS 34

3 Delay-Doppler Channel Estimation with Embedded Pilot **39**
3.1 The integer Doppler case 40

3.2 The fractional Doppler case 43
3.2.1 Fractional Doppler estimation with cross-correlation 44
3.2.2 Channel gain estimation 46

4 Numerical Results **47**
4.1 OFDM over different channels 47
4.2 OTFS channel estimation in delay-Doppler domain 55

5 Software Defined Radio (SDR) **65**
5.1 Flowgraph implementation 65
5.2 Loop-back configuration 68
5.2.1 SNR computation 72

6 Conclusions **77**
6.1 Future developments 78

List of Figures **81**

Abstract

Modern wireless communication networks, including future 6G systems, are expected to support use cases in high-mobility environments. These scenarios often involve Non-Terrestrial Networks (NTNs), such as low Earth orbit (LEO) satellites, unmanned aerial vehicles (UAVs), vehicular communications, and high-speed railway systems. The Orthogonal Frequency-Division Multiplexing (OFDM) scheme is widely employed today thanks to its great properties in static multipath channels. The strength of OFDM lies in the use of the cyclic prefix (CP), which provides a circulant channel matrix that enables a low-complexity single-tap equalization at the receiver. While OFDM is expected to remain a baseline waveform in 6G systems, it faces significant challenges in rapidly time-varying environments. In such scenarios, the strong Doppler effect disrupts subcarrier orthogonality, leading to inter-carrier interference (ICI) and a loss of the circulant structure of the channel matrix. Consequently, OFDM performance degrades, and the great advantage of having a low complexity receiver is lost. To overcome OFDM's limitations in high-mobility scenarios, one approach is to employ it with additional precoding. Building on this idea, Orthogonal Time Frequency Space (OTFS) was introduced in 2017 as a novel modulation scheme. Since then, it has gained growing interest as a promising alternative to OFDM in strongly time-variant channels.

In this thesis, channel estimation algorithms for OTFS-based communication systems are investigated. The approach employs an embedded pilot-aided scheme that inserts a single pilot and guard symbols within the delay-Doppler grid. Numerical results are presented to compare channel estimation performance accounting for both integer and fractional Doppler shifts. The outcomes show that the impact of fractional Doppler can be effectively mitigated with relatively simple additional signal processing, resulting in only minor performance differences. The ultimate objective of this thesis is the integration of the complete OTFS transceiver chain onto a Software Defined Radio (SDR) platform. In particular, a loop-back configuration is implemented and preliminary results confirm the correctness of the system.

Introduction

Radio communications originated around 1820 with Oersted's experiments demonstrating that an electric field could move a compass needle, therefore establishing a connection between electricity and magnetism. Further work by Ampere, Gauss, Henry, Faraday and others advanced knowledge of electromagnetic waves, culminating in Maxwell's theory of electromagnetism, published in 1865. A major milestone occured in 1895, when Guglielmo Marconi successfully transmitted a radio signal across his father's estate in Bologna. This is considered to be the birth of radio communications, a term coined in the early 1900s. Marconi moved to England to continue his experiments with increasingly long transmission ranges, culminating in the first transatlantic radio transmission in 1901. In 1900, Fessenden became the first person to send a speech signal over radio waves, and six years later, he made the first public radio broadcast. From these early beginnings, radio technology advanced rapidly, enabling transmissions over larger distances with better quality and using less power and smaller, cheaper devices. This enabled public and private radio communications, television and wireless networking.

Nowadays wireless communication networks are expected to support higher data rates and lower latency. These demands are particularly critical in high-mobility scenarios, such as NTNs, which are positioned above the Earth's surface and include high-altitude platform stations (HAPS), UAVs, and satellites. In these scenarios, the largely adopted OFDM scheme, which is already known to be the baseline waveform for 6G, faces significant challenges. The strength of the OFDM system lies in its use of the CP, which provides a circulant channel matrix that enables a low-complexity single-tap equalization at the receiver. This makes OFDM effective in typical static multipath channels. However, in rapidly time-varying scenarios, for example when dealing with high speed railways, fast-moving UAVs or LEO satellites, the Doppler effect becomes significant and compromises the orthogonality between subcarriers, leading to ICI, and making the channel matrix no longer circulant. As a result, the performance of OFDM is severely degraded and single-tap equalization becomes ineffective. Hence, the great advantage of having a low complexity receiver is lost. Moreover, in these high mobility scenarios, being the strongly

time-variant channel, it needs to be estimated more frequently. As a consequence, more pilot resources have to be employed and the channel estimation process becomes more complex. To address these problems, one idea is to employ OFDM as the baseline waveform, with additional precoding blocks.

Following this line of thought, a group of researchers proposed a novel communication scheme known as OTFS modulation [1]. The OTFS modulation was first proposed in 2017 and since then has attracted an increasing interest among the research community since it seems to be a promising alternative to the standard OFDM system in high-mobility settings. The mathematical derivation of OTFS can be found in [2]. Important insights into the stationarity and predictability of the delay-Doppler domain are presented in [3], where authors outline guidelines for OTFS to preserve these properties. The Zak transform was initially introduced in [4], later explored for signal processing purposes in [5], and then its discrete version was detailed in [6]. A derivation of OTFS using the discrete Zak transform is presented in [7], while a comprehensive discussion on wireless channel representations and their relations can be found in [8].

In recent years, several studies have addressed the challenge of channel estimation for the OTFS system. Leveraging the delay–Doppler domain representation of time-varying multipath channels, OTFS is specifically designed to combat the dynamics of these scenarios. Most existing OTFS channel estimation techniques are data-aided, employing pilot symbols embedded in or superimposed on the delay–Doppler grid. One of the earliest methods was proposed in [9], where the authors introduce a scheme that embeds a single pilot symbol within the OTFS frame alongside data symbols. Orthogonality between the pilot and data is ensured through the use of guard symbols, which facilitate channel estimation at the receiver. To enhance spectral efficiency, superimposed pilot techniques have been proposed. The work in [10] introduces a two-dimensional spread pilot sequence directly added to data symbols, eliminating guard intervals but resulting in high Peak-to-Average Power Ratio (PAPR). This issue is addressed in [11], where a single superimposed pilot is embedded and joint estimation is performed, although requiring a significantly higher pilot power. A more practical solution is provided in [12], which spreads the pilot energy over a one-dimensional sequence to lower PAPR.

The objective of this thesis is to implement a complete OTFS communication chain on a SDR platform. To achieve this goal, leveraging the sparse representation of wireless channels in the delay-Doppler domain, channel estimation algorithms, inspired by [9] and [13], are investigated and implemented, addressing both integer and fractional Doppler

shifts. Subsequently, the full OTFS chain is integrated in SDR, using GNU Radio as the software environment.

The structure of the dissertation is as follows: the Introduction outlines the background and motivations behind the work. Chapter 1 provides an overview of various wireless channel models. Chapter 2 discusses multicarrier modulation schemes, with particular emphasis on OFDM and OTFS. Chapter 3 delves into channel estimation approaches in the delay-Doppler domain, accounting for the presence of both integer and fractional Doppler shifts. In Chapter 4, using Python as the programming language, we present numerical results obtained via Monte Carlo simulations, that validate the proposed channel estimation techniques. Chapter 5 describes the implementation of an OTFS-based communication pipeline in SDR and reports the experimental results obtained through laboratory testing. Finally, the Conclusion summarizes the key outcomes of the thesis and suggests possible directions for future research.

1 The Wireless Channel

The performance limits of a wireless communication system are determined by the characteristics of the channel in which the system operates. Therefore, realistic channel models are essential for both system design and performance evaluation. In this chapter, we start our analysis by spending some words about thermal noise and then a single-path channel with additive white Gaussian noise (AWGN) is introduced. Later on, we focus on multipath channels, considering both the static and the time-variant models.

Thermal noise

All electrical conductors and devices generate thermal noise due to the Johnson effect. From the point of view of the receiver, it can be equivalently modeled as additive white noise at the receiver input and all the components of the system can be considered noiseless. In a band-limited system with bandwidth B, the equivalence holds if the noise $n(t)$ belongs to an ergodic Gaussian zero mean random process with variance σ^2, $n(t) \sim \mathcal{N}(0, \sigma^2)$, with two-sided power spectral density $G_{\mathrm{N}}(f) = \frac{N_0}{2}\mathrm{Rect}\left(\frac{f}{2B}\right)$, where f denotes the frequency [Hz] and

$$\mathrm{Rect}\left(\frac{f}{2B}\right) = \begin{cases} 1, & |f| \leq B, \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

$N_0 = k_{\mathrm{B}} T_{\mathrm{sys}}$ [W/Hz] is the noise power spectral density, with $k_{\mathrm{B}} = 1.38 \times 10^{-23}$ [J/K] being the Boltzmann's constant and T_{sys} [K] the system noise equivalent temperature. Here, a fundamental mathematical tool that will be used throughout the thesis is introduced: the Fourier transform [14]. Specifically, we define its inverse with respect to the variable f, applied to a general signal $X(f)$. This operation is denoted as $\mathcal{F}_f^{-1}\{X(f)\}$ and is defined as:

$$\mathcal{F}_f^{-1}\{X(f)\} = \int_{-\infty}^{+\infty} X(f)\, e^{j2\pi f t}\, df. \tag{1.2}$$

If the variable f represents frequency, then $\mathcal{F}_f^{-1}\{X(f)\}$ corresponds to the time-domain representation of the signal $X(f)$. It is therefore possible to define the noise auto-

correlation function as follows:

$$R_{\mathrm{N}}(\tau) = \mathcal{F}_f^{-1}[G_{\mathrm{N}}(f)] = N_0 B \,\mathrm{Sinc}(\tau 2B) \tag{1.3}$$

where τ is the time difference between two instances of $n(t)$. The noise variance is given by:

$$\sigma^2 = \mathbb{E}[n(t)^2] = P_{\mathrm{N}} = R_{\mathrm{N}}(0) = N_0 B, \tag{1.4}$$

equal to the noise power.

1.1 Single-path channel with AWGN

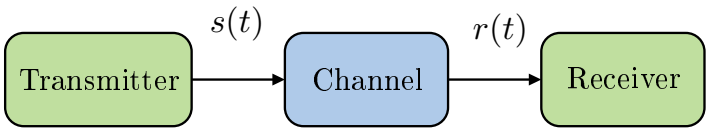


Figure 1.1: Communication scheme over single-path channel.

In reality, the noise introduced by the channel is a zero mean complex circular symmetric Gaussian random variable $\mathrm{w}(t) \sim \mathcal{CN}(0, \sigma_{\mathrm{w}}^2)$, whose expression is $\mathrm{w}(t) = \mathrm{w}_I(t) + j\mathrm{w}_Q(t)$ where $\mathrm{w}_I(t), \mathrm{w}_Q(t) \sim \mathcal{N}(0, \frac{\sigma^2}{2})$ and $\sigma_{\mathrm{w}}^2 = P_{\mathrm{w}} = N_0 B$. Considering Fig. 1.1, where a transmitter sends information to a receiver over a wireless channel, the effect of the single-path channel is just a complex constant h_0. Thus, its performance is limited exclusively by the thermal noise. The received passband signal $r(t)$ can be expressed by the formula:

$$r(t) = h_0\, s(t) + \mathrm{w}(t). \tag{1.5}$$

Note that, if $h_0 = 1$, i.e. there is no attenuation introduced by the channel, the single-path channel reduces to a canonical AWGN channel. An important quantity that will be used in the following chapters is the signal-to-noise ratio (SNR). This is defined as:

$$\mathrm{SNR} = \frac{P_r}{P_{\mathrm{w}}} = \frac{P_r}{N_0 B}$$

where P_{w} corresponds to the noise power and P_r represents the received signal power which comprehends attenuations introduced by the channel due to physical phenomena such as pathloss, shadowing and fading.

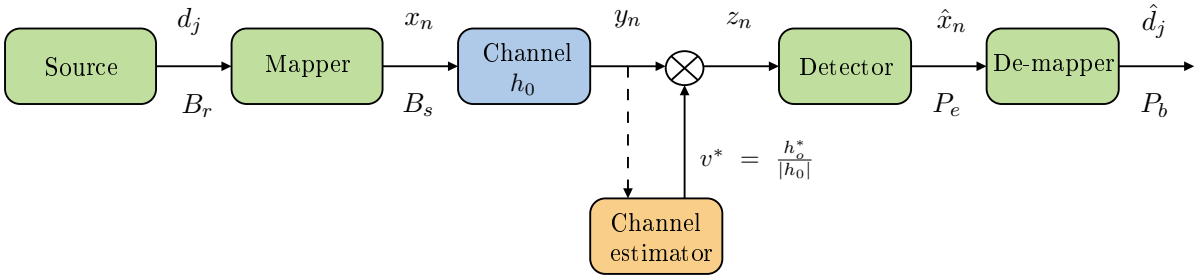


Figure 1.2: Equivalent low-pass discrete-time channel.

1.1.1 Uncoded transmission in AWGN channels

In this section, we assume to transmit independent and identically distributed (i.i.d.) symbols $x_n \in \mathbb{C}$ uniformly drawn from a complex constellation $\mathcal{C}_X$ of cardinality $M = |\mathcal{C}_X|$. The information bits d_j are therefore mapped through a suitable mapper and sent over a channel with bandwidth $B = B_s$, where B_s is the symbol rate and $T = 1/B_s$ is the symbol time. Considering now the equivalent low-pass discrete-time AWGN channel shown in Fig. 1.2, the received signal in (1.5) becomes:

$$y_n = h_0 x_n + w_n. \tag{1.6}$$

At the receiver side, the goal is to recover the transmitted bits. In order to do so, we start by assuming that h_0 is known (estimated) and then we drop the time index n so that we can write the normalized sufficient statistics as:

$$z = v^* y = \frac{h_0^*}{|h_0|} h_0 x + \frac{h_0^*}{|h_0|} \mathrm{w} = |h_0| x + \tilde{\mathrm{w}} \tag{1.7}$$

where $\tilde{\mathrm{w}} \sim \mathrm{w}$, being $|v| = 1$. Then, the symbol-by-symbol detector at the generic time instant computes

$$\hat{x} = \arg \min_{x \in \mathcal{C}_X} |z - |h_0| x|. \tag{1.8}$$

Specifically, at every symbol received, a decision is taken looking at the euclidean distances between points of the constellation $|h_0|x$ and the received symbol z. In order to evaluate the performance of the system, the error probability P_e is derived. In general, this is expressed as:

$$P_e = \frac{1}{M} \sum_{m=1}^{M} P_e \Big|_{x=x^{(m)}}. \tag{1.9}$$

where $x^{(m)}$ is the generic symbol belonging to the constellation $\mathcal{C}_X$, and $P_e \big|_{x=x^{(m)}}$ is the conditional error probability given that $x^{(m)}$ is transmitted. However, closed-form expressions for the computation of P_e, when a generic constellation $\mathcal{C}_X$ is adopted, are

not available. Therefore, we consider a Union Bound, assuming that the constellation is symmetric and keeping only the dominant terms corresponding to the couples of points having the minimum distance. According to this approximation, we obtain the error probability formulation for generic constellations:

$$P_e \approx \frac{k}{2}\text{erfc}\left(\sqrt{\frac{d^2}{8N_0B}}\right) \tag{1.10}$$

where $d = \min_i d_{1i}$ is the minimum distance between the couples of points on $\mathcal{C}_X$ and k is the number of points that are located at the minimum distance from the reference point 1.

1.1.2 M-QAM mapping

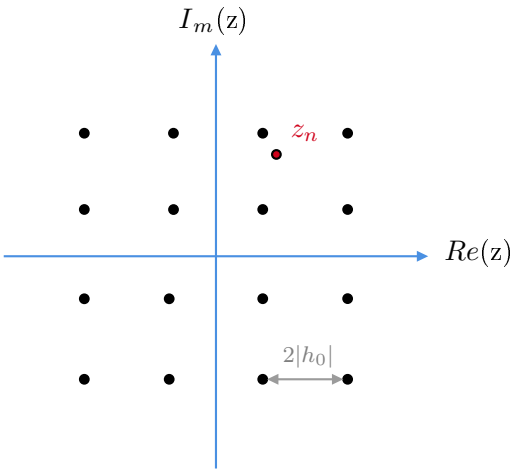


Figure 1.3: Constellation points and received symbol.

Considering the particular case of M-quadrature amplitude modulation (QAM), the transmitted symbols are expressed as $x = x^{(I)} + jx^{(Q)}$. The total number of points composing the constellation is M, and $L = \log_2 M$ is the number of bits encoded per symbol. As a consequence, we can define the relationship between the symbol rate B_s and the bit rate B_r as follows:

$$B_s = \frac{B_r}{2\log_2 L}. \tag{1.11}$$

Similarly for the bit time T_b and the symbol time T, we have $T = T_b \log_2 L$. Moreover, bits are associated to symbols using the Gray code. Thus, the bit error probability can be approximated as:

$$P_b \approx \frac{P_e}{2\log_2 L}. \tag{1.12}$$

To find the performance of M-QAM, we use the approximation of (1.10) and referring to Fig. 1.3, we can substitute $d = 2|h_0|$ and $k = 4$, leading to:

$$P_e \approx \frac{k}{2}\,\mathrm{erfc}\left(\sqrt{\frac{d^2}{8N_0B}}\right) = 2\,\mathrm{erfc}\left(\sqrt{\frac{|h_0|^2}{2N_0B}}\right). \tag{1.13}$$

It is convenient to express the result as a function of the conventional SNR $\frac{E_b}{N_0} = \frac{P_r T_b}{N_0}$ where E_b represents the energy per bit. The received power at radio frequency is defined as:

$$P_r = \frac{\mathbb{E}[|h_0\,x_n|^2]}{2} = |h_0|^2\frac{L^2-1}{3} \tag{1.14}$$

and the bit time is:

$$T_b = \frac{T}{2\log_2 L}. \tag{1.15}$$

Substituting the terms and using (1.12), we get an approximated formula for the bit error probability employing the M-QAM mapper:

$$P_b \simeq \frac{1}{\log_2 L}\frac{L-1}{L}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}\frac{3\log_2 L}{L^2-1}}\right). \tag{1.16}$$

4-QAM or QPSK

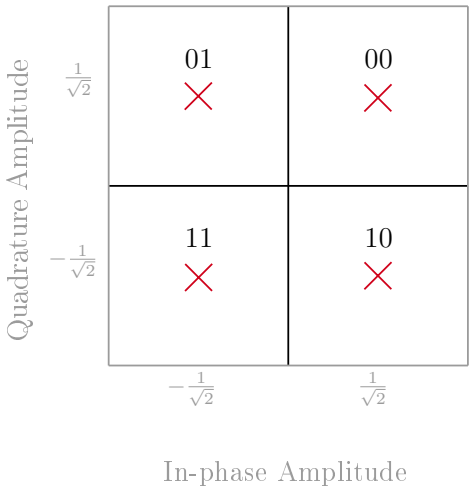


Figure 1.4: Constellation diagram of 4-QAM.

Throughout this thesis, we are going to consider the particular case of 4-QAM (equivalent to Quadrature Phase Shift Keying (QPSK)), whose constellation is presented in Fig. 1.4. Here, the parameters become $M = 4$ and $L = \log_2 M = 2$. As a consequence, the

expression of the bit error probability simplifies to:

$$P_b = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{1.17}$$

1.2 Multipath channels

We now investigate an alternative model, that is able to better capture the wireless propagation phenomenon. In wireless communication systems, multipath propagation plays a central role in shaping the characteristics of the received signal. Rather than receiving a single copy of the transmitted waveform, the receiver typically collects multiple versions of the same signal, each affected by the environment through different propagation delays, attenuations, and potential Doppler shifts. These replicas arise due to the various paths that the electromagnetic wave can take, from direct line-of-sight (LoS) to reflections off buildings, ground, or other obstacles. To select the appropriate modeling approach for system design, it is important to first introduce some key definitions:

1. Coherence Time (T_{co}): this is the time interval over which the channel's response can be considered correlated. In other words, if two observations of the channel are taken within a period shorter than T_{co}, the channel can be approximated as time-invariant.

2. Coherence Bandwidth (B_{co}): this represents the frequency range over which the channel's response remains correlated. If a signal has a bandwidth B smaller than B_{co}, the channel can be considered non-dispersive, meaning it does not cause frequency-selective distortion.

3. Delay spread of the channel $\Delta\tau$: it is defined as the difference between the propagation delays of the longest and shortest paths, i.e., $\Delta\tau = \tau_{max} - \tau_{min}$, and represents the time dispersion of a signal. If the delay spread of the channel is comparable or exceeds the symbol duration of the transmitted signal, intersymbol interference (ISI) occurs. Equivalently, this effect can be seen with reference to the bandwidth B of the transmitted signal: if this exceeds the coherence bandwidth B_{co} of the channel, ISI phenomenon is present, the channel is called frequency selective and causes some parts of the received signal spectrum to be attenuated or phase-shifted more than others. B_{co} and $\Delta\tau$ are related according to the formula:

$$B_{co} \approx \frac{1}{\Delta\tau}$$

4. Doppler spread of the channel $\Delta\nu$: this is a measure of the spectral broadening of a signal due to the Doppler effect, which is caused by the relative motion between the transmitter, receiver, and surrounding objects. It is defined as $\Delta\nu = |\nu_{max} - \nu_{min}|$ where ν_{max} and ν_{min} are the maximum and minimum Doppler shifts observed across all paths. T_{co} and $\Delta\nu$ are inversely related according to the approximation:

$$T_{co} \approx \frac{1}{\Delta\nu}.$$

These parameters are crucial for determining whether a system operates in a time-invariant or time-varying regime, and whether it experiences flat or frequency-selective fading.

1.2.1 Static multipath channels

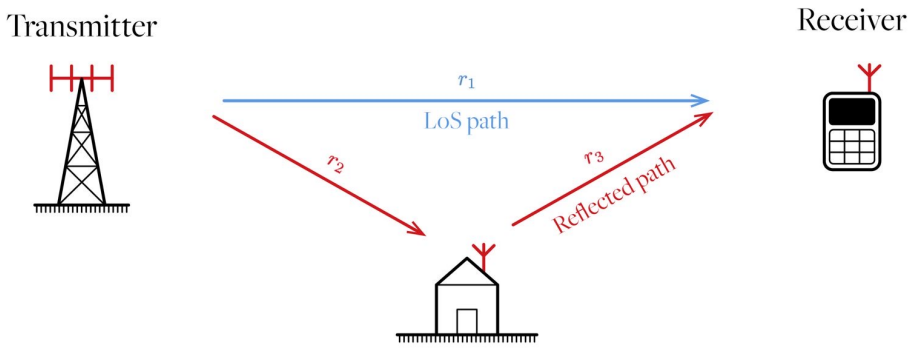


Figure 1.5: Static multipath channel.

We consider first the simplified scenario depicted in Fig. 1.5, where a base station transmits a signal $s(t)$ to a mobile receiver. A nearby building acts as a reflector. Since all elements in this example (transmitter, receiver, and reflector) are static, the received signal is not affected by Doppler shifts. However, two distinct propagation paths exist: a direct LoS path of length r_1, and a reflected path of length r_2+r_3. The corresponding delays, $\tau_1 = \frac{r_1}{c}$ and $\tau_2 = \frac{r_2+r_3}{c}$, depend on the travel distances and the speed of light $c = 3 \cdot 10^8$ m/s. Each path also introduces a unique complex gain, denoted h_1 and h_2, representing the amplitude attenuation and phase shift due to propagation effects. By the principle of superposition, the total received signal $r(t)$ is expressed as the sum of all delayed and attenuated versions of the transmitted signal. In this simple example, it can be written as:

$$r(t) = h_1\, s(t - \tau_1) + h_2\, s(t - \tau_2). \tag{1.18}$$

In general, we can model a multipath channel as being composed of P distinct propagation paths. Therefore, we can generalize the expression in equation (1.18) as:

$$r(t) = \sum_{i=1}^{P} h_i s(t - \tau_i) \tag{1.19}$$

By isolating the effect of the channel, the received signal can be rewritten as:

$$r(t) = \underbrace{\sum_{i=1}^{P} h_i \delta(\tau - \tau_i)}_{g(\tau)} * s(t) \tag{1.20}$$

where $g(\tau)$ is the impulse response of the channel and $*$ denotes the linear convolution. In this model, $g(\tau)$ does not vary with time and is thus referred to as a static multipath channel.

1.2.2 Time-variant multipath channels

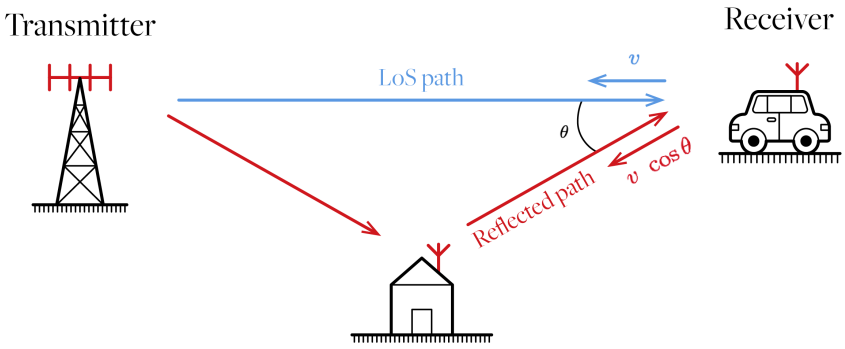


Figure 1.6: Time-variant multipath channel.

We now extend this model by incorporating the effect of mobility. This leads to a more general and closer representation of high-mobility wireless channels. As shown in Figure 1.6, the receiver is no longer a static entity; instead, its position changes over time. Consequently, the spatial distribution of the channel scatterers, as observed from the receiver, becomes time-dependent. This variation may be caused not only by the motion of the receiver, as assumed in our example, but also by changes in the environment, such as the displacement or velocity of individual scatterers.

Before proceeding, it is important to account for a key physical phenomenon that arises in dynamic scenarios: the Doppler effect. When there is a non-zero relative velocity between the emitter and observer of the waveform, the observed signal is seen with an additional

frequency shift, given by:

$$\Delta f = \frac{v}{c} f_0 \tag{1.21}$$

where v is the relative speed between the transmitter and receiver, and f_0 is the central frequency of the transmitted signal. In multipath environments, different propagation paths may be affected by different Doppler shifts due to varying velocities and directions of the reflectors. Channels that exhibit both time and frequency dispersion are referred to as doubly dispersive channels, or sometimes doubly selective channels, as they cause the transmitted signal energy to spread in both the time and frequency domains. Considering these effects, the received signal can be expressed as a sum of multiple time-varying, delayed, frequency and amplitude modulated replicas of the transmitted signal $s(t)$, as follows:

$$r(t) = \sum_{i=1}^{P} h_i \, e^{j2\pi\nu_i(t-\tau_i)} \, s(t-\tau_i). \tag{1.22}$$

This equation represents the baseband equivalent of the transmitted and received signals. The exponential term $e^{j2\pi\nu_i(t-\tau_i)}$ models the Doppler-induced frequency shift for the i-th path, while h_i and τ_i represent the time-varying complex gain and delay, respectively.

1.2.3 Channel representations

In this section, we visualize and investigate the relationship between the frequency-time and delay-Doppler representations of the channel. Considering the doubly-dispersive channel described in section 1.2.2, the received signal $r(t)$ in (1.22) can be written as:

$$r(t) = \int_0^{\infty} g(\tau, t) s(t-\tau) d\tau \tag{1.23}$$

where $0 \leq \tau \leq \infty$ is the propagation delay and $g(\tau, t)$ represents the delay-time impulse response of the channel and it is expressed as:

$$g(\tau, t) = \sum_{i=1}^{P} h_i \, e^{j2\pi\nu_i(t-\tau_i)} \, \delta_i(\tau-\tau_i). \tag{1.24}$$

By taking the Fourier transform along the delay dimension of $g(\tau, t)$, we can derive the frequency-time impulse response of the channel at a fixed time t:

$$H(f, t) = \int_{\tau} g(\tau, t) e^{-j2\pi\tau} d\tau. \tag{1.25}$$

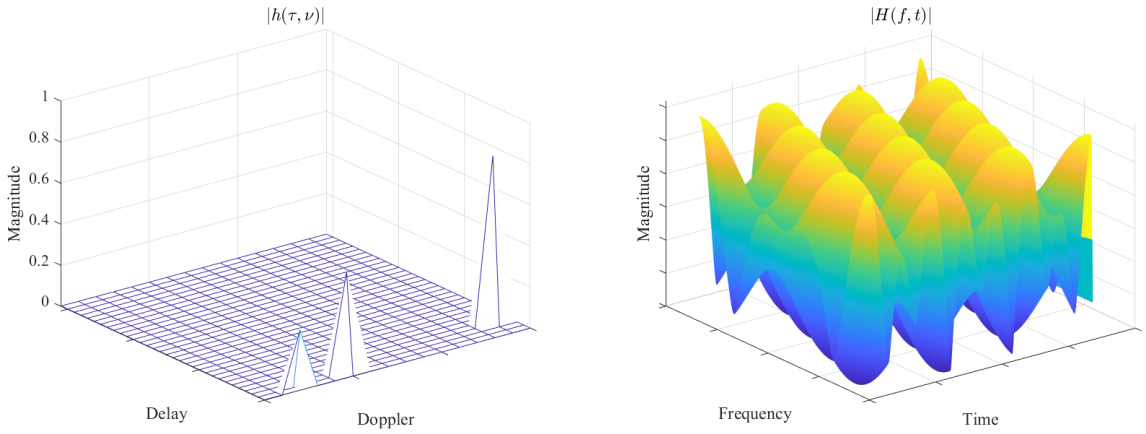


Figure 1.8: Delay-Doppler and frequency-time channel representations, with $P = 3$ paths.

Whereas, if we apply the Fourier transform along the time dimension of $g(\tau, t)$ expressed in (1.24), we obtain the delay-Doppler channel response $h(\tau, \nu)$:

$$h(\tau,\nu) = \int_{-\infty}^{\infty} g(\tau,t) e^{-j2\pi\nu t} dt = \sum_{i=1}^{P} h_i \delta_i(\tau - \tau_i)\delta_i(\nu - \nu_i)\, e^{j2\pi\nu_i\tau_i}. \tag{1.26}$$

The term $e^{j2\pi\nu_i\tau_i}$ is a constant phase shift that can be absorbed into the complex gain h_i. From (1.26) we note that the channel is entirely defined by the parameters (h_i, τ_i, ν_i) for $i = 1, \ldots, P$. These two representations, $H(f, t)$ and $h(\tau, \nu)$, can be directly connected through the 2D symplectic Fourier transform (SFT) and the inverse symplectic Fourier transform (ISFT), as shown in Fig. (1.7), in the following way:

$$h(\tau,\nu) = \text{SFT}\{H(f,t)\} = \mathcal{F}_f^{-1}\{\mathcal{F}_t\{H(f,t)\}\}, \tag{1.27}$$

$$H(f,t) = \text{ISFT}\{h(\tau,\nu)\} = \mathcal{F}_\tau\{\mathcal{F}_\nu^{-1}\{h(\tau,\nu)\}\}. \tag{1.28}$$

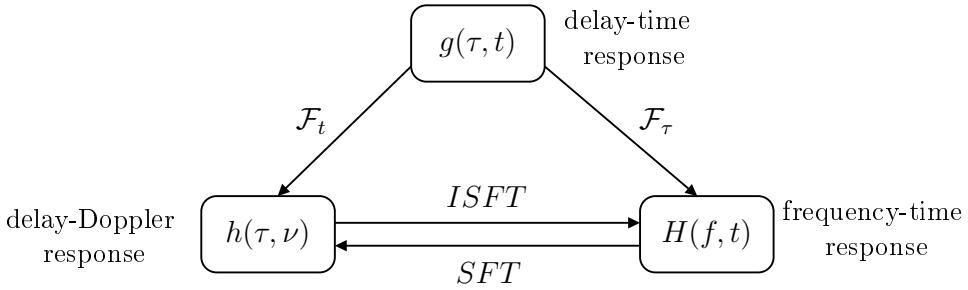


Figure 1.7: Different representations of the channel impulse response and respective relations [15].

Fig. 1.8 provides a graphical comparison of the two channel representations, for the sim-

ulated case of $P = 3$ propagation paths. In OFDM-based systems, the frequency-time channel response is typically adopted, however, as we can notice in the right-hand plot, this tends to be highly varying in doubly dispersive scenarios, requiring more coefficients to accurately describe the channel. Furthermore, since the channel response is time-variant, $H(f, t)$ must be frequently re-estimated and this means that many pilot resources have to be employed and the channel estimation becomes come complex. On the other hand, the delay-Doppler response $h(\tau, \nu)$ offers a much sparser representation of the channel due to the fact that common wireless propagation environments are characterized by a limited number of reflectors that carry the most significant portion of the received power.

2 Multicarrier Modulations

In recent years, there has been a growing trend toward the development of high-speed transmission systems capable of supporting high bit rates B_r. This has led to the use of large bandwidths, causing the transmitted signals to undergo frequency-selective fading, as the channel's coherence bandwidth B_{co} is smaller than the system bandwidth B. The basic idea of multicarrier modulation is to break this wideband system into M different subchannels, each with bandwidth $B_M = B/M$ and data rate $B_{r_M} \approx B_r/M$. Typically the subchannels are orthogonal under ideal propagation conditions. For M sufficiently large, the subchannel bandwidth $B_M \ll B_{co}$, which guarantees flat fading on each subchannel. This can also be seen in the time-domain: the symbol time T_M of the modulated signal in each subchannel is proportional to the subchannel bandwidth $1/B_M$. So $B_M \ll B_{co}$ implies that $T_M \approx 1/B_M \gg 1/B_{co} \approx \Delta\tau$, where $\Delta\tau$ is the delay spread of the channel. Hence, if M is large, the symbol time is much greater than the delay spread, meaning that each subchannel experiences little ISI degradation.

2.1 Preliminaries

In this section, the discrete Fourier transform (DFT) and the inverse discrete Fourier transform (IDFT), together with the CP technique, are introduced, as they play a fundamental role in the implementation of multicarrier modulation schemes.

2.1.1 The DFT and Its Properties

The development of simple and cheap implementations of the DFT and the IDFT, combined with the fact that multicarrier modulation could be implemented with these algorithms, sparked their widespread use. In this section, we review the basic properties of the DFT [16]. Consider a discrete-time sequence $s[m]$, $0 \leq m \leq M-1$, its M-point DFT is defined as:

$$\mathrm{DFT}\{s[m]\} = S[i] = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} s[m]\, e^{-j\frac{2\pi mi}{M}}, \quad 0 \leq i \leq M-1. \tag{2.1}$$

The matrix representation of the DFT, which will be utilized in the subsequent sections, is denoted by $\mathbf{F}_M \in \mathbb{C}^{M\times M}$ and defined as follows:

$$\mathbf{F}_M = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{M-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{M-1} & \omega^{2(M-1)} & \cdots & \omega^{(M-1)(M-1)} \end{bmatrix}, \tag{2.2}$$

where $\omega = e^{-j\frac{2\pi}{M}}$. The DFT is the discrete-time equivalent to the continuous-time Fourier transform and $S[i]$ represents the frequency domain of the time samples $s[m]$ associated with the original signal $s(t)$. The time-domain sequence $s[m]$ can be recovered using the IDFT:

$$\text{IDFT}\{S[i]\} = s[m] = \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} S[i]\, e^{j\frac{2\pi mi}{M}}, \quad 0 \le m \le M-1. \tag{2.3}$$

The correspondent IDFT matrix is given by $\mathbf{F}_M^{-1} = \mathbf{F}_M^H$, with $\mathbf{F}_M^H$ being the conjugate transpose (i.e. Hermitian conjugate) of $\mathbf{F}_M$. When an input data stream $s[m]$ is sent through a linear time-invariant (LTI) discrete-time channel with impulse response $h[m]$, the output $r[m]$ can be characterized by means of the discrete-time linear convolution as:

$$r[m] = h[m] * s[m] = \sum_k h[k]\, s[m-k]. \tag{2.4}$$

We also define the M-point circular convolution as:

$$r[m] = s[m] \circledast h[m] = \sum_k h[k]\, s[m-k]_M \tag{2.5}$$

where $s[m-k]_M$ is a periodic version of $s[m-k]$ with period M and $r[m]$ given by (2.5) is also periodic with period M. A well known DFT property states that the circular convolution in time is equivalent to point-wise multiplication in frequency:

$$R[i] = \text{DFT}\{\, r[m] = s[m] \circledast h[m] \,\} = S[i]\, H[i], \quad 0 \le i \le M-1. \tag{2.6}$$

where $H[i]$ represents the frequency response of the channel. With reference to (2.6), if the channel and input are circularly convoluted then, if $h[m]$ is known at the receiver, the original sequence $s[m]$ can be recovered by applying the IDFT of $R[i]/H[i]$. Unfortunately, the channel output $r[m]$ is not a circular convolution but a linear convolution. Luckily, the linear convolution between the transmitted sequence and the channel impulse response can be turned into a circular convolution by adding a CP.

2.1.2 The Cyclic Prefix (CP)

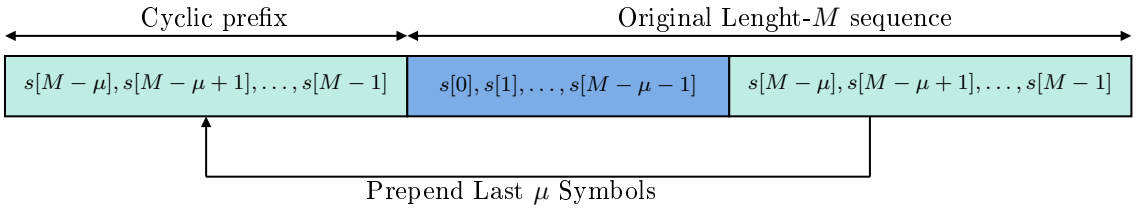


Figure 2.1: CP of length μ.

Consider now a discrete-time channel with finite impulse response (FIR) $h[m] = h[0], \ldots, h[\mu]$ of length $\mu + 1$, the CP for $s[m]$ is defined as $s[M-\mu], \ldots, s[M-1]$ and it consists of the last μ values of the $s[m]$ sequence prepended to it. As shown in Fig. 2.1, this yields a new sequence $\tilde{s}[m], -\mu \leq m \leq M-1$, of length $M+\mu$, where $\tilde{s}[-\mu], \ldots, \tilde{s}[M-1] = s[M-\mu], \ldots, s[M-1], s[0], ..., s[M-1]$. Using this definition, we can write $\tilde{s}[m] = s[m]_M$ for $-\mu \leq m \leq M-1$, which implies that $\tilde{s}[m-k] = s[m-k]_M$ for $-\mu \leq m-k \leq M-1$. Now, the channel output $r[m]$, $0 \leq m \leq M-1$, can be expressed as:

$$\begin{aligned} r[m] &= \tilde{s}[m] * h[m] \\ &= \sum_{k=0}^{\mu} h[k]\, \tilde{s}[m-k] \\ &= \sum_{k=0}^{\mu} h[k]\, s[m-k]_M \\ &= s[m] \circledast h[m] \end{aligned}$$

where the third equality follows from the fact that, for $0 \leq k \leq \mu$, $\tilde{s}[m-k] = s[m-k]_M$ for $0 \leq m \leq M-1$. Therefore, by appending a cyclic prefix to the transmitted sequence $s[m]$, the linear convolution with the channel impulse response effectively results in a circular convolution [16]. By taking the DFT of $r[m]$, in the absence of noise, we have:

$$R[i] = \mathrm{DFT}\{\, r[m] = s[m] \circledast h[m] \,\} = S[i]\, H[i], \quad 0 \leq i \leq M-1$$

and the time domain sequence $s[m]$ can be easily recovered as:

$$s[m] = \mathrm{IDFT}\left\{ \frac{R[i]}{H[i]} \right\} = \mathrm{IDFT}\left\{ \frac{\mathrm{DFT}\{r[m]\}}{\mathrm{DFT}\{h[m]\}} \right\}. \tag{2.7}$$

Assuming that the input $s[m]$ is divided into blocks of size M with a cyclic prefix appended to each one of them to form $\tilde{s}[m]$, then the first μ samples of $r[m] = h[m] * \tilde{s}[m]$ are

corrupted by ISI associated with the last μ samples of $s[m]$ in the prior block, as illustrated in Fig. 2.2. Therefore, the cyclic prefix has the purpose of eliminating ISI between the data blocks, since the first μ samples of the channel output affected by ISI are discarded. However, the CP addition introduces an overhead of μ/M that results in a data-rate reduction of $M/(\mu + M)$. This overhead can be reduced if the symbol duration grows larger.

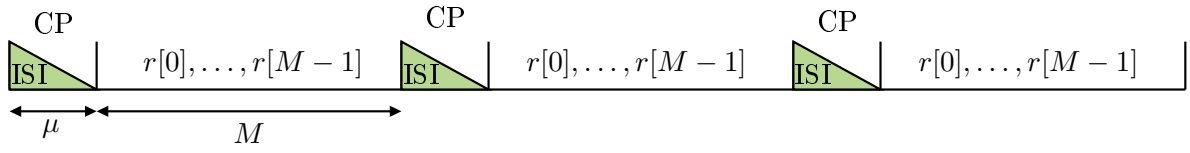


Figure 2.2: ISI between data blocks in channel output.

2.2 Orthogonal Frequency-Division Multiplexing (OFDM)

In this section we focus on the widely used OFDM modulation scheme. In order to compare the OFDM and the OTFS schemes in the following chapters, we assume an $M \times N$ system, where M is the number of the subcarriers and N denotes the time slots, respectively. The OFDM signal has a total bandwidth of $B = M\Delta f$ where $\Delta f = 1/T$ represents the subcarrier spacing and $T = MT_s$ is the OFDM symbol duration. The frame is composed of N OFDM symbols and its duration is $T_f = NT = NMT_s$, with $T_s = 1/f_s$ being the sampling interval and f_s the sampling frequency. We consider a static multipath channel, where $\tau_{\max} = \Delta\tau$ is the maximum tolerable delay spread (assuming that $\tau_{\min} = 0$) and $l_{\max} < M$ is the maximum channel delay tap. Typically, OFDM adopts a CP of sufficient length $L_{CP} \geq l_{\max}$ to avoid ISI. In this analysis, we choose $L_{CP} = l_{\max}$. Fig. 2.3 shows the transmitter block diagram, containing NM QAM modulated information symbols. These are arranged in the frequency-time plane (M subcarriers and N time slots), forming a matrix $\mathbf{X}_{\text{FT}} \in \mathbb{C}^{M\times N}$. The OFDM chain starts with applying an M-point inverse fast Fourier transform (IFFT) along the columns of the matrix $\mathbf{X}_{\text{FT}}$, yielding the time-domain matrix $\tilde{\mathbf{X}} \in \mathbb{C}^{M\times N}$. Assuming to apply rectangular pulse shaping, we can then perform a parallel-to-serial (P/S) conversion, leading to each OFDM symbol vector $\mathbf{s} \in \mathbb{C}^{M\times 1}$. The last step consists in prepending to each $\mathbf{s}$ a CP of length $L_{CP} = l_{\max}$. Subsequently, by means of digital-to-analog converter (DAC) and frequency up-conversion, the bandpass signal is transmitted over the channel.

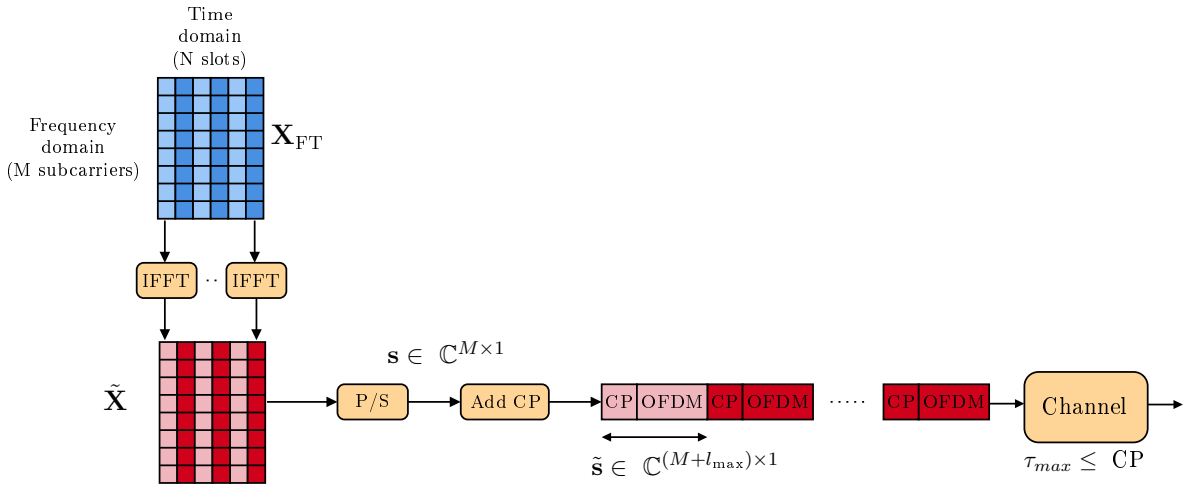


Figure 2.3: OFDM transmitter [15].

2.2.1 OFDM frequency domain input–output relation

As already mentioned in 1.2.1, a static multipath channel with P distinct delay paths, has a baseband equivalent impulse response given by:

$$h(t) = \sum_{i=1}^{P} h_i \, \delta(t - \tau_i)$$

where h_i and τ_i represent the complex path gain and the delay associated with the i-th path. As shown in Fig. 2.4, at the receiver side, after CP removal and serial-to-parallel (S/P) conversion, the received vector $\mathbf{r} \in \mathbb{C}^{M\times 1}$ can be expressed as:

$$\mathbf{r} = \mathbf{h} \circledast \mathbf{s} + \mathbf{w} \tag{2.8}$$

where $\mathbf{h}$ is the discrete-time equivalent channel vector $\mathbf{h} = [\mathbf{h}[0], \ldots, \mathbf{h}[M-1]] = [h_0, \ldots, h_{\mathrm{lmax}}, 0, \ldots, 0]$, $\mathbf{s} \in \mathbb{C}^{M\times 1}$ is the transmitted vector and $\mathbf{w}$ represents the AWGN noise vector of length M. Equivalently, in matrix form we have:

$$\mathbf{r} = \mathbf{G}\mathbf{s} + \mathbf{w} \tag{2.9}$$

with $\mathbf{G}$ being a $M \times M$ circulant matrix. Thanks to the circulant property, $\mathbf{G}$ can be diagonalized as:

$$\mathbf{G} = \mathbf{F}_M^H \mathbf{H} \mathbf{F}_M \tag{2.10}$$

where $\mathbf{H} \in \mathbb{C}^{M\times M}$ is the diagonal matrix given by:

$$\mathbf{H} = \mathrm{diag}(\mathbf{F}_M \, \mathbf{h}). \tag{2.11}$$

Substituting (2.10) into (2.9), leads to the discrete-time domain relation:

$$\mathbf{r} = \mathbf{G}\mathbf{s} + \mathbf{w} = \mathbf{F}_M^H \mathbf{H} \mathbf{F}_M \, \mathbf{s} + \mathbf{w}. \tag{2.12}$$

After S/P conversion and rectangular pulse shaping, the matrix $\tilde{\mathbf{Y}} \in \mathbb{C}^{M \times N}$ is obtained.

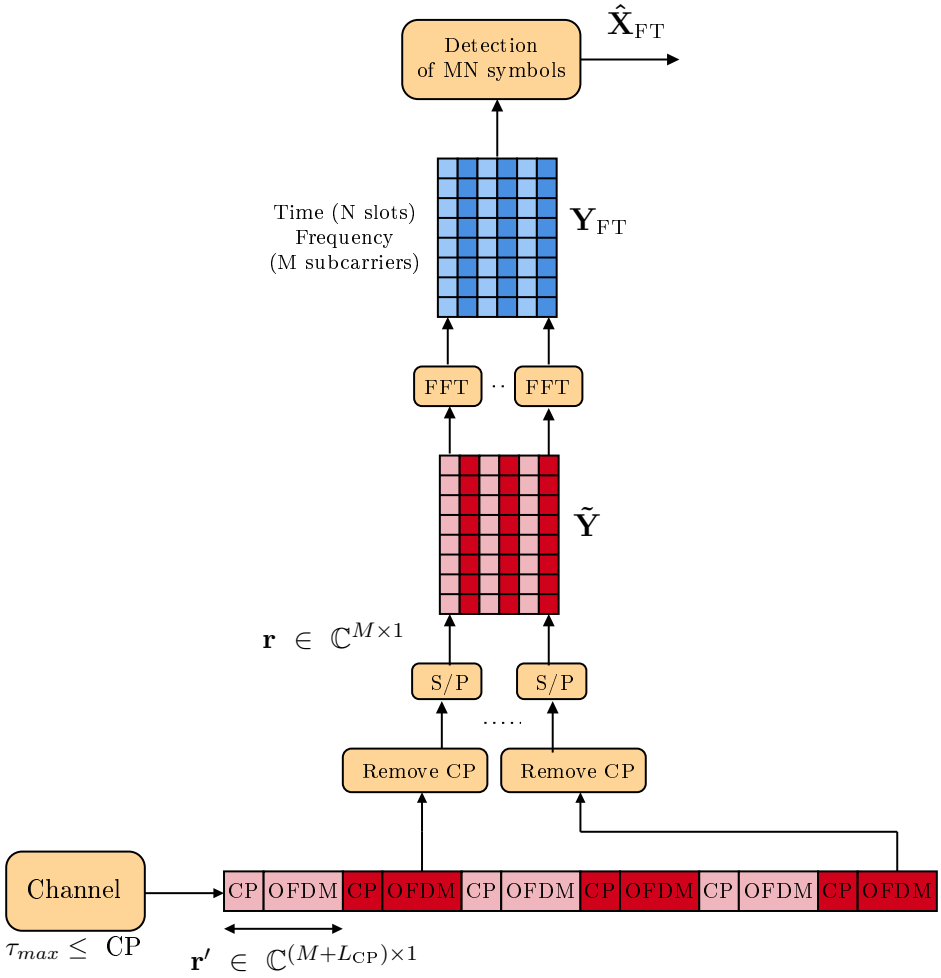


Figure 2.4: OFDM receiver [15].

The fast Fourier transform (FFT) operation is now applied along the columns of $\tilde{\mathbf{Y}}$, yielding the frequency domain received column vector $\mathbf{y}_{\mathrm{FT}} \in \mathbb{C}^{M \times 1}$ of the matrix $\mathbf{Y}_{\mathrm{FT}} \in \mathbb{C}^{M \times N}$, expressed as:

$$\mathbf{y}_{\mathrm{FT}} = \mathbf{F}_M \, \mathbf{r} = \mathbf{H}\mathbf{x}_{\mathrm{FT}} + \bar{\mathbf{w}} \tag{2.13}$$

where $\bar{\mathbf{w}} = \mathbf{F}_M \, \mathbf{w}$ is the i.i.d. noise vector and $\mathbf{x}_{\mathrm{FT}} = \mathbf{F}_M \, \mathbf{s}$. At this point, the single-tap equalizer can be adopted to estimate the symbol vector $\hat{\mathbf{x}}_{\mathrm{FT}} \in \mathbb{C}^{M \times 1}$, of the matrix $\hat{\mathbf{X}}_{\mathrm{FT}} \in \mathbb{C}^{M \times N}$, as:

$$\hat{\mathbf{x}}_{\mathrm{FT}} = \mathbf{H}^{-1} \, \mathbf{y}_{\mathrm{FT}}. \tag{2.14}$$

2.2.2 Frequency equalization

The advantage of multicarrier modulation is that each subchannel is narrowband, resulting in the mitigation of the ISI effect. However, each subchannel experiences flat fading, which can cause large bit error rates. In OFDM, the demodulator can use the channel gains to recover the original QAM symbols by dividing out these gains: $\hat{x}_{\mathrm{FT}}[i] = y_{\mathrm{FT}}[i]/H[i]$. However, for OFDM, frequency equalization leads to noise enhancement because the noise in the i-th subchannel is also scaled by $1/H[i]$ and it is, thus, enhanced. Hence, while the effect of flat fading on $\hat{x}_{\mathrm{FT}}[i]$ is removed by this equalization, the resultant SNR on the i-th subchannel after frequency equalization is the same as before [16].

2.2.3 Advantages and disadvantages of OFDM

In summary, OFDM has the following key features:

1. OFDM adopts multiple orthogonal subcarriers to carry information;
2. a CP of length $l_{\max}$ is prepended to each OFDM symbol to overcome ISI;
3. the CP provides a circulant channel matrix structure, which enables low-complexity single-tap equalization.

On the other hand, OFDM has some disadvantages that need to be considered, for example high PAPR, high Out of Band (OOB) radiation and sensitivity to frequency offset. If we consider OFDM in a high-mobility multipath channel, where each path has different Doppler shifts ν_i, multiplying the channel matrix $\mathbf{G}$ by $\mathbf{s}$ implies that the l_i-th path introduces an l_i-step cyclic shift of the transmitted signal vector $\mathbf{s}$, and modulates it with a carrier at frequency ν_i. The result is that multiple Dopplers make $\mathbf{G}$ no longer a circulant matrix and, as a consequence, single-tap equalization is no longer possible. Moreover, severe ICI appears, resulting in performance degradation.

2.3 Orthogonal Time Frequency Space Modulation (OTFS)

OFDM is known to have good robustness and high spectral efficiency for slowly time-varying frequency selective channels. However, in high-mobility environments, the channels are affected by large Doppler spreads and short coherence time. Under such conditions, off the shelf OFDM is no longer effective and suffers heavy performance degradations, due to the high CP overhead and to the ineffectiveness of single-tap equalization. On the other hand, OTFS modulation is a promising alternative waveform for addressing this challenge. This modulation was first proposed in [1], where authors demonstrated

that, by directly embedding the information in delay-Doppler domain, it is possible to leverage the appealing properties of the delay-Doppler channel representation.

2.3.1 Pre-coded OFDM

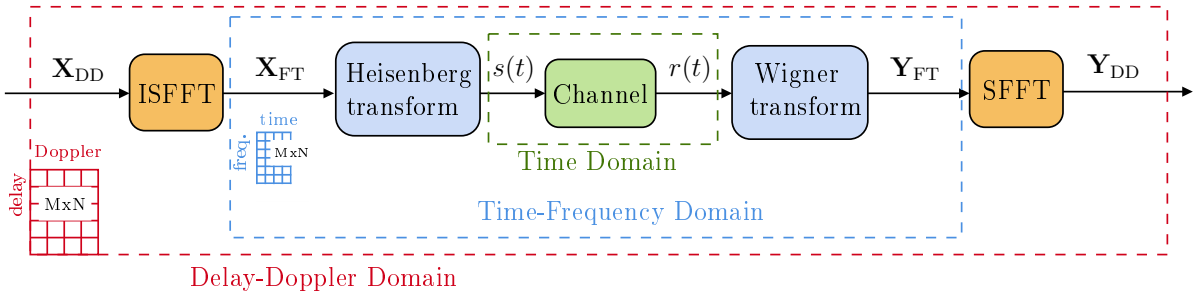


Figure 2.5: Pre-coded OFDM block diagram.

The first proposed OTFS scheme was constructed as a pre-coded version of the OFDM modulation employing N successive OFDM symbols, [15]. In particular, as illustrated in Fig. 2.5, the OFDM system is modified by introducing two outer processing blocks, performing the inverse symplectic fast Fourier transform (ISFFT) and the symplectic fast Fourier transform (SFFT), respectively. At the transmitter, NM information symbols are taken from a modulation alphabet $\mathcal{C}_X$ and are placed in the matrix $\mathbf{X}_{\mathrm{DD}} \in \mathbb{C}^{M\times N}$. These symbols should be considered in the delay-Doppler domain in the analogous way for which input symbols to an OFDM system are considered in the frequency-time domain. Subsequently, the symbols in $\mathbf{X}_{\mathrm{DD}}$ are mapped to the frequency-time domain matrix $\mathbf{X}_{\mathrm{FT}}$ via the ISFFT operation:

$$\mathbf{X}_{\mathrm{FT}}[l,k] = \frac{1}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \mathbf{X}_{\mathrm{DD}}[m,n]\, e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)} \tag{2.15}$$

for $l = 0, \ldots, M-1$, $k = 0, \ldots, N-1$. The ISFFT consists in a 2D transformation which applies an M-point DFT along the columns of $\mathbf{X}_{\mathrm{DD}}$ and an N-point IDFT on the rows. Therefore, equation (2.15) can also be written by means of the DFT and IDFT matrices, $\mathbf{F}_M$ and $\mathbf{F}_N^H$, as follows:

$$\mathbf{X}_{\mathrm{FT}} = \mathbf{F}_M \cdot \mathbf{X}_{\mathrm{DD}} \cdot \mathbf{F}_N^H. \tag{2.16}$$

Next, the Heisenberg transform is applied; this block can be thought as an OFDM modulator employing a general pulse shaping waveform. This operation converts the 2D samples in $\mathbf{X}_{\mathrm{FT}}$ from the frequency-time domain to a continuous-time signal $s(t)$ using a transmit

waveform $g_{\text{tx}}(t)$, leading to:

$$s(t) = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \mathbf{X}_{\text{FT}}[l,k]\, g_{\text{tx}}(t-kT)\; e^{j2\pi l\, \Delta f\, (t-kT)} \tag{2.17}$$

where $\Delta f = 1/T$ is the subcarrier spacing, $T = MT_s$ is the duration of each block and T_s is the sampling interval. Once the signal is received, a matched filter computes the cross-ambiguity function $A_{g_{\text{rx}},r}(\tau,\nu)$:

$$A_{g_{\text{rx}},r}(\tau,\nu) = \int_{-\infty}^{\infty} r(t)\, g^*_{\text{rx}}(t-\tau)\, e^{-j2\pi\nu(t-\tau)}\, dt. \tag{2.18}$$

The function $A_{g_{\text{rx}},r}(\tau,\nu)$ is then sampled N times, at integers multiples of T, for M different Doppler shifts multiples of Δf, leading to:

$$\mathbf{Y}_{\text{FT}}[l,k] = A_{g_{\text{rx}},r}(\tau,\nu)\,\big|_{\tau=k\,T,\;\nu=l\Delta f}. \tag{2.19}$$

$\mathbf{Y}_{\text{FT}}$ has MN entries and (2.18),(2.19) are jointly performed by the Wigner transform block. Finally, the received delay-Doppler matrix $\mathbf{Y}_{\text{DD}} \in \mathbb{C}^{M\times N}$ can be obtained by applying the SFFT on $\mathbf{Y}_{\text{FT}}$ as:

$$\mathbf{Y}_{\text{DD}}[m,n] = \frac{1}{\sqrt{NM}} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \mathbf{Y}_{\text{FT}}[l,k]\, e^{-j2\pi\left(\frac{nk}{N}-\frac{ml}{M}\right)}. \tag{2.20}$$

The SFFT operation corresponds to a 2D transformation which performs an M-point IDFT along the columns of $\mathbf{Y}_{\text{FT}}$ and an N-point DFT of the rows. As for (2.15), equation (2.20) can also be rewritten in terms of the DFT and IDFT matrices, $\mathbf{F}_N$ and $\mathbf{F}_M^H$, as follows:

$$\mathbf{Y}_{\text{DD}} = \mathbf{F}_M^H \cdot \mathbf{Y}_{\text{FT}} \cdot \mathbf{F}_N. \tag{2.21}$$

Since we want to employ the OTFS system in high-mobility channels, additional operations are needed at the receiver side in order to equalize and recover the original transmitted symbols.

2.3.2 IDZT-based OTFS transceiver

In this section, we introduce a less complex, yet equivalent, OTFS transceiver. Recalling the transmitted information symbol matrix $\mathbf{X}_{\text{DD}} \in \mathbb{C}^{M\times N}$ in the delay-Doppler domain, through equation (2.16) we obtain the frequency-time domain matrix $\mathbf{X}_{\text{FT}} \in \mathbb{C}^{M\times N}$. Next, we can split the Heisenberg transform in two steps: the first one converts $\mathbf{X}_{\text{FT}}$ in

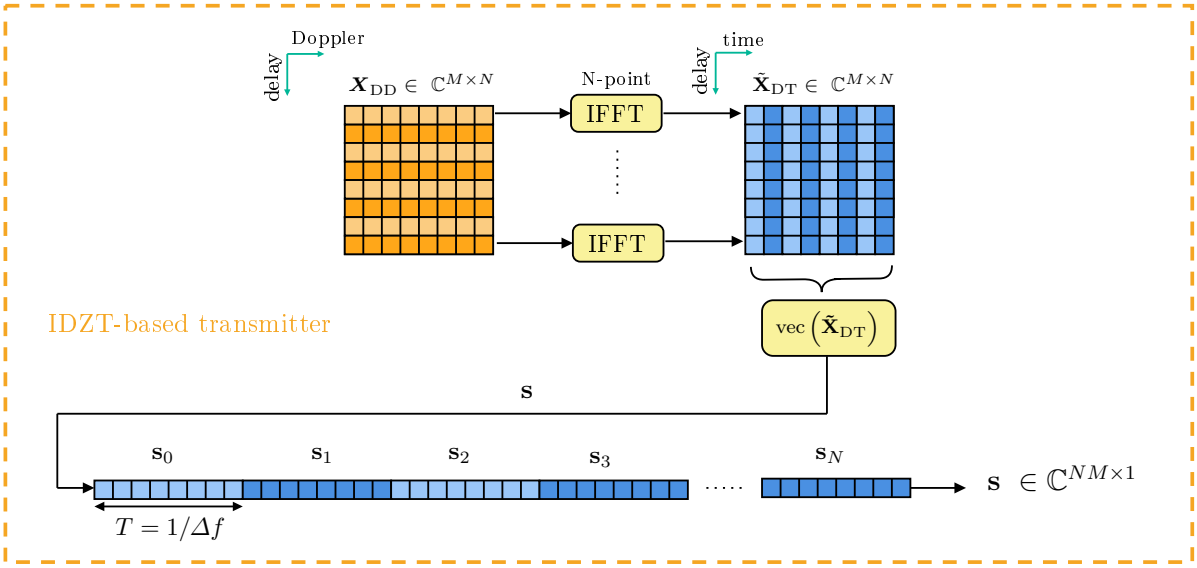


Figure 2.6: IDZT block using rectangular pulse shaping.

the delay-time domain matrix $\tilde{\mathbf{X}}_{\mathrm{DT}} \in \mathbb{C}^{M\times N}$ by applying the M-point IFFT as:

$$\tilde{\mathbf{X}}_{\mathrm{DT}} = \mathbf{F}_M^H \cdot \mathbf{X}_{\mathrm{FT}} = \underbrace{\mathbf{F}_M^H \mathbf{F}_M}_{\mathbf{I}_M} \cdot \mathbf{X}_{\mathrm{DD}} \cdot \mathbf{F}_N^H = \mathbf{X}_{\mathrm{DD}} \cdot \mathbf{F}_N^H \tag{2.22}$$

where the second equality is due to (2.16) and where $\mathbf{I}_M$ is the $M \times M$ identity matrix. The next step of the Heisenberg transform performs pulse shaping on $\tilde{\mathbf{X}}_{\mathrm{DT}}$ followed by the concatenation of all its columns, obtaining the NM length time-domain vector

$$\mathbf{s} = \mathrm{vec}\Big(\mathbf{G}_{\mathrm{tx}}\, \tilde{\mathbf{X}}_{\mathrm{DT}}\Big) \in \mathbb{C}^{NM\times 1}. \tag{2.23}$$

By assuming a rectangular frequency-time pulse shaping waveform, i.e. a finite length sinc function in the time-domain, we have that the diagonal matrix $\mathbf{G}_{\mathrm{tx}}$ reduces to $\mathbf{G}_{\mathrm{tx}} = \mathbf{I}_M$. In summary, we can express the transmitted vector $\mathbf{s} \in \mathbb{C}^{NM\times 1}$ as:

$$\mathbf{s} = \mathrm{vec}\Big(\tilde{\mathbf{X}}_{\mathrm{DT}}\Big) = \mathrm{vec}\big(\mathbf{X}_{\mathrm{DD}} \cdot \mathbf{F}_N^H\big) \tag{2.24}$$

with entries:

$$\mathbf{s}[q] = \mathbf{s}[m + nM] = \frac{1}{\sqrt{N}} \sum_{p=0}^{N-1} \mathbf{X}_{\mathrm{DD}}[m,p]\, e^{j2\pi \frac{pn}{N}} \tag{2.25}$$

where $q = m+nM$ for $m = 0,\ldots,M-1$ and $n = 0,\ldots,N-1$. This latter representation is knows as inverse discrete Zak transform (IDZT) and the operations performed in (2.24) or equivalently in (2.25) can be logically represented into a single processing block, as shown in Fig. 2.6. In particular, the IDZT block converts the delay-Doppler domain matrix

$\mathbf{X}_{\mathrm{DD}}$ in the discrete-time domain vector $\mathbf{s}$. The complete IDZT-based block diagram is shown in Fig. 2.7, where the discrete-time domain vector $\mathbf{s}$ is digital-to-analog converted and transmitted over the channel. The inverse operation is done by the analog-to-digital converter (ADC) at the receiver side and the time-domain vector $\mathbf{r}$ is directly related to the delay-Doppler domain matrix $\mathbf{Y}_{\mathrm{DD}} \in \mathbb{C}^{M\times N}$ by the discrete Zak transform (DZT) as:

$$\mathbf{Y}_{\mathrm{DD}} = \left(\mathrm{vec}^{-1}_{M,N}(\mathbf{r})\right) \cdot \mathbf{F}_N \tag{2.26}$$

where vec^{-1} takes the vector $\mathbf{r}$ and reshapes it into the delay-time matrix $\tilde{\mathbf{Y}}_{\mathrm{DT}} \in \mathbb{C}^{M\times N}$. Specifically, $\mathbf{Y}_{\mathrm{DD}} \in \mathbb{C}^{M\times N}$ has entries:

$$\mathbf{Y}_{\mathrm{DD}}[m,n] = \frac{1}{\sqrt{N}} \sum_{p=0}^{N-1} \mathbf{r}[m+pM]\, e^{-j2\pi \frac{np}{N}} \tag{2.27}$$

for $m = 0,\ldots,M-1$ and $n = 0,\ldots,N-1$.

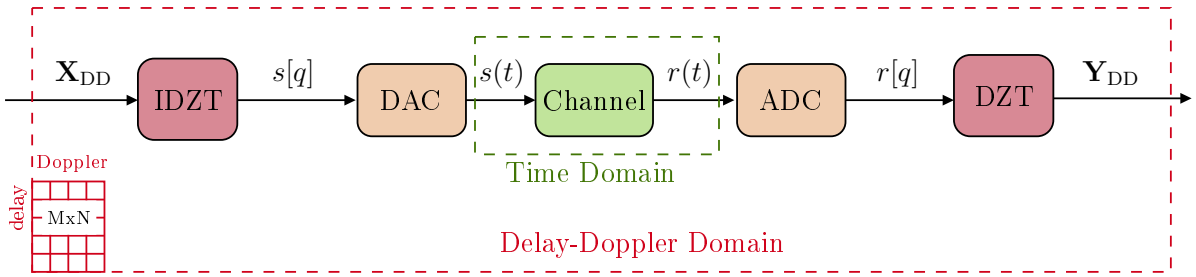


Figure 2.7: IDZT-based OTFS block diagram.

Crystallization condition

The Zak transform is a signal transformation that can map a mono-dimensional continuous time signal to a 2D delay-Doppler representation. An important property of the Zak transform is that it is periodic along the Doppler domain, expressed by ν, with period $\Delta f = 1/T$ and quasi-periodic along the delay dimension, represented by τ, with period T. Due to this periodicity, the representation of a signal in the delay-Doppler domain results in replicas across the delay and Doppler axes [15]. To avoid aliasing between these replicas, it is essential that the support of the channel delay-Doppler response lies entirely within the fundamental region of size $\{\tau \in [0,T),\ \nu \in [0,1/T)\}$. In order to guarantee no interference between replicas, a necessary condition has to hold:

$$\tau_{\max}\, \nu_{\max} \ll 1$$

which can also be seen as

$$\tau_{\max} \leq T \quad \text{and} \quad \nu_{\max} \leq 1/T = \Delta f. \tag{2.28}$$

where $\nu_{\max} = \Delta\nu$ is the Doppler spread (assuming that $\nu_{\min} = 0$). This ensures that the channel is underspread. When moving to the discrete domain, the signal is sampled over a grid of size $M \times N$, with resolutions $1/M\Delta f$ and $1/NT$ on the delay and Doppler axes, respectively. The discrete counterparts of (2.28) become:

$$\tau_{\max} = T_s \cdot \ell_{\max} \tag{2.29}$$

$$\nu_{\max} = \frac{\kappa_{\max}}{T_f} \tag{2.30}$$

where $T_f = NMT_s = NT$ is the frame duration, $\ell_{\max} \in \mathbb{R}$ is the normalized delay associated with $\tau_{\max}$ and $\kappa_{max} \in \mathbb{R}$ is the normalized Doppler associated with ν_{max}. Recalling that $T = T_s M$ and combining (2.28) with (2.29), we get:

$$T_s \cdot \ell_{\max} \leq T_s \cdot M \rightarrow \ell_{\max} \leq M$$

and by merging (2.28) and (2.30) it follows:

$$\frac{\kappa_{\max}}{T_f} \leq \frac{1}{T} \rightarrow \kappa_{\max} \leq N.$$

Thus, to satisfy the crystallization condition for the OTFS system [17], we must select M and N such that the channel footprint in delay-Doppler domain fits within the grid. When this property holds, symbols in the delay-Doppler domain see the same channel and at the receiver they are shifted and scaled in the same way, up to phase rotations. Hence, the channel parameters h_i, τ_i, and ν_i can be considered constant for the duration of the frame T_f.

2.3.3 Discrete-time input-output relations

We now summarize a set of input–output relations, following the framework presented in [15, Chapter 4], which will be used in the next sections. Continuing the analysis from section 1.2.3, when modeling a bandpass communication system, it is convenient to work with a discrete baseband equivalent representation of the system. At the receiver side, the impaired signal is down-converted to baseband and sampled at $f_s = B = M\Delta f$ Hz, yielding NM complex samples per frame of duration $T_f = NT$. The received signal $r(t)$ is sampled at $t = qT/M$, where $q = 0, \ldots, NM-1$, and the variable $\tau = lT/M$ is discretized

for $l = 0, \ldots, M-1$, obtaining the discrete input-output relation:

$$\mathbf{r}[q] = \sum_{l} g^s[l,q]\, \mathbf{s}[q-l] + \mathbf{w}[q], \tag{2.31}$$

where $\mathbf{w}[q]$ is the i.i.d. AWGN with variance σ_{w}^2 and the discrete delay-time response $g^s[l,q]$ is given by:

$$g^s[l,q] = g(lT/M, qT/M) = \sum_{\ell \in \mathcal{L}} \Big(\sum_{\kappa \in \mathcal{K}_\ell} \nu_\ell(\kappa)\, z^{\kappa(q-l)} \Big) \operatorname{sinc}(l-\ell) \tag{2.32}$$

with $\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$, $z = e^{j\frac{2\pi}{NM}}$ and $\nu_\ell(\kappa)$ the Doppler response at delay shift $\ell T/M$. The set $\mathcal{L}$ contains the distinct normalized delay shifts ℓ of the channel, and $\mathcal{K}_\ell$ is the set of normalized Doppler shifts κ of all the paths sharing the same delay shift $\ell T/M$. We now consider a time-varying multipath channel of P paths, where the i-th path has channel gain h_i, delay shift τ_i, and Doppler shift ν_i. The Doppler response at delay shift $\ell = \ell_i = \tau_i \cdot (M\Delta f)$ is given as:

$$\nu_\ell(\kappa) = \begin{cases} h_i, & \text{if } \ell = \ell_i = \tau_i \cdot (M\Delta f) \text{ and } \kappa = \kappa_i = \nu_i \cdot (NT), \\ 0, & \text{otherwise.} \end{cases} \tag{2.33}$$

Specifically, when the normalized delay shifts ℓ_i and the normalized Doppler shifts κ_i are integers, we denote them with l_i, and k_i, and we suppose $l_i < l_{max}$. As a consequence, the sinc function in (2.32) is replaced by the unit pulse at l_i yielding the discrete delay-time channel response:

$$\begin{aligned} g^s[l,q] &= \sum_{i=1}^{P} \nu_{l_i}(\kappa_i)\, z^{\kappa_i(q-l)}\, \delta[l-l_i] \\ &= \sum_{i=1}^{P} h_i\, z^{\kappa_i(q-l)}\, \delta[l-l_i]. \end{aligned} \tag{2.34}$$

where the second step is due to (2.33). The vectorized input–output relation in (2.31) can also be written in terms of the time-domain channel matrix $\mathbf{G} \in \mathbb{C}^{NM \times NM}$ as:

$$\mathbf{r} = \mathbf{G} \cdot \mathbf{s} + \mathbf{w} \tag{2.35}$$

where $\mathbf{w} \in \mathbb{C}^{NM \times 1}$ is the time-domain AWGN vector and $\mathbf{s}$ is the transmitted sample vector given in (2.23). The matrix $\mathbf{G}$ has $l_{max}+1$ nonzero diagonal and sub-diagonals

with elements in each sub-diagonal given by:

$$\mathbf{G}[q, q-l] = g^s[l, q], \quad \text{for } l \leq q \leq NM-1 \tag{2.36}$$

and zero elsewhere. Fig. 2.8 shows an example describing the channel matrix $\mathbf{G}$ with

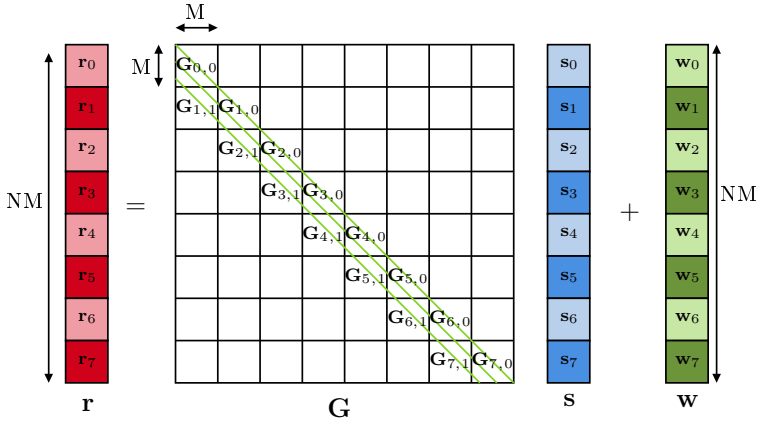


Figure 2.8: Time-domain channel matrix $\mathbf{G}$ with three delay paths ($l_{\max} = 2$) divided into $M \times M$ sub-matrices [15].

three delay paths ($l_{max} = 2$) and $N = 8$, where $\mathbf{s}_n \in \mathbb{C}^{M\times 1}$ and $\mathbf{r}_n \in \mathbb{C}^{M\times 1}$, $n = 0, \ldots, N-1$, denote the transmitted and received time-domain vectors. Then the block-wise input–output relation can be written as:

$$\begin{cases} \mathbf{r}_n = \mathbf{G}_{0,0} \cdot \mathbf{s}_0 + \mathbf{w}_0, & \text{for } n = 0 \\ \mathbf{r}_n = \mathbf{G}_{n,0} \cdot \mathbf{s}_n + \mathbf{G}_{n,1} \cdot \mathbf{s}_{n-1} + \mathbf{w}_n, & \text{for } 1 \leq n \leq N-1 \end{cases} \tag{2.37}$$

where $\mathbf{w}_n \in \mathbb{C}^{M\times 1}$ is the n-th block AWGN vector and $\mathbf{G}_{n,n'} \in \mathbb{C}^{M\times M}$ represents the channel between the n-th received and $(n-n')$-th transmitted blocks.

2.3.4 Delay-Doppler input-output relations

In this section the OTFS delay-Doppler input-output relation in vectorized form is presented [15], assuming rectangular pulse shaping waveforms, i.e. $\mathbf{G}_{\text{tx}} = \mathbf{G}_{\text{rx}} = \mathbf{I}_M$. The time-domain vectors $\mathbf{s}$ and $\mathbf{r}$, expressed in (2.35), are related to the transmitted and received delay-Doppler vectors, $\mathbf{x}$ and $\mathbf{y}$ respectively, as follows:

$$\mathbf{x} = (\mathbf{I}_M \otimes \mathbf{F}_N)\mathbf{P}^T \cdot \mathbf{s} \tag{2.38}$$

$$\mathbf{y} = (\mathbf{I}_M \otimes \mathbf{F}_N)\mathbf{P}^T \cdot \mathbf{r} \tag{2.39}$$

where $\otimes$ is the Kronecker product and $\mathbf{P} \in \mathbb{Z}^{NM \times NM}$ is the row-column interleaver matrix defined as:

$$P = \begin{bmatrix} E_{1,1} & E_{2,1} & \cdots & E_{M,1} \\ E_{1,2} & E_{2,2} & \cdots & E_{M,2} \\ \vdots & \vdots & \ddots & \vdots \\ E_{1,N} & E_{2,N} & \cdots & E_{M,N} \end{bmatrix},$$

with the $M \times N$ matrix $E_{i,j}$ being:

$$E_{i,j}(i', i') = \begin{cases} 1, & \text{if } i' = i \text{ and } i' = j, \\ 0, & \text{otherwise} \end{cases}$$

for $i, i' \in [1, M]$ and $j, j' \in [1, N]$. By defining $\mathbf{A} = (\mathbf{I}_M \otimes \mathbf{F}_N)\mathbf{P}^T$, the result of multiplying a vector by the matrix $\mathbf{A}$ is equivalent to the signal processing operations implemented inside the DZT block by the OTFS receiver as shown in the block diagram of Fig. 2.7. Multiplying both sides of (2.35) by $\mathbf{A}$, leads to:

$$\underbrace{\mathbf{A} \cdot \mathbf{r}}_{\mathbf{y}} = \underbrace{\mathbf{A} \cdot \mathbf{G}\mathbf{A}^{\mathbf{H}}}_{\mathbf{H}} \underbrace{\mathbf{A} \cdot \mathbf{s}}_{\mathbf{x}} + \underbrace{\mathbf{A}\mathbf{w}}_{\mathbf{w}'}. \tag{2.40}$$

More clearly, we obtain:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}' \tag{2.41}$$

where $\mathbf{H} \in \mathbb{C}^{NM \times NM}$ is the delay-Doppler channel matrix and $\mathbf{w}'$ has the same statistical

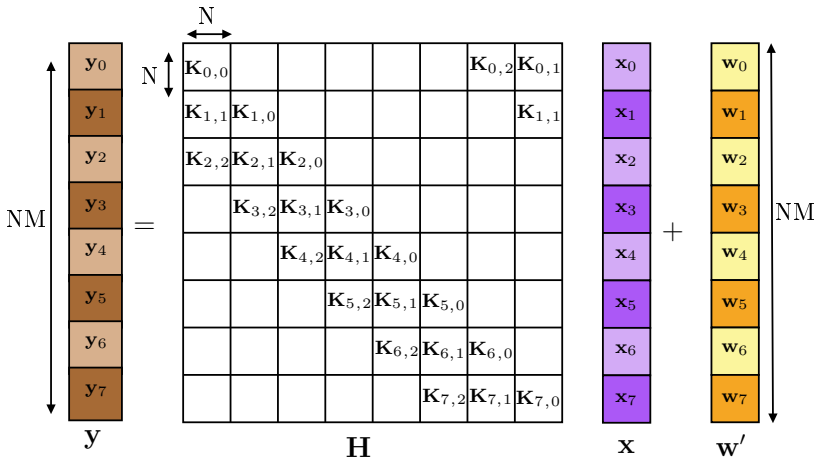


Figure 2.9: Delay-Doppler channel matrix $\mathbf{H}$ with three delay paths ($l_{\max} = 2$) divided into $N \times N$ submatrices [15].

properties of $\mathbf{w}$ since they are related by unitary transformations. Focusing now on the block-wise delay-Doppler input-output relation, we define $\mathbf{K}_{m,l} \in \mathbb{C}^{N \times N}$ the sub-matrices

of $\mathbf{H}$, as shown in Fig. 2.9 and we derive the block-wise relation as:

$$\mathbf{y}_m = \sum_{l \in \mathcal{L}} \mathbf{K}_{m,l} \cdot \mathbf{x}_{[m-l]_M} + \mathbf{w'}_m \tag{2.42}$$

where $\mathbf{x}_m$, $\mathbf{y}_m$ and $\mathbf{w'}_m \in \mathbb{C}^{N \times 1}$ for $m = 0, \ldots, M-1$.

2.3.5 Zero Padded OTFS

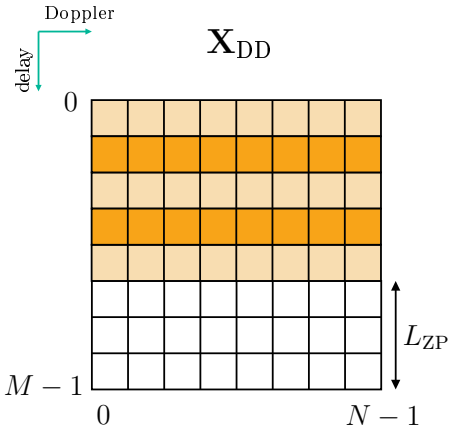


Figure 2.10: ZP-OTFS delay-Doppler matrix.

In this section, we present a particular variant of OTFS in which the number of information symbols in the OTFS frame is reduced to $(M - L_{\mathrm{ZP}}) \times N$, where L_{ZP} denotes the length of the zero samples appended along the columns of $\mathbf{X}_{\mathrm{DD}}$. This reduction ensures that the total frame size remains $M \times N$ after the addition of the zero padding (ZP), as illustrated in Fig. 2.10. At the receiver side, the ZP is not discarded and this ensures that the Doppler shifts remain unscaled in the $M \times N$ delay-Doppler grid. If $M > L_{\mathrm{ZP}} \geq l_{\max}$, the last L_{ZP} samples of each time block $\mathbf{s}_n \in \mathbb{C}^{M \times 1}$ for $n = 0, \ldots, N-1$, will be equal to zero. As a consequence, we can ignore the channel sub-matrices $\mathbf{G}_{n,1}$, for $n = 0, \ldots, N-1$ in Fig. 2.8, since they multiply the zero components of the sub-vectors $\mathbf{s}_n$. Hence, the block-wise input-output relation in (2.37) simplifies to:

$$\mathbf{r}_n = \mathbf{G}_{n,0} \cdot \mathbf{s}_n + \mathbf{w}_n. \tag{2.43}$$

Time-domain LMMSE equalization for ZP-OTFS

In general, the received time-domain signal $r(t)$ is different from the transmitted one $s(t)$, due to the impairments introduced by the channel. In order to minimize and possibly remove the channel distortions, the receiver has to perform some operations that are

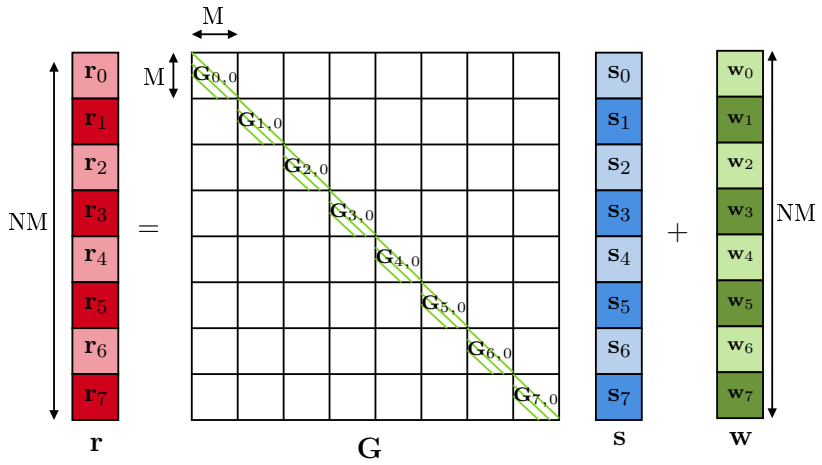


Figure 2.11: ZP-OTFS time-domain channel matrix $\mathbf{G}$ [15].

known as equalization. Recalling the time-domain input–output relation of OTFS in (2.35), the linear minimum mean-square error (LMMSE) estimate of $\mathbf{s}$ is given as:

$$\hat{\mathbf{s}} = \mathbf{G}^H \left(\mathbf{G}\mathbf{G}^H + \frac{\sigma_{\mathrm{w}}^2}{P_s} \mathbf{I}_{MN} \right)^{-1} \mathbf{r}. \tag{2.44}$$

where P_s is the average power of the transmitted symbols. The operation in (2.44) requires inverting an $NM \times NM$ matrix. However, if we adopt ZP-OTFS, the LMMSE computation can be performed block-wise by inverting N complex sub-matrices of size $M \times M$, as illustrated in Fig. 2.11, and recalling the block-wise input–output relation in (2.43), the LMMSE estimate becomes:

$$\hat{\mathbf{s}}_n = \mathbf{G}_{n,0}^H \left(\mathbf{G}_{n,0}\mathbf{G}_{n,0}^H + \frac{\sigma_{\mathrm{w}}^2}{P_s} \mathbf{I}_M \right)^{-1} \mathbf{r}_n \tag{2.45}$$

for $n = 0, \ldots, N-1$. The time-domain LMMSE method of (2.44) has complexity of $O((NM)^3)$. In the case of ZP-OTFS, the LMMSE equalization can be performed block-wise for N blocks, as expressed in (2.45), with an overall complexity of $O(NM^3)$, which is significantly lower than for the LMMSE method in (2.44).

ZP-OTFS delay-Doppler domain analysis

To enable block-wise time-domain LMMSE equalization, channel state information (CSI) must be obtained. In this thesis, we address this by presenting channel estimation techniques in the delay-Doppler domain. To that end, we first derive the input-output relation for ZP-OTFS in this domain, following the approach in [15, Section 4.5]. Starting from the input-output relation derived in (2.42), we can write it omitting the noise term and

introducing the delay-Doppler channel vectors $\boldsymbol{\nu}_{m,l} \in \mathbb{C}^{N\times 1}$ as:

$$\mathbf{y}_m = \sum_{l\in\mathcal{L}} \mathbf{K}_{m,l} \cdot \mathbf{x}_{m-l} = \sum_{l\in\mathcal{L}} \boldsymbol{\nu}_{m,l} \circledast \mathbf{x}_{m-l}, \tag{2.46}$$

where $m = 0, \ldots, M-1$ with

$$\mathbf{x}_{m-l}[n] = 0 \quad \text{for } m < l. \tag{2.47}$$

The second equality in (2.46) follows from the fact that the sub-matrices $\mathbf{K}_{m,l} \in \mathbb{C}^{N\times N}$ are circulant since $m \geq l$ and are expressed as:

$$\mathbf{K}_{m,l} = \text{circ}[\boldsymbol{\nu}_{m,l}]. \tag{2.48}$$

The delay-Doppler channel vectors $\boldsymbol{\nu}_{m,l}$ are related to the delay-time vectors $\tilde{\boldsymbol{\nu}}_{m,l} \in \mathbb{C}^{N\times 1}$ via the Fourier transform as:

$$\boldsymbol{\nu}_{m,l}[k] = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{\boldsymbol{\nu}}_{m,l}[n] e^{-j\frac{2\pi kn}{N}} \tag{2.49}$$

for $0 \leq k \leq N-1$ and $0 \leq m \leq M-1$. $\tilde{\boldsymbol{\nu}}_{m,l}$ can be derived using (2.32) and substituting $q = m + nM$ as:

$$\begin{aligned} \tilde{\boldsymbol{\nu}}_{m,l}[n] &= g^s[l, q] = g^s[l,\, m + nM] \\ &= \sum_{\ell\in\mathcal{L}} \sum_{\kappa\in\mathcal{K}_\ell} \nu_\ell(\kappa)\, z^{\kappa(m+nM-l)}\, \text{sinc}(l-\ell) \\ &= \sum_{\ell\in\mathcal{L}} \sum_{\kappa\in\mathcal{K}_\ell} \nu_\ell(\kappa)\, z^{\kappa(m-l)} e^{j\frac{2\pi\kappa n}{N}}\, \text{sinc}(l-\ell) \end{aligned} \tag{2.50}$$

for $m = 0, \ldots, M-1$ and $n = 0, \ldots, N-1$, recalling that $z = e^{j\frac{2\pi}{MN}}$. Assuming integer delay taps, i.e. $\ell = l \in \mathbb{Z}$, (2.50) becomes:

$$\tilde{\boldsymbol{\nu}}_{m,l}[n] = \sum_{\kappa\in\mathcal{K}_l} \nu_l(\kappa)\, z^{\kappa(m-l)} e^{j\frac{2\pi\kappa n}{N}}, \quad n = 0, \ldots, N-1 \tag{2.51}$$

for $\kappa \in \mathbb{R}$. Considering the case of fractional Doppler shifts, we now substitute (2.51) in (2.49) obtaining:

$$\boldsymbol{\nu}_{m,l}[k] = \frac{1}{N} \sum_{\kappa\in\mathcal{K}_l} \nu_l(\kappa)\, z^{\kappa(m-l)} \sum_{n=0}^{N} e^{j\frac{2\pi\kappa n}{N}} e^{-j\frac{2\pi kn}{N}}$$

$$= \frac{1}{N} \sum_{\kappa \in \mathcal{K}_l} \nu_l(\kappa)\, z^{\kappa(m-l)} \sum_{n=0}^{N} e^{j\frac{2\pi n(\kappa-k)}{N}}$$
$$= \frac{1}{N} \sum_{\kappa \in \mathcal{K}_l} \nu_l(\kappa)\, z^{\kappa(m-l)} \zeta_N(\kappa - k) \tag{2.52}$$

where the periodic sinc function $\zeta(\cdot)$ includes the extra phase and magnitude variations in the delay-Doppler channel vectors due to fractional Doppler shifts, given as:

$$\zeta_N(x) = \sum_{n=0}^{N-1} e^{j\frac{2\pi x n}{N}} = \frac{\sin(\pi x)}{\sin\left(\frac{\pi x}{N}\right)}\, e^{j\pi x \frac{N-1}{N}}. \tag{2.53}$$

If we assume that also the Doppler shifts are integer, the sinc function $\zeta(\cdot)$ evaluates to N when $x = 0$ and zero elsewhere. Therefore, (2.52) simplifies to:

$$\boldsymbol{\nu}_{m,l}[k] = \begin{cases} \nu_l(\kappa)\, z^{\kappa(m-l)}, & \text{if } l = \ell \text{ and } k = [\kappa]_N, \\ 0, & \text{otherwise} \end{cases} \tag{2.54}$$

for $l = \ell \in \mathbb{Z}$ and $\kappa \in \mathbb{Z}$. Since κ can be a negative integer, we ensure $k = [\kappa]_N \in [0, N-1]$. Now, using (2.46) and (2.54), the delay-Doppler relation becomes:

$$\mathbf{y}_m[n] = \sum_{l \in \mathcal{L}} \sum_{k=0}^{N-1} \boldsymbol{\nu}_{m,l}[k]\, \mathbf{x}_{[m-l]_M}[[n-k]_N]$$
$$= \sum_{l \in \mathcal{L}} \sum_{k=-N/2}^{N/2-1} \nu_l(k)\, z^{k(m-l)}\, \mathbf{x}_{[m-l]_M}[[n-k]_N]\,. \tag{2.55}$$

If we consider now a wireless channel with P paths, where each has complex channel gain h_i, normalized integer delay tap l_i and normalized integer Doppler shift k_i, we recall from section 2.3.3 that the Doppler response is:

$$\nu_l(\kappa) = \begin{cases} h_i, & \text{if } l = l_i \text{ and } k = k_i \\ 0, & \text{otherwise} \end{cases} \tag{2.56}$$

and by substituting (2.56) in (2.55), and replacing $\mathbf{y}_m[n]$ and $\mathbf{x}_m[n]$ respectively with $\mathbf{Y}_{\mathrm{DD}}[m,n]$ and $\mathbf{X}_{\mathrm{DD}}[m,n]$, we derive the explicit input-output delay-Doppler relation:

$$\boldsymbol{Y}_{\mathrm{DD}}[m,n] = \sum_{i=1}^{P} h_i\, z^{k_i(m-l_i)}\, \boldsymbol{X}_{\mathrm{DD}}[m-l_i,\, [n-k_i]_N]\,, \quad m \geq l_i \tag{2.57}$$

where $\boldsymbol{X}_{\mathrm{DD}}[m - l_i, n] = 0$ for $m < l_i$.

3 Delay-Doppler Channel Estimation with Embedded Pilot

In this chapter, we address the problem of acquiring CSI at the receiver. Such information is essential for performing the equalization step, which compensates for the distortions introduced by the channel and enables reliable recovery of the transmitted data. Channel estimation techniques can be broadly categorized into data-aided (DA) and non data-aided (NDA) methods. In DA approaches, also referred to as pilot-aided or training-based methods, a set of known symbols (pilots) is inserted into the transmitted signal to facilitate the estimation of the channel response at the receiver. These techniques generally provide high estimation accuracy, at the expense of additional overhead due to the transmission of pilot symbols. Conversely, NDA methods, also known as blind or decision-directed techniques, perform channel estimation without relying on known data. Although NDA schemes improve spectral efficiency by eliminating the pilot overhead, they typically suffer from higher computational complexity, slower convergence, and reduced estimation accuracy compared to DA approaches.

Leveraging the delay–Doppler domain representation of time-varying multipath channels, OTFS is specifically designed to combat the dynamics of these scenarios. Most existing OTFS channel estimation techniques are data-aided, employing pilot symbols embedded in or superimposed on the delay–Doppler grid. In this thesis, a channel estimation method for ZP-OTFS, inspired by [9] and [13], is implemented. The approach assumes that the crystallization condition is satisfied (see section 2.3.2) and employs an embedded pilot-aided scheme that inserts a single pilot and guard symbols within the delay–Doppler grid to estimate both integer and fractional Doppler shifts. Using the received pilot in the delay-Doppler domain, the purpose of channel estimation is to retrieve the parameters h_i, ℓ_i and κ_i (defined in section 2.3.3), which fully characterize the channel-induced distortions. Once these parameters have been estimated, the time-domain channel matrix $\mathbf{G}$ can be constructed and used in the time-domain LMMSE equalization block. We assume integer delay taps, i.e. $\ell_i = l_i$; this assumption comes from the fact that we are considering

wideband systems. Recalling that the resolution of our delay-Doppler grid along the delay domain is $1/M\Delta f$, this is fine enough that the actual delays introduced by the channel can be approximated as integer multiples of the delay grid spacing, without introducing significant error.

3.1 The integer Doppler case

We initially consider the ideal case where the Doppler shifts are integers, i.e. $\kappa_i = k_i$. This is based on the assumption that the Doppler values are integer multiples of the resolution along the Doppler axis, which is $1/NT$. We recall now the input–output relation for ZP-OTFS in (2.57):

$$\boldsymbol{Y}_{\mathrm{DD}}[m,n] = \sum_{i=1}^{P} h_i \, z^{k_i(m-l_i)} \, \boldsymbol{X}_{\mathrm{DD}}\left[m-l_i, \, [n-k_i]_N\right] \tag{3.1}$$

where h_i is the complex channel gain of the i-th path, $l_i \leq l_{\max}$ are the integer delay taps and $|k_i| \leq k_{\max}$ are the integer Doppler taps. Thanks to the addition of zero samples in the ZP-OTFS, in (3.1), we can ignore the case $m < l_i$ since $\boldsymbol{X}_{\mathrm{DD}}[m-l_i, n] = 0$ for $m < l_i$. At the transmitter, the OTFS frame consists of one pilot symbol, N_G guard symbols, and $MN - N_G - 1$ data symbols, as represented in Fig. 3.1 on the left. Multiple quantities of interest are defined in the following list:

- $X_p \in \mathbb{C}$: single pilot symbol per frame associated with $\mathrm{SNR_p} = |X_p|^2/\sigma_{\mathrm{w}}^2$.
- $X_d[m,n]$: data symbols with $\mathrm{SNR_d} = \mathbb{E}[|X_d|^2]/\sigma_{\mathrm{w}}^2$, located at position $[m,n]$ in the delay-Doppler information grid.
- $l_{\max}$ and $k_{\max}$: maximum delay and Doppler taps among the channel paths, respectively.

We assume that $l_{\max} \leq \frac{M-1}{2}$ and $k_{\max} \leq N/2$ in order to satisfy the crystallization condition. As shown in Fig. 3.1 on the left, the pilot and data symbols in the delay-Doppler grid at the transmitter side are arranged as:

$$\mathbf{X}_{\mathrm{DD}}[m,n] = \begin{cases} X_p & m = l_p, \, n = k_p, \\ 0 & l_p - l_{\max} \leq m \leq l_p + l_{\max}, \\ X_d[m,n] & \text{otherwise.} \end{cases}$$

The pilot symbol X_p is located at (l_p, k_p), where $l_p = M - 1 - l_{\max}$ and $k_p = \lfloor N/2 \rfloor$. The N_G guard symbols have the purpose of preventing interference between data symbols and

pilot at the receiver. However, the addition of $N_G = (2l_{\max}+1)N - 1$ guard symbols, together with the pilot contribution, result in a per-frame overhead equal to:

$$\frac{N_G + 1}{MN} = \frac{(2l_{\max}+1)}{M}.$$

Note that for the integer Doppler case, it would be sufficient to place only $2k_{\max}$ guard symbols around the pilot along the Doppler domain. This would allow data symbols to be inserted at the edges of $\mathbf{X}_{\mathrm{DD}}$, thereby reducing the overhead. However, such arrangement would prevent the use of block-wise LMMSE equalization, discussed in section 2.3.5, which is crucial for significantly lowering the receiver complexity. As shown in Fig. 3.1 (right),

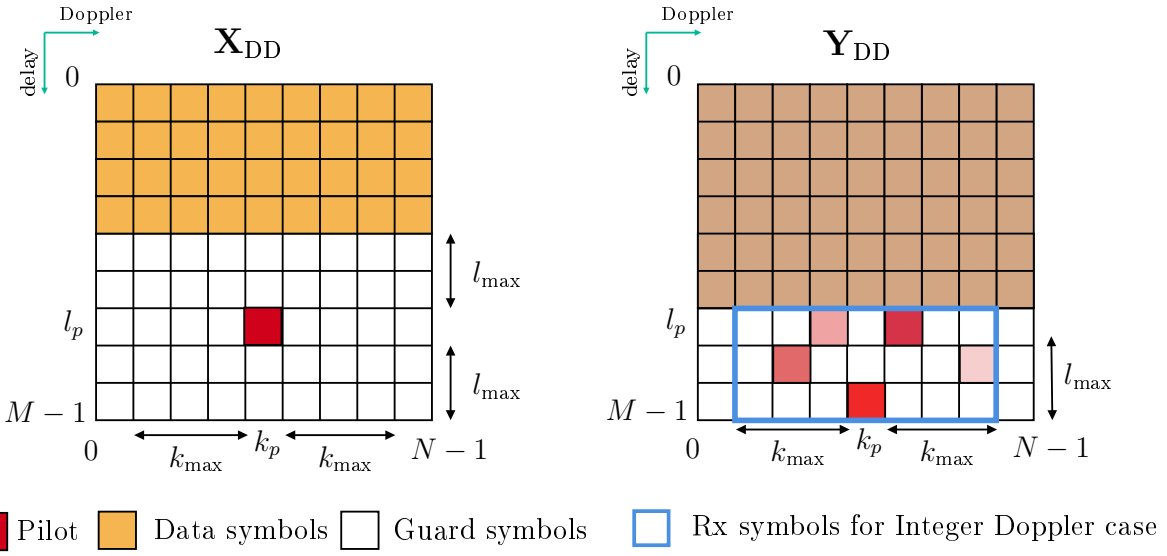


Figure 3.1: Tx symbol arrangement on the left and Rx symbol pattern on the right.

at the receiver side, by focusing only on the last $l_{\max}+1$ rows of the matrix $\mathbf{Y}_{\mathrm{DD}}$, the effect of the channel is just a number of shifted and attenuated replicas of X_p according to the delay and Doppler shifts. In Fig. 3.2 is illustrated an example of a transmitted OTFS frame on the left that includes only the pilot symbol with $\mathrm{SNR}_p = 40$ dB and a channel that introduces a single propagation path. In this case, the received pilot on the right appears as a single point within the delay-Doppler grid and is shifted from its original position according to the delay and Doppler channel characteristics. In order to estimate the channel, which is completely determined by the parameters $\{h_i, l_i, k_i\}$ for $i = 0, \ldots, P$ we consider only the received symbols $\mathbf{Y}_{\mathrm{DD}}[m, n]$, for $l_p \leq m \leq l_p + l_{\max}$ and $k_p - k_{\max} \leq n \leq k_p + k_{\max}$, as highlighted in Fig. 3.1 on the right. By replacing m and n by $l_p + l_i$ and $k_p + k_i$ in (3.1), the received pilot symbols for the i-th path can be written in terms of the transmitted pilot symbols as:

$$\mathbf{Y}_{\mathrm{DD}}[l_p + l_i,\, k_p + k_i] = h_i\, z^{k_i\, l_p}\, \mathbf{X}_{\mathrm{DD}}[l_p, k_p] = h_i\, z^{k_i\, l_p}\, X_p. \tag{3.2}$$

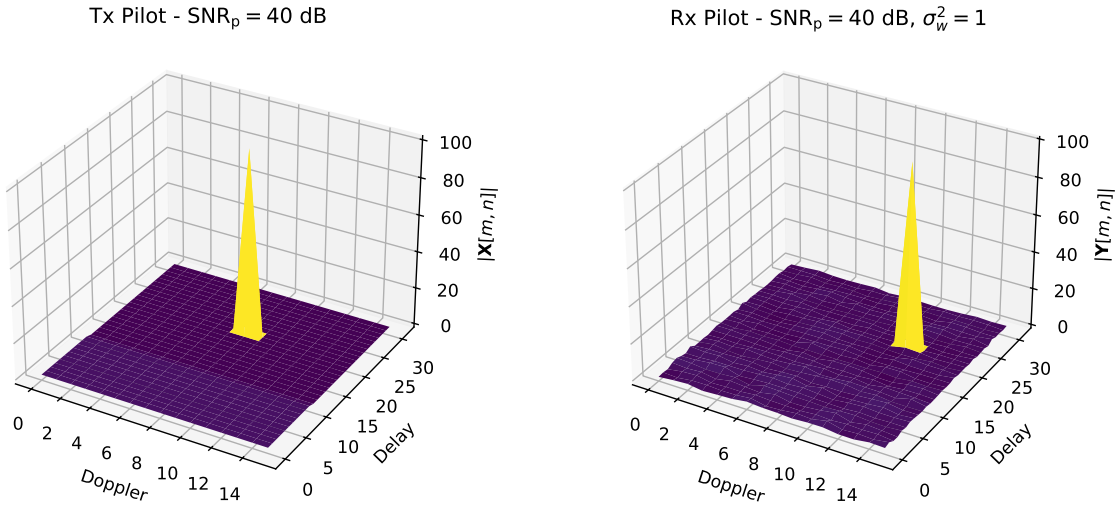


Figure 3.2: Transmitted and received pilot in delay-Doppler domain for the integer Doppler case.

Embedded-Pilot channel estimation using the threshold method

To avoid mistaking a channel path for additive noise, the transmitter must provide enough pilot power to achieve a high enough $\mathrm{SNR_p}$ at the receiver, enabling the pilot signal to be effectively discriminated from the noise floor. Here is presented a threshold-based channel estimation method for path detection [9]. Looking at Fig. 3.1 (right) and continuing from (3.2), for $0 \leq l_i \leq l_{\max}$ and $-k_{\max} \leq k_i \leq k_{\max}$, if $|\mathbf{Y}[l_p + l_i,\, k_p + k_i]| \geq \mathcal{T}$, where $\mathcal{T}$ represents the threshold, then a path with delay l_i and Doppler shift k_i is considered to be present. The channel gain relative to this component is estimated as:

$$\hat{h}[l_i, k_i] = \frac{\mathbf{Y}_{\mathrm{DD}}[l_p + l_i, k_p + k_i]}{X_p\, z^{k_i\, l_p}}.$$

The choice of the threshold $\mathcal{T}$ is crucial to retrieve a good channel estimate. If $\mathcal{T}$ is too low, the probability of mistaking a path for the noise floor increases, i.e. the probability of false alarm increases, and if it is too high the probability of miss detection rises. As a result, the error performance of data detection is affected by $\mathcal{T}$ and it is therefore important to find the best tradeoff. In this thesis, the threshold is defined as a function of the noise standard deviation via the formula $\mathcal{T} = \beta\, \sigma_{\mathrm{w}}$, where $\beta \in \mathbb{R}^+$.

3.2 The fractional Doppler case

We now turn to the more general and realistic scenario of fractional Doppler shifts. In this case, the previous assumption of "on-grid" Doppler shifts no longer holds. This is because the resolution along the Doppler axis, given by $1/NT$, is inversely proportional to the frame duration, and practical constraints prevent us from transmitting overly long signals. Therefore, the limited Doppler resolution does not allow to approximate the real Doppler values to the next integer on the delay-Doppler grid without resulting in significant error. In this scenario, we define $k_{\nu_i} \in \mathbb{Z}$ as the integer part of the normalized Doppler shift and $\kappa_{\nu_i} \in [0, 1)$ as the fractional one. Then, by performing the same manipulations as in section 2.3.5, substituting now (2.52) into (2.46), the input–output relation for fractional Doppler shifts is derived as follows:

$$\mathbf{Y}_{\mathrm{DD}}[m,n] = \sum_{i=1}^{P} h_i \, z^{(k_{\nu_i}+\kappa_{\nu_i})(m-l_i)} \left(\sum_{k=0}^{N-1} \zeta_N(k_{\nu_i} + \kappa_{\nu_i} - k) \, \mathbf{X}_{\mathrm{DD}} \left[m - l_i, \, [n-k]_N \right] \right) \tag{3.3}$$

recalling that $\zeta(\cdot)$ is the normalized periodic sinc function expressed in (2.53). The system setting considered at the transmitter is the same as for the integer case, as shown in Fig. 3.1 on the left. It can be observed from (3.3) that due to κ_{ν_i}, each $\mathbf{Y}_{\mathrm{DD}}[m,n]$ can result in the aggregation of $P \cdot N$ transmitted information symbols. This is because $\zeta_N(k+\kappa_{\nu_i})$ is localized at the integer k only in the absence of fractional Doppler ($\kappa_{\nu_i} = 0$). Instead, nonzero values of $0 \leq |\zeta_N(k + \kappa_{\nu_i})| < N$ appear according to κ_{ν_i} at any k when $\kappa_{\nu_i} \neq 0$. Thus, this is the source of the inter-Doppler interference (IDI) since each fractional Doppler path causes the data symbols to interfere with all symbols belonging to the same delay coordinate. This is determined by the convolution between the function $\zeta_N(\cdot)$ and the N symbols along the Doppler domain, as shown in (3.3). Fig. 3.3 shows a graphical example of what we just explained, where we consider the transmission of a frame with only one pilot with $\mathrm{SNR_p} = 40$ dB, and we assume that the channel introduces a single propagation path. Notice how the received pilot symbol, on the right, is spread across all the N Doppler shift indices (i.e., $k = 0, \ldots, N-1$) within a single delay bin.

As for the integer Doppler case (section 3.1), our focus is solely on the last $l_{\max} + 1$ rows of $\mathbf{Y}_{\mathrm{DD}}$, as shown in Fig. 3.4. Therefore, we substitute in (3.3) $m = l_p + l_i$ and $n = k_p + k$, for $l_i \in [0, l_{\max}]$ and $k \in [0, N-1]$. The received pilot symbols then can be expressed, for $i = 1, \ldots, P$, as:

$$\begin{aligned} \mathbf{Y}_{\mathrm{DD}}[l_p + l_i, \, k_p + k] &= h_i z^{(k_{\nu_i}+\kappa_{\nu_i})l_p} \, \zeta_N(k_{\nu_i} + \kappa_{\nu_i} - k) \, \mathbf{X}_{\mathrm{DD}}[l_p, k_p] \\ &= h_i z^{(k_{\nu_i}+\kappa_{\nu_i})l_p} \, \zeta_N(k_{\nu_i} + \kappa_{\nu_i} - k) \, X_p. \end{aligned} \tag{3.4}$$

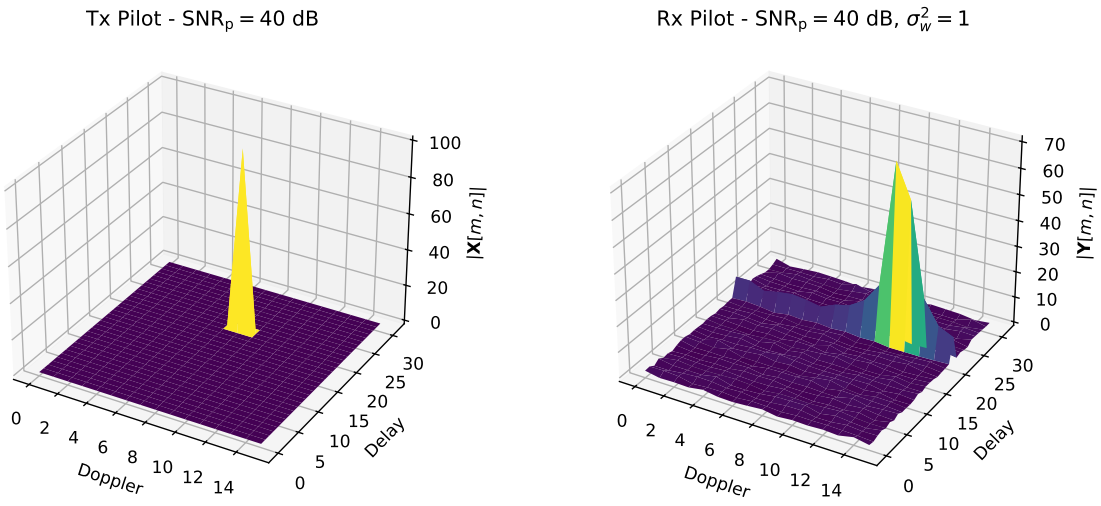


Figure 3.3: Transmitted and received pilot in the delay-Doppler domain for the fractional Doppler case.

Since the delay-Doppler grid is discrete, the channel can be observed only at integer multiples of the delay and Doppler resolutions, respectively given by $1/M\Delta f$ and $1/NT$. However, since the Doppler shift is fractional, this results in an "off-grid" problem. Consequently, the actual Doppler shifts $k_{\nu_i} + \kappa_{\nu_i}$ are unknown at the receiver, and what happens is that one single fractional path is observed as multiple paths with integer Doppler shifts, as illustrated in the right part of Fig. 3.3.

3.2.1 Fractional Doppler estimation with cross-correlation

To retrieve the complex channel gain h_i of a path, we first need to estimate the delay and Doppler taps, l_i and $k_{\nu_i}+\kappa_{\nu_i}$ respectively. To that end, in this section, we derive a strategy to estimate $k_{\nu_i} + \kappa_{\nu_i}$ based on the knowledge of the normalized sinc function; a similar approach has been proposed also in [13]. Specifically, we implement a channel estimation method in the delay-Doppler domain employing the cross-correlation function over Doppler domain elements. In particular, the normalized cross-correlation is performed for all k between $\mathbf{Y}_{\mathrm{DD}}[l_p + l_i, k_p + k]$, expressed in (3.4), and $\zeta_N(k + \kappa)$ with parameter κ corresponding to the fractional Doppler. This operation is iterated for all the $l_p + l_i$-th delay bins with $0 \leq l_i \leq l_{\max}$. The cross-correlation function is given by:

$$R_{Y_{l_p+l_i},\zeta}(k+\kappa) = \frac{1}{N^2} \sum_{k'=0}^{N-1} \mathbf{Y}_{\mathrm{DD}}[l_p + l_i, k_p + k'] \, \zeta_N^*([k' - (k+\kappa)]_N) \tag{3.5}$$

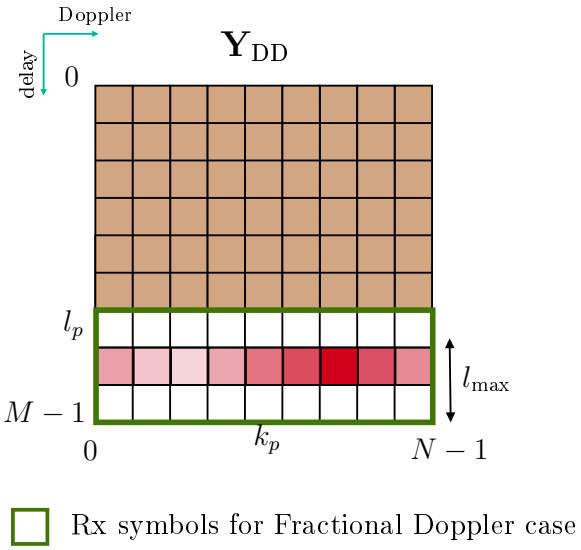


Figure 3.4: Rx symbol pattern.

and it measures the similarity between the received pilot signal and the known sinc function $\zeta_N(\cdot)$. More specifically, it searches for Doppler shifts, both integer and fractional, by sliding the reference function $\zeta_N(k+\kappa)$ across the Doppler domain. For each possible Doppler shift value $k+\kappa$, it computes a correlation value that reflects how well the received signal aligns with the reference. To accurately detect Doppler shifts that do not fall exactly on the integer delay-Doppler grid, we introduce a fractional Doppler resolution parameter D which defines the number of subdivisions within each integer Doppler bin. This allows to evaluate the cross-correlation for fractional values of κ with a step of $1/D$. For example, with $D = 10$, the search range includes Doppler shifts $\kappa = 0, 0.1, 0.2, \ldots, 0.9$. A larger value of D enables finer resolution in detecting fractional Doppler shifts, improving estimation accuracy, at the cost of increased computational complexity.

Threshold-based method

Similarly to the integer Doppler case, in order to distinguish the propagation paths from the noise floor, a threshold-based method is implemented. When the estimated Doppler index $\hat{k}+\hat{\kappa}$ matches the actual Doppler index $k_{\nu_i}+\kappa_{\nu_i}$ of the i-th path, the correlation is expected to reach the highest magnitude. More specifically, if $|R_{Y_{l_p+l_i},\zeta}(k+\kappa)| > \mathcal{T}$, we estimate the fractional Doppler shift for the l_p+l_i-th delay bin as follows:

$$\hat{k}_{\nu_i}+\hat{\kappa}_{\nu_i} = \hat{k}+\hat{\kappa}.$$

3.2.2 Channel gain estimation

For each channel path, once the delay and Doppler shifts are estimated, its complex gain $\hat{h}_i$ can be estimated too. In this section we discuss two equivalent techniques to estimate $\hat{h}_i$:

- By exploiting the cross-correlation peak, from the fractional Doppler estimation
- Through a weighted average approach

Correlation

The path gain $\hat{h}_i$ can be estimated from the peak of the cross-correlation $R_{Y_{l_p+l_i},\zeta}\left(\hat{k}_{\nu_i}+\hat{\kappa}_{\nu_i}\right)$ as:

$$\hat{h}_i = \frac{R_{Y_{l_p+l_i},\zeta}\left(\hat{k}_{\nu_i}+\hat{\kappa}_{\nu_i}\right)}{X_p\, z^{(\hat{k}_{\nu_i}+\hat{\kappa}_{\nu_i})l_p}}. \tag{3.6}$$

Weighted average

An equivalent method consists in inverting equation (3.4), as:

$$\bar{h}[l_i,k] = \frac{\mathbf{Y}_{\mathrm{DD}}[l_p+l_i,k_p+k]}{X_p\, z^{(k_{\nu_i}+\kappa_{\nu_i})\,l_p}\zeta_N(k_{\nu_i}+\kappa_{\nu_i}-k)}. \tag{3.7}$$

This gives us a vector of complex values $\bar{h}[l_i,k]$, $k=0,\ldots,N$, that are due to the i-th channel path. Due to the fractional Doppler effect, the channel's energy is spread across multiple Doppler bins following the shape of the sinc $\zeta_N(\cdot)$. This means that not all $\bar{h}[l_i,k]$ values contribute equally to the true path gain. The samples closer to the peak of the sinc carry more reliable information, while those farther away are more affected by the noise. In order to accurately estimate the gain $\hat{h}_i$, we compute a weighted average of the $\bar{h}[l_i,k]$ samples, giving more importance to the samples that lie closer to the sinc peak. This is achieved using the formula:

$$\hat{h}_i = \frac{\sum_{k=0}^{N-1} w_k x_k}{\sum_{n=0}^{N-1} w_k} \tag{3.8}$$

where $w_k = |\zeta_N(k_{\nu_i}+\kappa_{\nu_i}-k)|^2$ are the weights and $x_k = \bar{h}[l_i,k]$. By applying this weighted average, the influence of noisy samples is reduced, and a more accurate and robust estimate of the path gain $\hat{h}_i$ can be obtained.
For multipath scenarios, where different paths can exist in the same delay bin, the estimated path is removed from the pilot response $\mathbf{Y}_{\mathrm{DD}}[m,n]$, and then the estimation is iteratively executed until $|R_{Y',\zeta}(\hat{k}+\hat{\kappa})| < \mathcal{T}$, as explained in section 4.2.

4 Numerical Results

In this chapter, we present the numerical results obtained from Monte Carlo simulations of the OFDM and OTFS communication schemes, implemented in Python and MATLAB.

4.1 OFDM over different channels

First, we show results related to the OFDM system discussed in Chapter 2, simulating an AWGN channel. Then, we increase the channel complexity by introducing a time-invariant multipath channel affected by time dispersion, and finally, we include Doppler shifts. We evaluate the bit-error-rate (BER) over all the afore mentioned scenarios.

OFDM over the AWGN channel

Starting from the OFDM communication chain represented in Fig. 2.3, we evaluate the performance in terms of BER of the uncoded OFDM system over an AWGN channel adopting the system parameters shown in Table 4.1. The bits source block generates $2M$ i.i.d. bits and then the QPSK mapper performs Gray mapping between input bit pairs and constellation symbols, as illustrated in section 1.1.2. Each symbol encodes $L = 2$ bits and the resulting symbol rate is given by $B_s = B_r/(2\log_2 L) = B_r/2$, where B_r is the bit

Simulation parameters	
Bits per subcarrier	$L = 2$
Total subcarriers	$M = 1024$
Cyclic prefix length	$L_{CP} = 72$
Maximum number of errors	$N_{err} = 10^6$
Maximum number of iterations	$N_{iter} = 10^3$

Table 4.1: System parameters for OFDM in AWGN channel.

rate. The average symbol energy is normalized to 1. The noise power is given as:

$$P_{\mathrm{w}} = \sigma_{\mathrm{w}}^2 = \frac{1}{2\frac{E_b}{N_0}}$$

for chosen values of E_b/N_0. Since the CP has a power overhead, the noise power introduced by the channel must also be scaled accordingly. Specifically, this can be considered with the factor $\frac{1}{(1-L_{CP}/M)}$. The Monte Carlo simulations are performed for values of $E_b/N_0 \in [0, 1, 2, 3, \ldots, 10]$ dB. For each E_b/N_0 point the OFDM chain is iterated until either the number of iterations reaches N_{iter} or when the number of errors is equal or greater than N_{err}. The resulting BER, given by:

$$\mathrm{BER} = \frac{N_{err}}{2M \cdot N_{iter}}, \tag{4.1}$$

is then compared to the theoretical bit error probability P_b for QPSK modulation, given in (1.17), which after considering the CP overhead $\frac{\mathrm{M}}{\mathrm{M}+L_{CP}}$ becomes:

$$P_b^{\mathrm{OFDM}} = \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}\frac{\mathrm{M}}{\mathrm{M}+L_{CP}}}\right). \tag{4.2}$$

Fig. 4.1 illustrates the theoretical curves P_b for QPSK (in yellow) and P_b^{OFDM} for OFDM

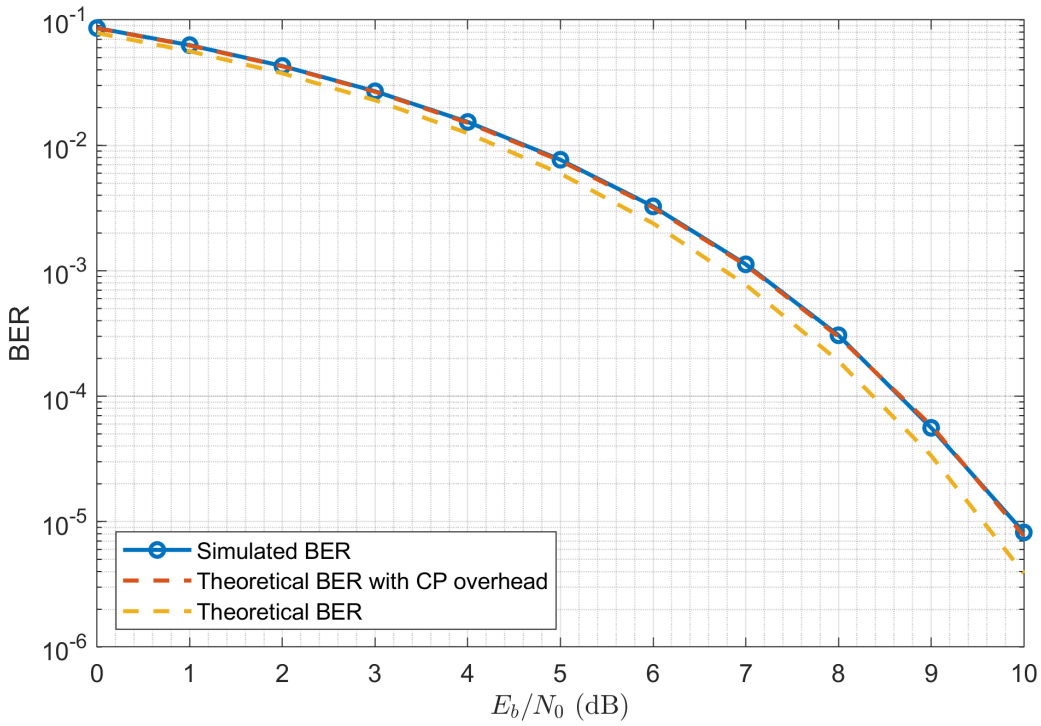


Figure 4.1: OFDM BER plots in AWGN channel with QPSK modulation.

(in red) in an AWGN channel, along with the BER (in blue) obtained via Monte Carlo simulations. As shown, the simulated BER closely matches the theoretical OFDM performance which accounts for the presence of the cyclic prefix.

OFDM over static multipath channels

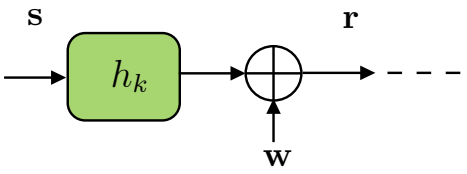


Figure 4.2: OFDM diagram block over static multipath channels.

In this section, we extend the basic OFDM model presented previously by introducing a multipath channel with delayed paths, represented in Fig. 4.2. Recalling the analysis in section 2.2.1, if a cyclic prefix of length $L_{CP} \geq l_{\max}$ is appended to the OFDM symbol, the received vector $\mathbf{r}$ is the result of the circular convolution between the channel impulse response $\mathbf{h}$ and the transmitted vector $\mathbf{s}$, as expressed in (2.8). In the following sections, we consider a single OFDM symbol (i.e. $N = 1$) and we support the discussion presented in sections 2.2.1 and 2.2.2 with plots and simulations, focusing on the impact of noise enhancement in frequency-domain equalization within an OFDM system. Specifically, the implemented equalization techniques rely on ideal CSI and two types of equalizers are considered: Zero forcing (ZF) and minimum mean-square error (MMSE), both at sub-channel level.

Zero Forcing (ZF)

The main objective of this approach is to correct the gain of each subcarrier by designing the equalizer such that the combined response of the channel and equalizer yields a flat frequency response. Under the theoretical assumption of an equalizer with an infinite number of taps, the ZF equalizer is defined as:

$$C[i] = \frac{1}{H[i]}$$

where $H[i]$ is the flat-fading channel gain associated with the ith subcarrier, expressed as:

$$H[i] = \mathrm{DFT}\{h[m]\} = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} h[m]\, e^{-j\frac{2\pi mi}{M}}, \quad 0 \leq i \leq M-1. \tag{4.3}$$

This leads to a combined system response $C[i]H[i] = 1$ which corresponds to a flat magnitude and linear phase response across all frequencies. Equalization in the frequency domain is performed by dividing the received signal $y_{\mathrm{FT}}[i]$ by the channel's frequency response $H[i]$, as shown in Fig. 4.3. The estimated received samples $\hat{x}_{\mathrm{FT}}[i]$ are recovered via the operation:

$$\hat{x}_{\mathrm{FT}}[i] = y_{\mathrm{FT}}[i]C[i] = x_{\mathrm{FT}}[i] + \frac{\bar{\mathrm{w}}[i]}{H[i]} \tag{4.4}$$

where $x_{\mathrm{FT}}[i]$ and $y_{\mathrm{FT}}[i]$ are the transmitted and received symbols and $\bar{\mathrm{w}}[i]$ is the additive noise, all in the frequency domain. While the ZF equalizer ideally recovers the transmitted signal, it is highly sensitive to deep fades in the channel frequency response. Specifically, if $H[i]$ approaches zero at certain frequencies, the inverse $1/H[i]$ becomes very large, significantly amplifying the noise component $\frac{\bar{\mathrm{w}}[i]}{H[i]}$. This phenomenon, known as noise enhancement, can result in severe performance degradation.

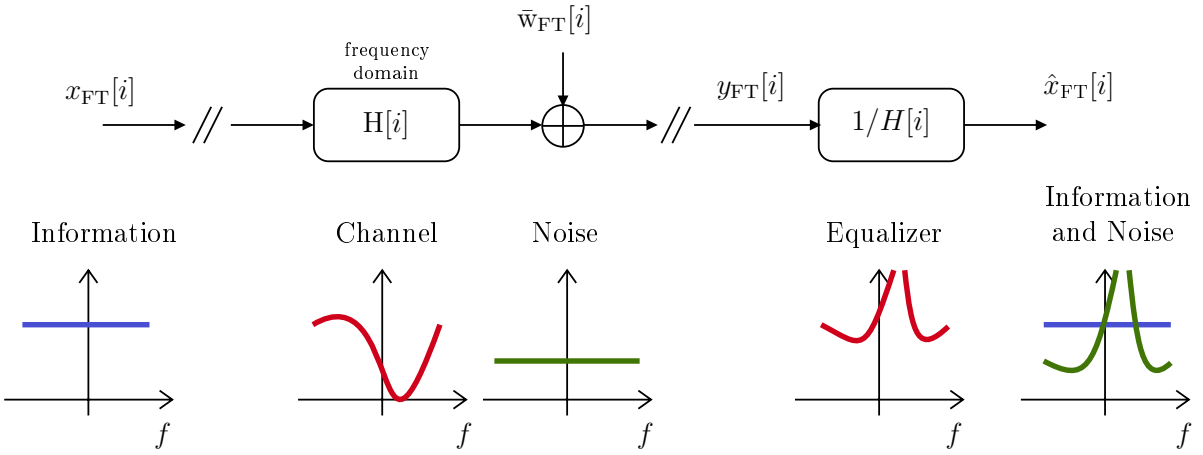


Figure 4.3: ZF diagram.

Minimum Mean Square Error (MMSE)

The MMSE criterium aims at correcting the channel gain minimizing the effect of thermal noise by computing the equalizer's coefficients that minimize the mean square error (MSE):

$$\mathrm{MSE} = \mathbb{E}\left[\left|\hat{\mathbf{x}}_{\mathrm{FT}} - \mathbf{x}_{\mathrm{FT}}\right|^2\right]$$

where $\hat{\mathbf{x}}_{\mathrm{FT}}$ are the estimated received symbols in output of the equalizer in the frequency domain, as illustrated in Fig. 4.4. The frequency domain MMSE equalizer is given by:

$$C[i] = \frac{H[i]^*}{|H[i]|^2 + \sigma_{\mathrm{w}}^2/\sigma_x^2}.$$

Here, $H[i]^*$ is the complex conjugate of the channel gain associated with the i-th sub-channel, σ_{w}^2 is the noise variance and σ_x^2 is the variance of the transmitted signal. The ratio $\sigma_{\mathrm{w}}^2/\sigma_x^2$ effectively incorporates the SNR into the equalization process. When a spectral null occurs in the channel frequency response, this formulation prevents the equalizer from excessively amplifying the noise, unlike the ZF approach. As a result, the MMSE equalizer provides a more robust solution, particularly in low SNR channels.

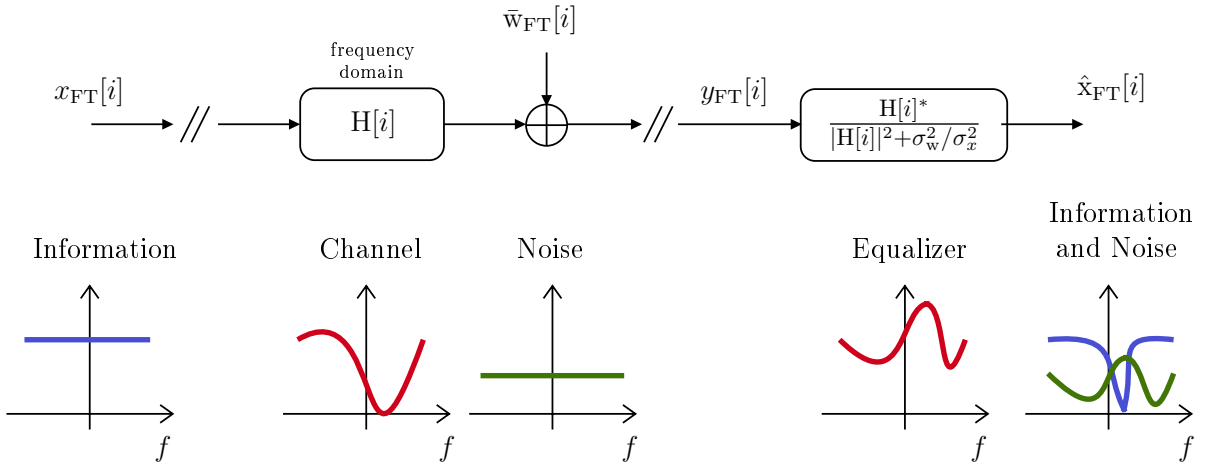


Figure 4.4: MMSE diagram.

Simulations

We consider the discrete-time impulse response $\mathbf{h}$ of the channel, whose coefficients model the multipath effects introduced by the delays and are generated as:

$$h_k = a^k \quad \text{for } k \in [0, \ldots, P-1]$$

where $a \in [0, 1)$ is the decay factor and P is the number of propagation paths. To maintain unitary channel gain, a normalized impulse response is used, i.e. $\frac{\mathbf{h}}{||\mathbf{h}||} = 1$ where $||\cdot||$ is the Euclidean norm. Moreover, to match the dimension of the transmitted vector $\mathbf{s} \in \mathbb{C}^{M\times 1}$, $M-P$ zero samples are appended to the impulse response. The corresponding frequency-domain ZF equalizer is given by:

$$C[i]^{\mathrm{ZF}} = \frac{1}{H[i]} = 1 - a\, e^{-j2\pi iT}$$

where T is the symbol time. The MMSE equalizer is:

$$C[i]^{\mathrm{MMSE}} = \frac{H[i]^*}{|H[i]|^2 + \sigma_{\mathrm{w}}^2/\sigma_x^2}.$$

In the context of the simulated OFDM system, the choice between ZF and MMSE equalization does not significantly impact the BER; as illustrated in Fig. 4.5, the yellow (for MMSE) and red (for ZF) curves overlap closely. This plot shows the simulation of the OFDM system over a channel that introduces only delayed paths, $P = 3$ in this case, with gains characterized by decay factor $a = 0.2$. With reference to the amplitude distortion, we already explained that the MMSE equalizer yields a less distorted output by limiting the amplification of the noise. However, from a phase distortion perspective, both equalizers behave equivalently. In fact, assuming that the channel response for a specific frequency is $H[i] = a + jb$, the ZF equalizer introduces a phase shift of $-\text{arctg}(\frac{b}{a})$. The MMSE equalizer, having a real-valued denominator, only affects the magnitude and retains the same phase, since $H[i]^* = a - jb$, so the phase is $\text{arctg}(\frac{-b}{a}) = -\text{arctg}(\frac{b}{a})$. As a result, both equalizers provide identical phase equalization, which explains the similar BER performance. Since QPSK demodulation relies solely on the sign of the real and imaginary components of the received symbols, amplitude differences introduced by the equalizers do not affect the decision process. As a result, it is not possible to observe significant differences in the BER performance between the two equalization strategies [18]. By comparing Figs. 4.5 and 4.6, it is possible to notice the effect of varying the parameter

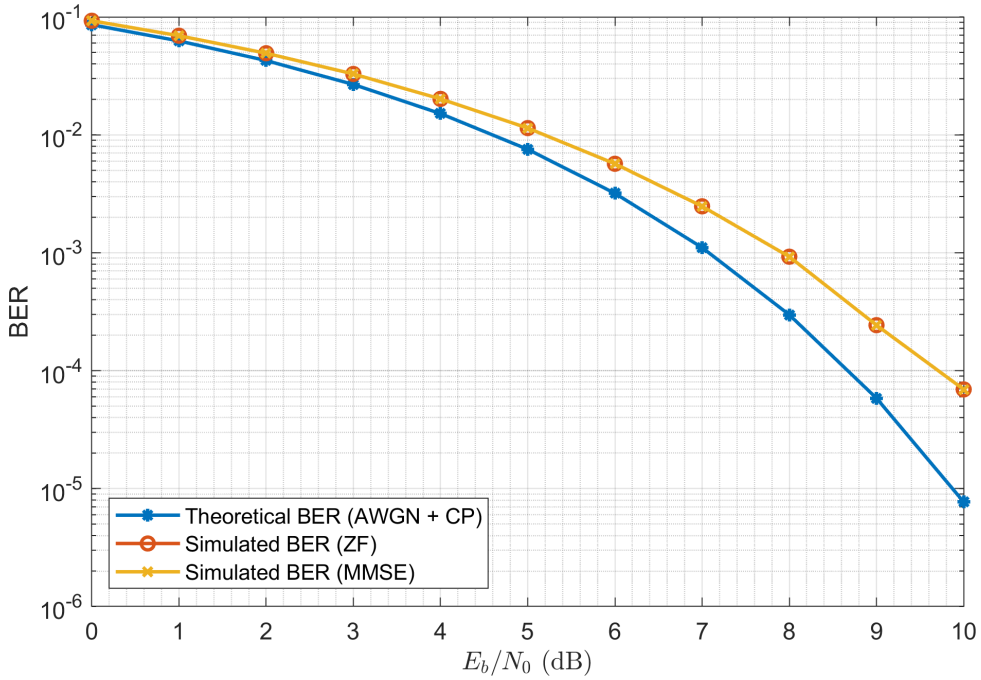


Figure 4.5: Performance comparison between ZF and MMSE equalizers in OFDM system with 3 delayed paths and decay factor $a = 0.2$.

a: as a increases (from 0.2 to 0.6), the energy contribution of the delayed paths becomes more significant, resulting in higher frequency selectivity and therefore higher BER. The P_b^{OFDM} curve (in blue) is included in the plots to qualitatively illustrate the performance

Simulation parameters	
Total subcarriers	$M = 32$
OFDM symbols	$N = 16$
Cyclic prefix length	$L_{CP} = 10$
Maximum delay tap	$l_{\max} = 10$
Maximum Doppler tap	$k_{\max} = 3$
Decay factor	$a = 0.6$
Maximum number of errors	$N_{err} = 10^6$
Maximum number of iterations	$N_{iter} = 10^3$

Table 4.2: System parameters for OFDM in multipath channel.

gap between the ideal AWGN scenario and the current case.

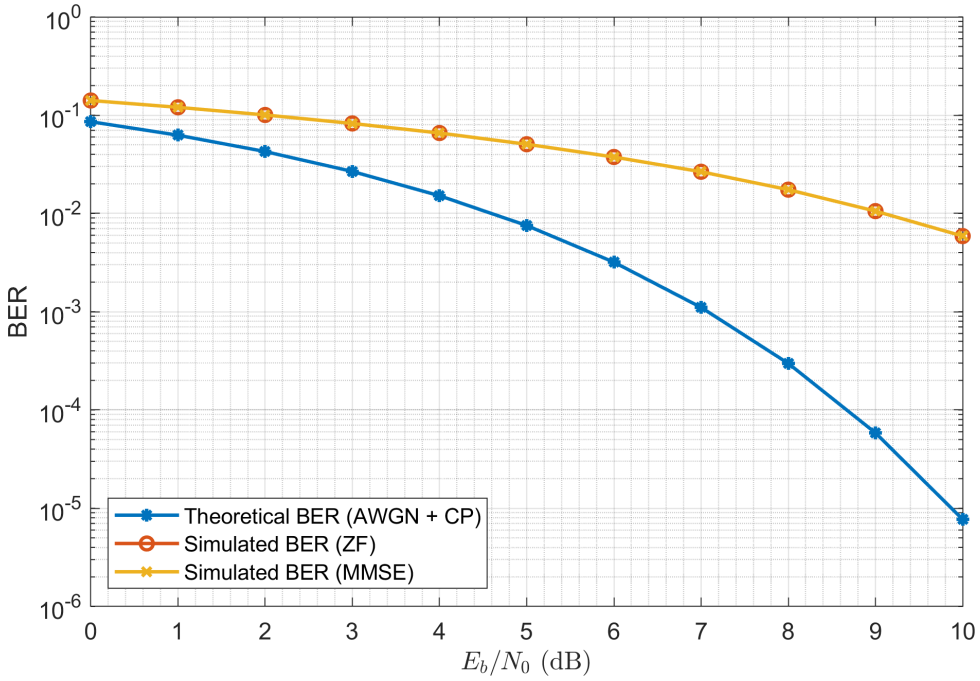


Figure 4.6: Performance comparison between ZF and MMSE equalizers in OFDM system with 3 delayed paths and decay factor $a = 0.6$.

OFDM over high-mobility channels

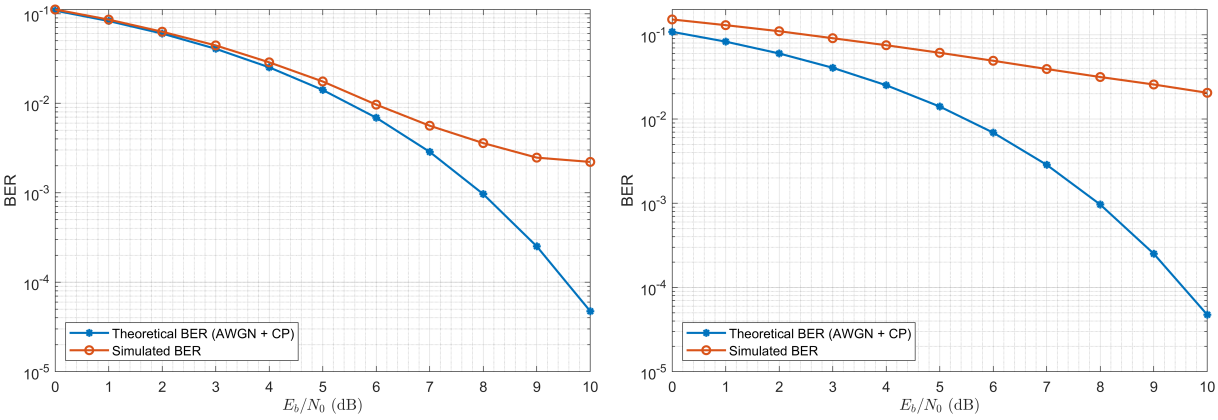
In this section, we extend the previous OFDM model by introducing integer Doppler shifts. The parameters adopted to simulate the OFDM system over multipath channels are listed in Table 4.2. Moreover, for each Monte Carlo iteration, channel parameters are generated as:

- Delays $l_i \sim \mathcal{U}_d[0, l_{\max}], l_i \in \mathbb{Z}$ and $i = 1, \ldots, P$
- Doppler shifts $k_i \sim \mathcal{U}_d[-k_{\max}, k_{\max}], k_i \in \mathbb{Z}$ and $i = 1, \ldots, P$
- Channel gains $h_i = \frac{a^i + ja^i}{\|a^i + ja^i\|}$ for $i = 0, \ldots, P-1$

where P is the number of paths simulated; these are generated so that they may share the same delay taps or Doppler shifts, but not both simultaneously. In order to simulate the OFDM system over high- mobility multipath channels we follow the analysis carried out in section 2.3.3 and we adapt it to fit the current case. In order to compare the OFDM and OTFS performances over high-mobility channels in the next section, at the receiver side, now the transmitted signal is recovered via the LMMSE time-domain equalization block by employing the formula expressed in (2.44) as:

$$\hat{\mathbf{s}} = \mathbf{G}_{cp}^H \left(\mathbf{G}_{cp} \mathbf{G}_{cp}^H + \sigma_{\mathrm{w}}^2 \, \mathbf{I}_{(M+L_{cp})N} \right)^{-1} \mathbf{r}'$$

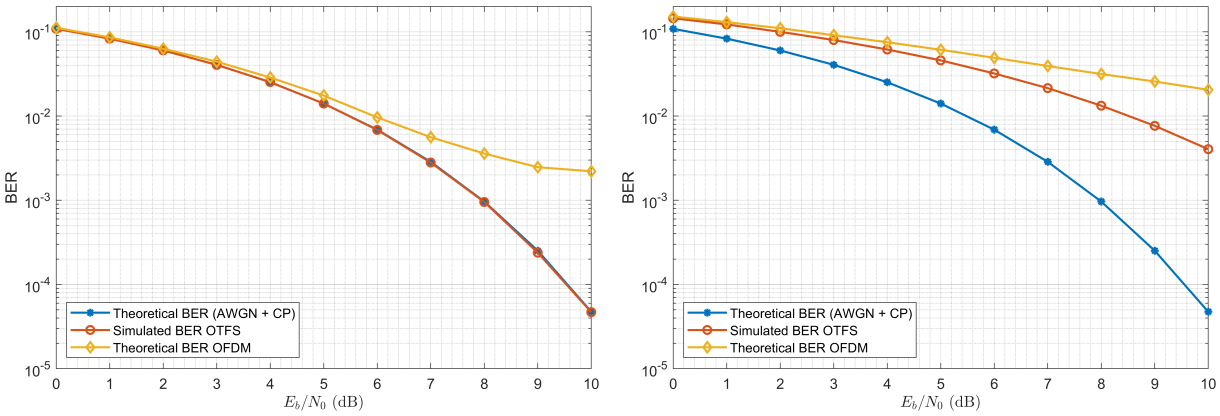
where $\mathbf{G}_{cp} \in \mathbb{C}^{N(M+L_{CP}) \times N(M+L_{CP})}$ represents the time-domain channel matrix and $\mathbf{r}' \in \mathbb{C}^{N(M+L_{CP}) \times 1}$ is the received signal, including the CP. Figs. 4.7a and 4.7b show the



(a) 1 path. (b) 3 paths.

Figure 4.7: OFDM BER plots in high-mobility multipath channels with $a = 0.6$.

effect of introducing Dopplers shifts, adopting $a = 0.6$. On the left, just one propagation path is present in the channel and even in this case we can notice that OFDM struggles for higher SNR, deviating from the theoretical P_b^{OFDM}. On the right, three propagation paths are simulated and it can be noticed that the simulated BER significantly diverges from P_b^{OFDM}. As anticipated in section 2.2.3, Doppler shifts make the channel matrix no longer circulant. As a result, ICI appears and the BER increases. In summary, since multiple Dopplers are hard to equalize, and subchannel gains are not equal and the lowest gain decides the performance, new modulation techniques are required to combat high Dopplers.



(a) 1 path. (b) 3 paths.

Figure 4.8: OTFS vs OFDM BER plots in multipath channels with delays and Doppler shifts, $a = 0.6$.

OFDM and OTFS systems over multipath channels

In the scenario of multipath channels, the pre-coded OFDM (i.e. OTFS) scheme, presented in Fig. 2.5 and discussed in section 2.3.1, is implemented and evaluated using the same parameters adopted in the Table 4.2. In order to compare the OTFS and OFDM performances, a cyclic prefix of length $L_{ZP} = L_{CP} = 10$ in the form of zero padding is appended to each OTFS block. Fig. 4.8a shows the BER performances of the two systems when simulating a channel with a single propagation path characterized by a delay l_i and a Doppler shift k_i, adopting again $a = 0.6$. In this scenario, it can be observed that the OTFS curve closely follows the theoretical bit error probability P_b^{OFDM} for AWGN channels, whereas the OFDM curve exhibits an error floor at higher SNR values. Fig. 4.8b compares the two systems in a scenario with three propagation paths. It can be observed that both curves deviate from the theoretical one; however, the OTFS system achieves significantly better performance, especially for increasing SNR.

4.2 OTFS channel estimation in delay-Doppler domain

In this section we show the performances of the delay-Doppler channel estimation techniques presented in Chapter 3 for the ZP-OTFS system discussed in 2.3.5. Fig. 4.9 shows the entire block diagram implemented in Python. At the transmitter side, $2M \cdot N$ i.i.d. bits are generated and then mapped into MN QPSK symbols. These together with the pilot are placed in the delay-Doppler matrix $\mathbf{X}_{\text{DD}} \in \mathbb{C}^{M \times N}$ according to the analysis carried out in Chapter 3. Then the IDZT operation, discussed in section 2.3.2, is implemented via M N-point IFFT and the P/S block. The resulting vector $\mathbf{s}$, containing NM

samples, is sent through the channel, which is represented by the time-domain matrix $\mathbf{G} \in \mathbb{C}^{NM \times NM}$. The received signal is expressed as in (2.35):

$$\mathbf{r} = \mathbf{G} \cdot \mathbf{s} + \mathbf{w}. \tag{4.5}$$

At the receiver side, the DZT block is applied, leading to the delay-Doppler matrix

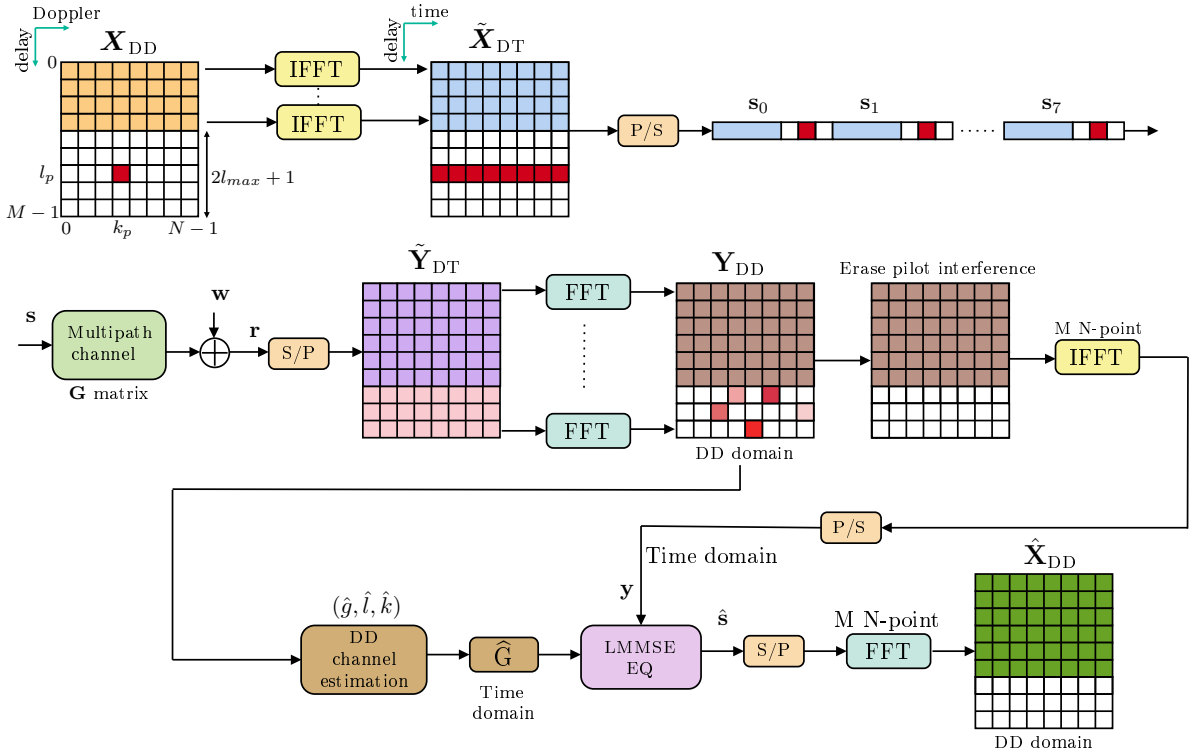


Figure 4.9: ZP-OTFS transceiver block diagram.

$\mathbf{Y}_{\mathrm{DD}} \in \mathbb{C}^{M \times N}$. This matrix is then used to perform channel estimation, as discussed in Chapter 3. Once the channel parameters $\{\hat{h}_i, \hat{l}_i, \hat{k}_{\nu_i} + \hat{\kappa}_{\nu_i}\}$ for each path are estimated, the matrix $\hat{\mathbf{G}}$ is reconstructed. In order to do time-domain equalization, the delay-Doppler matrix $\mathbf{Y}_{\mathrm{DD}}$ is first cleaned from the pilot interference by erasing the last $l_{\max} + 1$ rows and then converted to the time-domain by means of the IDZT, resulting in the vector $\mathbf{y}$. The block-wise LMMSE equalizer, described in section 2.3.5, recovers the time-domain samples. Once the time-domain samples are retrieved, another DZT is applied to obtain the QPSK symbols in the delay-Doppler domain. Now, a QPSK demapper produces the bits. As for the OFDM case, Monte Carlo simulations are conducted with parameters N_{err} and N_{iter}.

Case study

In order to evaluate the channel estimation performances, we compute the normalized minimum mean-square error (NMSE) defined as:

$$\text{NMSE} = \frac{||\mathbf{G} - \hat{\mathbf{G}}||^2}{||\mathbf{G}||^2}. \tag{4.6}$$

As discussed in Chapter 3, the channel estimation process is based solely on the received pilot signals and its error performance is strictly related to the choice of the threshold $\mathcal{T} = \beta\,\sigma_{\text{w}}$, which is function of the standard deviation of the noise. In general, the noise power can be estimated at the receiver side, here, without loss of generality, we assume $\sigma_{\text{w}}^2 = 1$ for all the simulations. We also recall that $\text{SNR}_{\text{p}} = |X_p|^2/\sigma_{\text{w}}^2$ and $\text{SNR}_{\text{d}} = \mathbb{E}[|X_d|^2]/\sigma_{\text{w}}^2$ represent the pilot and data SNRs, respectively. The adopted system parameters are listed in Table 4.3, while the channel parameters are generated as follows:

Simulation parameters	
Delay dimension	$M = 32$
Doppler dimension	$N = 16$
Maximum delay tap	$l_{\max} = 10$
Maximum Doppler tap	$k_{\max} = 7$
Pilot position along delay	$l_p = 21$
Pilot position along Doppler	$k_p = 8$
Decay factor	$a = 0.6$
Fractional Doppler resolution	$D = 100$

Table 4.3: System parameters for ZP-OTFS in multipath channel.

- Delays $l_i \sim \mathcal{U}_d[0, l_{\max}], l_i \in \mathbb{Z}$ for $i = 0, \ldots, P-1$
- Channel gains $h_i = \frac{a^i + ja^i}{\|a^i + ja^i\|}$ for $i = 0, \ldots, P-1$

where P is the actual number of paths simulated. These are generated so that they may share the same delay taps or Doppler shifts, but not both simultaneously.

Integer Doppler shifts

For the integer Doppler case, the Doppler shifts are generated as:

$$k_i \sim \mathcal{U}_d[-k_{\max}, k_{\max}], k_i \in \mathbb{Z} \quad \text{for} \quad i = 0, \ldots, P-1.$$

Fig. 4.10 on the left shows the NMSE curves as a function of the threshold $\mathcal{T} = \beta\sigma_{\mathrm{w}}$, for different values of P, with fixed $\mathrm{SNR_p} = 40$ dB. As illustrated, the overall trend shows that for low thresholds there is an increasing probability of false alarms, resulting in $\hat{P} > P$, where $\hat{P}$ is the average number of estimated paths represented by the numbers on the curves. Whereas, for high thresholds, the miss detection probability rises, leading to $\hat{P} < P$; this can be seen especially for $P = 5, 8$. For a certain interval of $\mathcal{T}$, the number of estimated paths $\hat{P}$ coincides with the true value P. The only exception is for $P = 8$, where paths with very low gain are always indistinguishable from noise and are thus not detected by the channel estimation algorithm. Therefore, in this regime, $\hat{P}$ is always lower than P. In general, the optimal threshold depends on the channel parameters, for example for $P \in [1, 3, 5]$ the threshold can be chosen in the range $[3.5\sigma_{\mathrm{w}}, \ldots, 8\sigma_{\mathrm{w}}]$ obtaining roughly the same NMSE; whereas for $P = 8$, a better choice for the threshold would be around $2.5\sigma_{\mathrm{w}}$ or $3\sigma_{\mathrm{w}}$. Moreover, as P increases, the NMSE curves shift upward. This is due to the total power being spread across more components, resulting in lower individual path gains. As a result, the paths become more susceptible to noise and the channel gains are estimated with less accuracy. Fig. 4.10 on the right reports the NMSE as function of the

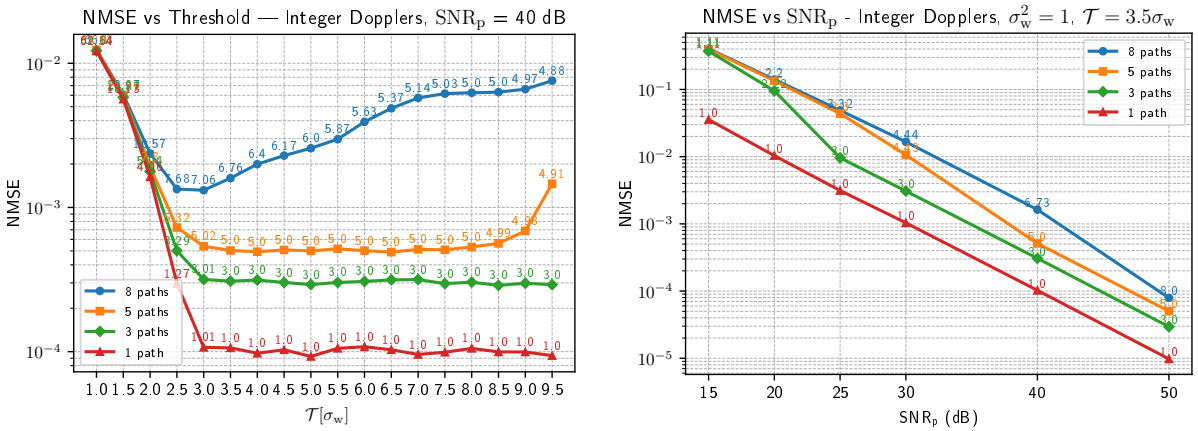


Figure 4.10: Channel estimation performances in terms of NMSE as a function of $\mathcal{T}$ (left) and of $\mathrm{SNR_p}$ (right), varying the number of paths $P \in [1, 3, 5, 8]$. The numbers on the curves represent the average number of estimated propagation paths $\hat{P}$ for each $\mathcal{T}$.

$\mathrm{SNR_p} \in [15, 20, 25, 30, 40, 50]$ dB, for $P = 3$ paths. In this case, leveraging the results from Fig. 4.10 (left), the threshold is fixed to $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$. The plot shows that, in general, increasing the pilot signal power relative to noise, leads to higher accuracy in the channel gain estimation and therefore in the matrix $\mathbf{G}$. This is also confirmed by the fact that, as $\mathrm{SNR_p}$ rises, the average number of estimated paths $\hat{P}$ approaches the actual number P.

Fractional Doppler shifts

The fractional Doppler shifts are generated as:

$$k_{\nu_i} + \kappa_{\nu_i},\ k_{\nu_i} \sim \mathcal{U}_d[-k_{\max}, k_{\max}],\ k_{\nu_i} \in \mathbb{Z}, \kappa_{\nu_i} \in [0,1) \quad \text{for} \quad i = 0, \ldots, P-1.$$

As previously discussed in Section 3.2, a key parameter that influences channel estimation is the fractional Doppler resolution, denoted by D. This parameter defines the number of subdivisions within each integer Doppler bin, allowing the cross-correlation to be evaluated for fractional Doppler values with a step size of $1/D$. Fig. 4.11 (left) illustrates an example

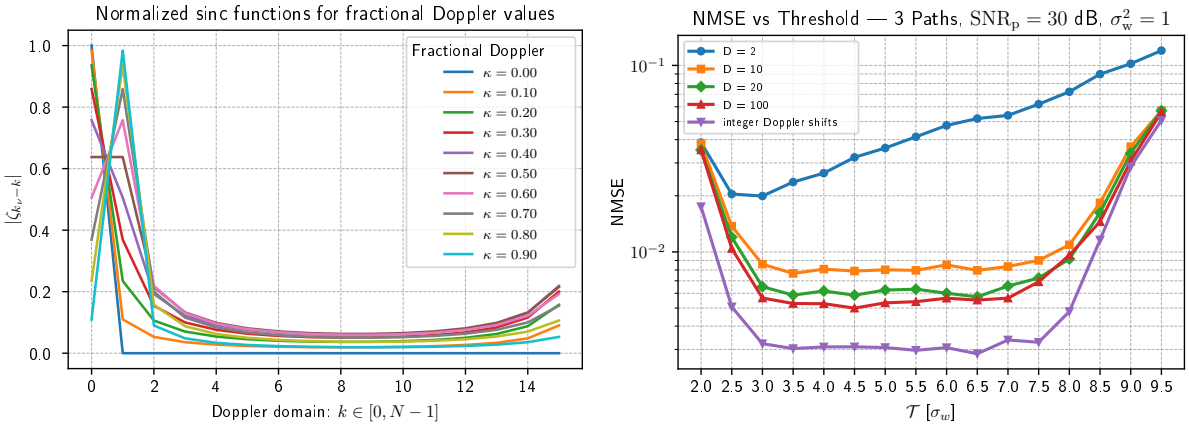


Figure 4.11: Normalized sinc functions (left) for fractional Doppler values with $D = 10$; Channel estimation performances in terms of NMSE (right) as a function of the threshold $\mathcal{T}$, varying the maximum delay tap $l_{\max}$, for $\mathrm{SNR_p} = 30$ dB.

of the sinc functions used for comparison with the received pilot signal when $D = 10$. Fig. 4.11 (right) shows the impact of the fractional Doppler resolution D on the estimation of Doppler shifts and, consequently, on the NMSE. A higher value of D provides finer resolution in detecting fractional Doppler components and improves estimation accuracy, approaching to the integer Doppler curve shown in purple. However, by adopting a larger D, the computational complexity increases.

Another fundamental issue associated with fractional Doppler shifts arises when two or more paths share the same delay l_i. In such cases, the sinc-shaped received pilot signals overlap and interfere with each other along the Doppler domain, as discussed in 3.2. In this scenario, the channel estimation procedure is performed iteratively by removing the contribution of each estimated path from the pilot response $\mathbf{Y}_{l_p+l_i}$. This process continues until the correlation peak falls below the threshold, i.e. when $|R_{Y',\zeta}(\hat{k}+\hat{\kappa})| < \mathcal{T}$ where Y' is the updated pilot response. Assuming the case of two propagation paths that share the same delay tap l_i but random Doppler shifts, specifically 6.11 and -2.45, Fig. 4.12 (top left) displays the cross-correlation values $|R_{Y_{l_p+l_i},\zeta}|$, highlighting the two

peaks corresponding to the two paths. Fig. 4.12 (top right) shows how the estimated pilot response closely matches the received signal with reference to the first path at Doppler bin $k_p + \lfloor 6.11 \rfloor$. After removing the estimated component from $Y_{l_p+l_i}$, Fig. 4.12 (bottom) illustrates how the corresponding peak disappears, effectively eliminating its interference and allowing the detection of the second path. However, this represent the relative simple case where only two paths share the same delay. Fig. 4.13 illustrates the NMSE as a function of the threshold $\mathcal{T}$ for different values of the maximum delay tap $l_{\max}$. In this scenario, we consider a pilot signal with $\mathrm{SNR_p} = 40$ dB, $P = 3$ propagation paths, and a fractional Doppler resolution of $D = 100$. The figure clearly shows the influence of $l_{\max}$ on channel estimation performance. Since the path delays are generated as a function of $l_{\max}$, smaller values of it increase the likelihood that multiple paths share the same delay l_i, thereby causing stronger IDI and leading to higher errors in the estimation of the channel gain.

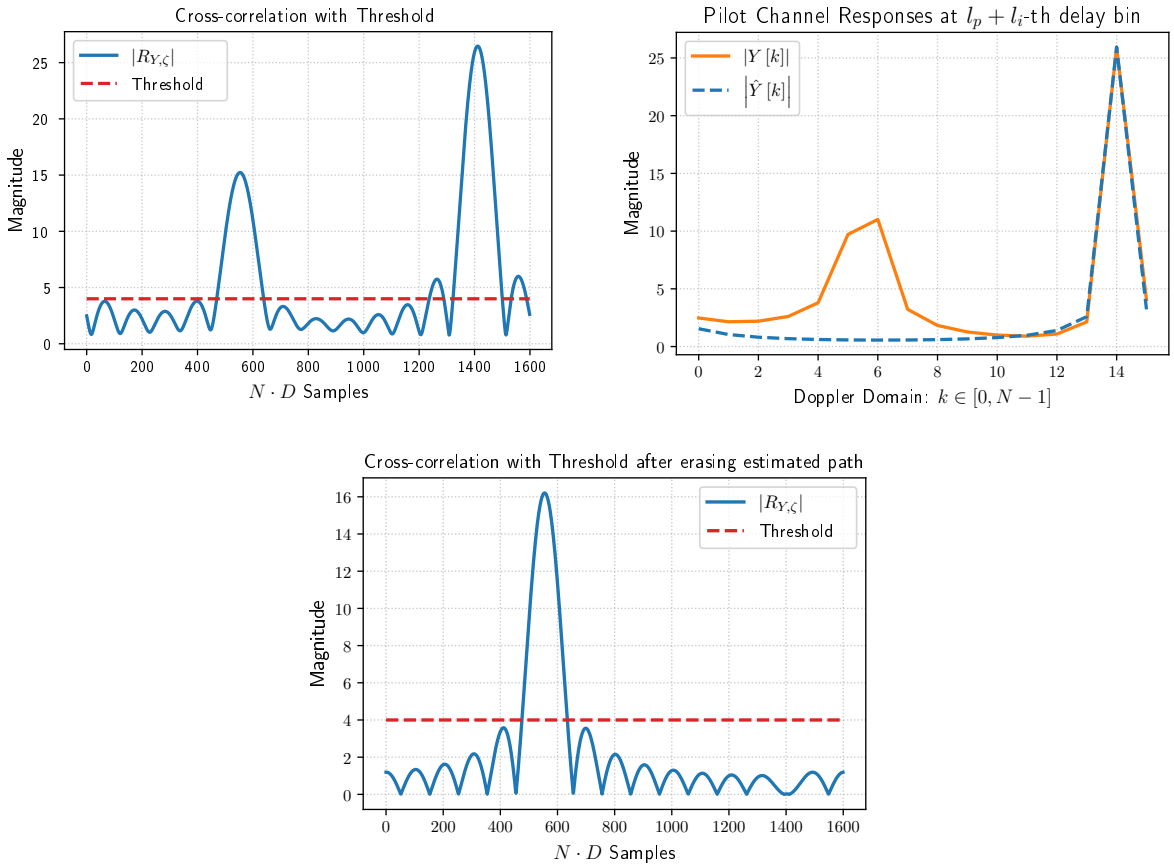


Figure 4.12: Iterative channel estimation: paths $P = 2$ having same delay tap.

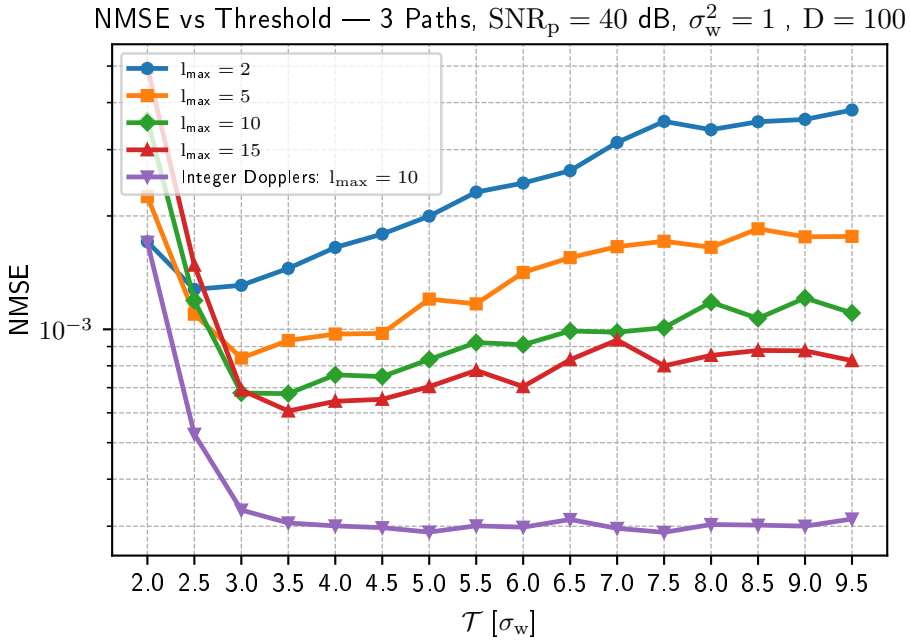


Figure 4.13: Channel estimation performances in terms of NMSE as a function of $\mathcal{T}$, for different values of the maximum delay tap $l_{\max}$, $D = 100$, $P = 3$ and $\mathrm{SNR_p} = 40$ dB.

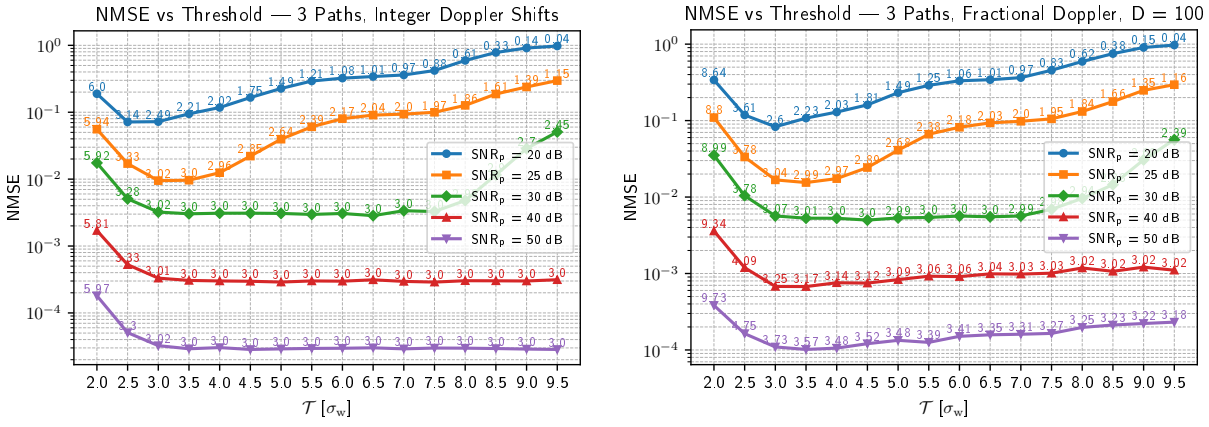


Figure 4.14: NMSE comparison between the integer and fractional Doppler cases as a function of $\mathcal{T}$, with $P = 3$ paths and for different values of $\mathrm{SNR_p}$.

Integer vs Fractional Doppler shifts

We now compare the channel estimation performance between the integer and fractional Doppler scenarios in terms of NMSE and BER, adopting again the parameters presented in Table 4.3. Figure 4.14 shows the NMSE curves as a function of the threshold $\mathcal{T}$ for various values of $\mathrm{SNR_p}$. In this setup, three propagation paths are considered, and for the fractional Doppler case, a resolution of $D = 100$ is chosen. The plots reveal that the integer Doppler case consistently outperforms the fractional one across all $\mathrm{SNR_p}$ values. However, the performance gap is not substantial, as the fractional NMSE curves remain relatively close to their integer counterparts. Fig. 4.15 confirms this result showing that, for a fixed threshold value $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$, the two performances sightly diverge as $\mathrm{SNR_p}$ increases. The final BER is computed as:

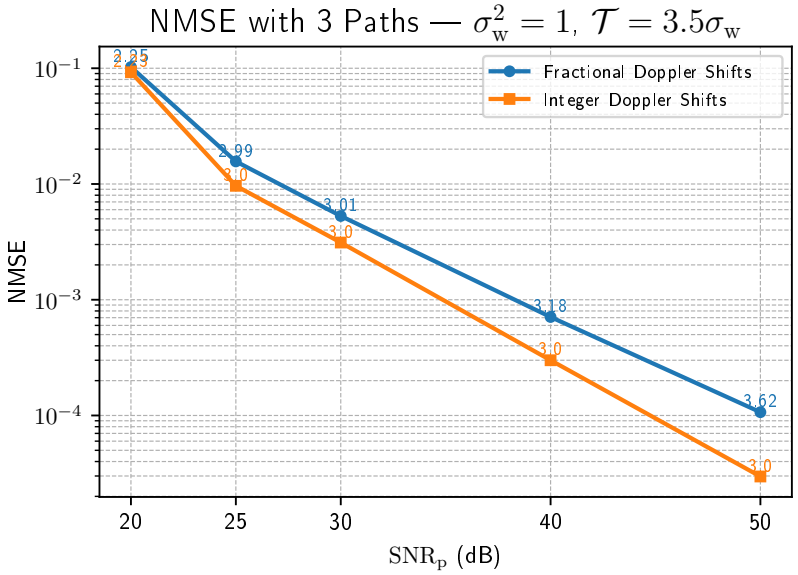


Figure 4.15: NMSE comparison between the integer and fractional Doppler cases as a function of $\mathrm{SNR_p}$, with 3 paths and for $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$.

$$\mathrm{BER} = \frac{N_{err}}{(2M - 2(2l_{\max} + 1))N \cdot N_{iter}}. \tag{4.7}$$

Fig. 4.16 presents the BER performance for the two cases as a function of $\mathrm{SNR_d}$, for different values of $\mathrm{SNR_p}$, with a fixed threshold $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$ and $P = 3$. First, we can observe that for $\mathrm{SNR_p} = 20$ dB, both channel estimation methods diverge significantly from the ideal curve (obtained with CSI) as $\mathrm{SNR_d}$ increases. This occurs because the pilot power is not high enough relative to the noise level, making it difficult to detect and distinguish the propagation paths. In general, the results clearly show that the performance of the two channel estimation methods differs only marginally, yielding nearly identical BER values. This behavior is likely due to the QPSK modulation scheme used

in the OTFS system, which is robust against amplitude distortions. If a higher-order modulation scheme was employed, the difference in error performance would likely be more pronounced.

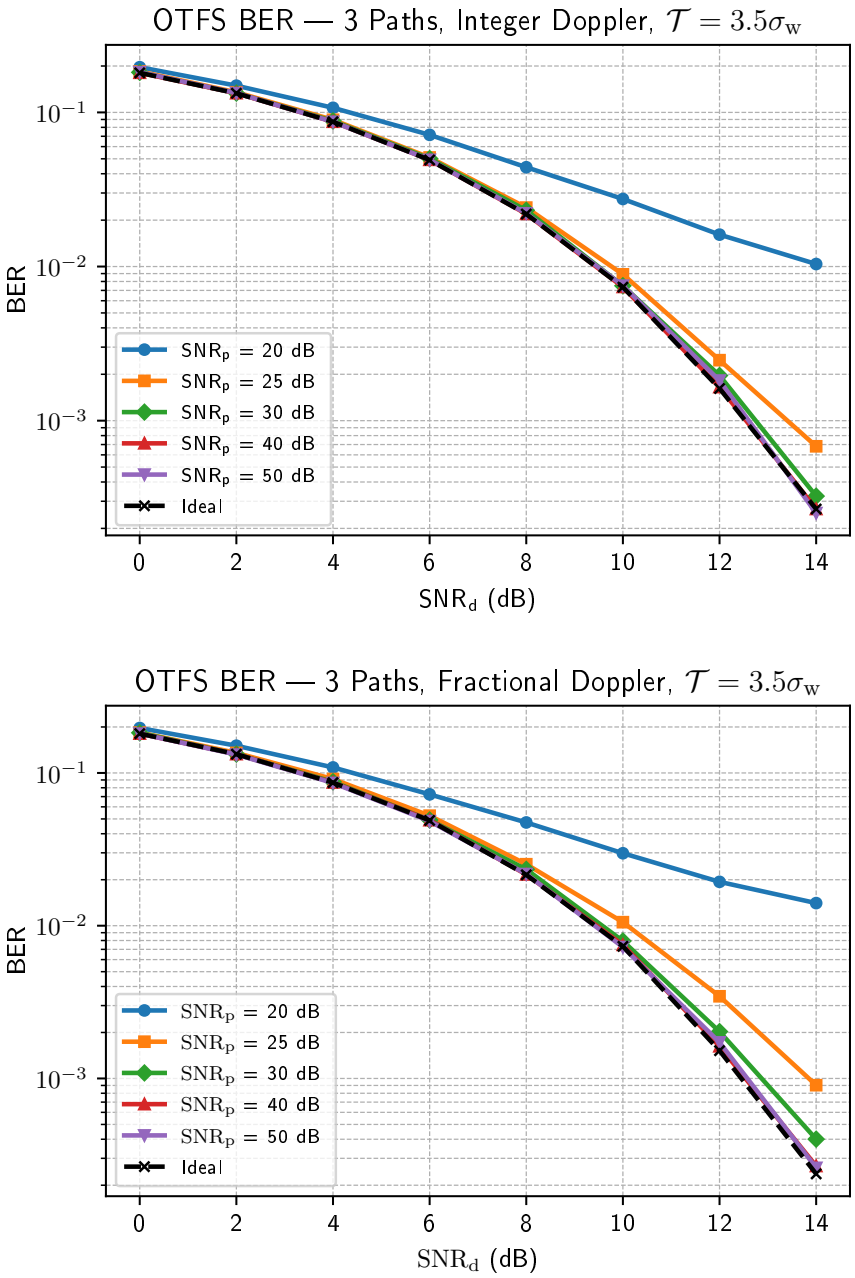


Figure 4.16: OTFS BER plots for the integer and fractional Doppler cases as a function of $\mathrm{SNR_d}$, with fixed threshold $\mathcal{T} = 3.5\sigma_w$, $P = 3$ and for different values of $\mathrm{SNR_p}$.

5 Software Defined Radio (SDR)

In this chapter we describe the implementation of ZP-OTFS on SDR, with reference to the block diagram presented in Fig. 4.9. SDR is a radio communication system where components that are typically implemented in hardware (filters, modulators/demodulators, detectors, etc.) are instead implemented in software on a general-purpose computing platform. The main components are:

- Radio frequency (RF) front end: converts incoming radio waves to baseband signals and viceversa.
- analog-to-digital converter (ADC) and digital-to-analog converter (DAC): convert analog signals to digital samples and viceversa, respectively.
- Computing platform: where software performs modulation/demodulation, filtering, detection, etc.

We use GNU Radio as the software development framework, as it provides signal processing functions for implementing SDRs. The framework offers both graphical flowgraph design in addition to development using Python and C++.

5.1 Flowgraph implementation

In this section we present the corresponding GNU Radio flowgraph implementation of the system depicted in Fig. 4.9, using the personal computer (PC) as the computing platform. To ensure a fair comparison, we adopt the same system parameters as those in Table 4.3. Moreover, to facilitate debugging and allow full customization of the signal processing chain, we developed most of the embedded Python blocks shown in Figs. 5.1 and 5.2. First, random bits are generated by the *Random Source* block (highlighted in light blue). These bits are then mapped using QPSK modulation, performed by the blocks inside in the magenta box. Next, the symbols are placed in the delay-Doppler matrix. Specifically, only the first $(M - 2l_{\max} - 1) \cdot N$ symbols are taken and the remaining positions are filled with the N_G guard symbols together with the pilot X_p in position (l_p, k_p), forming a $M \times N$

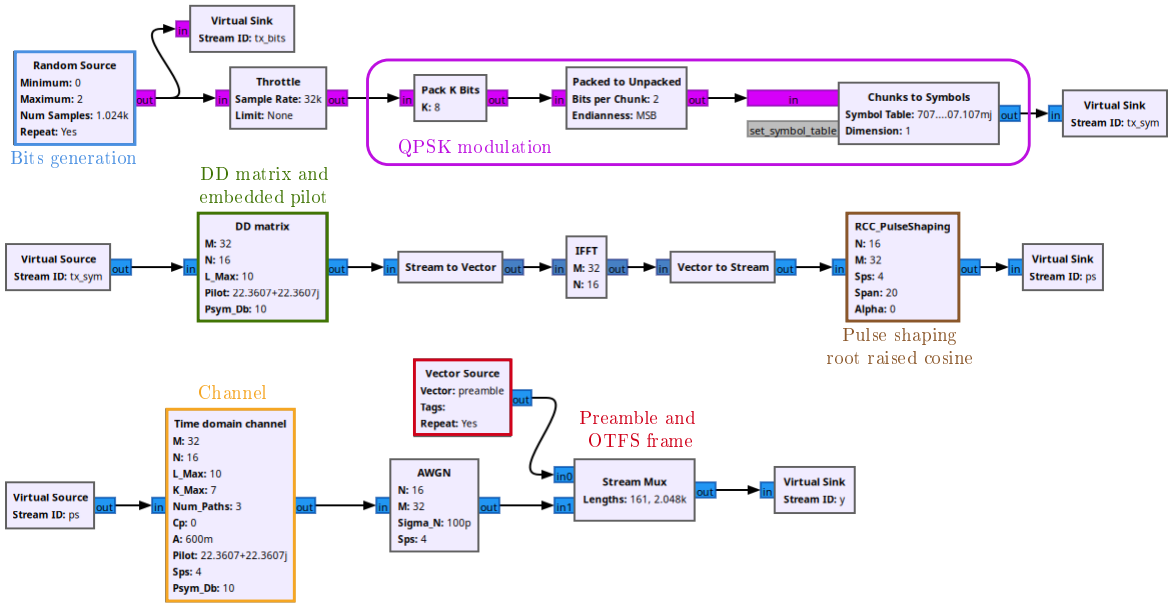


Figure 5.1: ZP-OTFS flowgraph, transmitter side.

matrix. This is done by the embedded python block *DD matrix* (in green). The next step consists in performing the IDZT, as discussed in section 2.3.2. This involves applying M N-point IFFT, followed by the *Stream to Vector* block. After the OTFS modulation, to emulate a more realistic system, we apply the *RRC Pulse Shaping*. This embedded Python block first oversamples the OTFS frame by a factor of $sps = 4$, then convolves it with the root raised cosine (RRC) pulse characterized by roll-off factor $alpha = 0$, i.e. resulting in a sinc pulse in time-domain, and pulse length $span = 5 \cdot sps = 20$. After the pulse shaping, the stream is sent to the *Time domain channel* block (in yellow). Here, the channel parameters $\{h_i, l_i, k_{\nu_i} + \kappa_{\nu_i}\}$ are generated for $i = 0, \dots, P-1$ as discussed in section 2.3.5 and the time-domain channel matrix $\mathbf{G}$ in constructed. $\mathbf{G}$ is also oversampled with factor sps, resulting in $\mathbf{G}_{up} \in \mathbb{C}^{MNsps \times MNsps}$, so that it can be multiplied by the pulse shaped OTFS frame. Subsequently, the *AWGN* block adds Gaussian noise with variance $\sigma_w^2 = 1$, consistent with the setup described in section 4.2. Finally, in order to simulate the realistic scenario where the receiver does not know the exact starting point of the OTFS frame (also due to the fact that the SDR devices introduce a non-deterministic delay between the transmission and reception time), a preamble is prepended to the transmitted signal. Specifically, the pulse-shaped Barker sequence of original length 13 is used to ensure good synchronization at the receiver. At the receiver side, as shown in Fig. 5.2, a *Delay* of 500 samples is used to emulate the real-world conditions just explained. The *Correlation Estimator* then performs the cross-correlation between the received signal and the known pulse-shaped Barker preamble. When the correlation peak associated with the preamble is detected, this block inserts some tags into the received

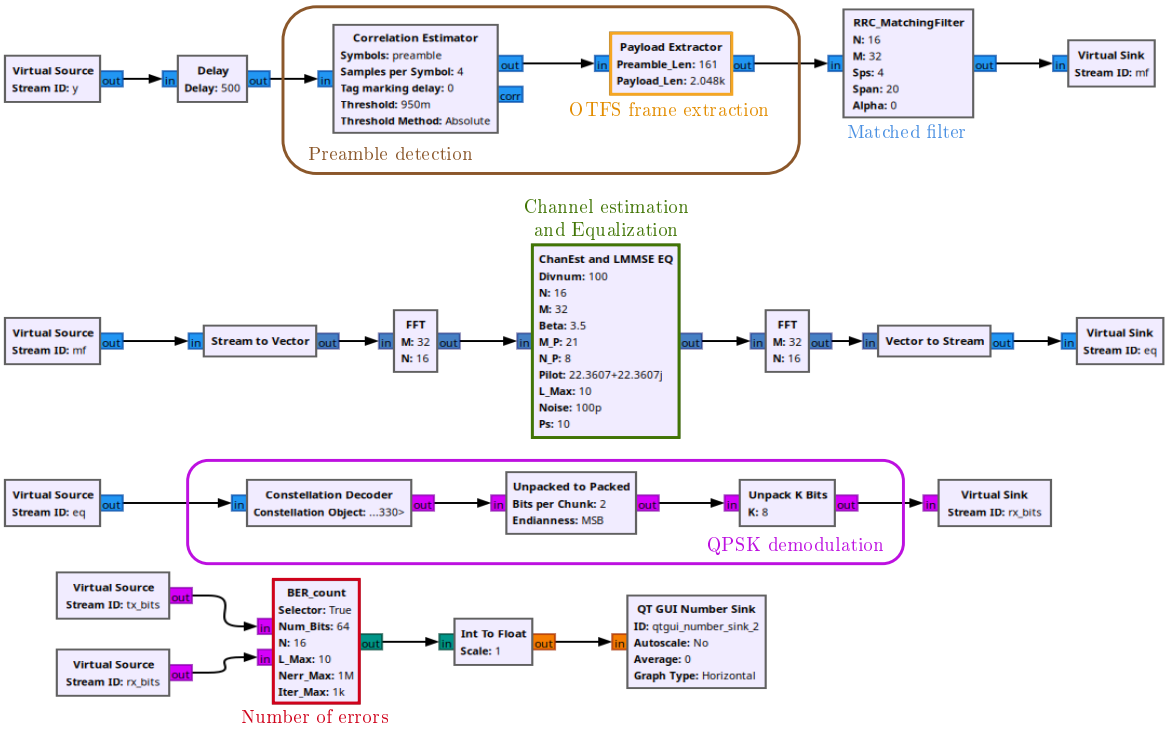


Figure 5.2: ZP-OTFS flowgraph receiver side.

stream, marking the estimated starting point of the received preamble. These tags are then used by the *Payload Extractor* Python block (in yellow) to isolate the useful OTFS frame from the incoming stream and discard the preamble. Next, the *RRC Matched Filter* block first convolves the extracted pulse-shaped OTFS frame with the RRC filter taps and then downsamples the resulting stream by a factor of *sps*. The signal is then transformed to the delay-Doppler domain using the DZT, implemented via the *Stream to Vector* block followed by M N-point IFFT. At this point, the *ChanEst and LMMSE EQ* block (in green) performs channel estimation in the delay-Doppler domain, as discussed in section 3.2. Based on the estimated channel parameters $\{\hat{h}_i, \hat{l}_i, \hat{k}_{\nu_i} + \hat{\kappa}_{\nu_i}\}$, the time-domain channel matrix $\hat{\mathbf{G}}$ is constructed. Now, the pilot interference is erased from the delay-Doppler matrix and another IDZT is applied to convert the signal back to the time-domain. Subsequently, the time-domain blockwise LMMSE equalization is carried out, as explained in section 4.2. Finally, the stream is transformed once more to the delay-Doppler domain via the DZT, and the QPSK demodulation is carried out via the blocks grouped in the magenta box, allowing the retrieval of the bits. At this point, the *BER count* block (in red) compares the originally transmitted useful bits with the decoded bits and counts the number of errors. This entire flowgraph is simulated N_{iter} times and the total number of bit errors N_{err} is memorized. Moreover, the simulations are conducted for different values of $\mathrm{SNR_d} \in [0, 2, 4, 6, \ldots, 14]$ dB, by changing the power of the transmitted

symbols P_s. The N_{iter} and N_{err} values corresponding to each $\mathrm{SNR_d}$ are then used for the post-processing computation of the BER, as defined in (4.7). Fig. 5.3 compares the performance of the standard Python implementation described in section 4.2 with that of the GNU Radio flowgraph, in terms of BER as a function of $\mathrm{SNR_d}$. This figure clearly shows that, for $\mathrm{SNR_p} = 30$ dB, threshold $\mathcal{T} = 3.5\sigma_\mathrm{w}$ and simulated paths $P = 3$, the two BER curves closely match and both slightly deviate from the ideal case where the CSI is known.

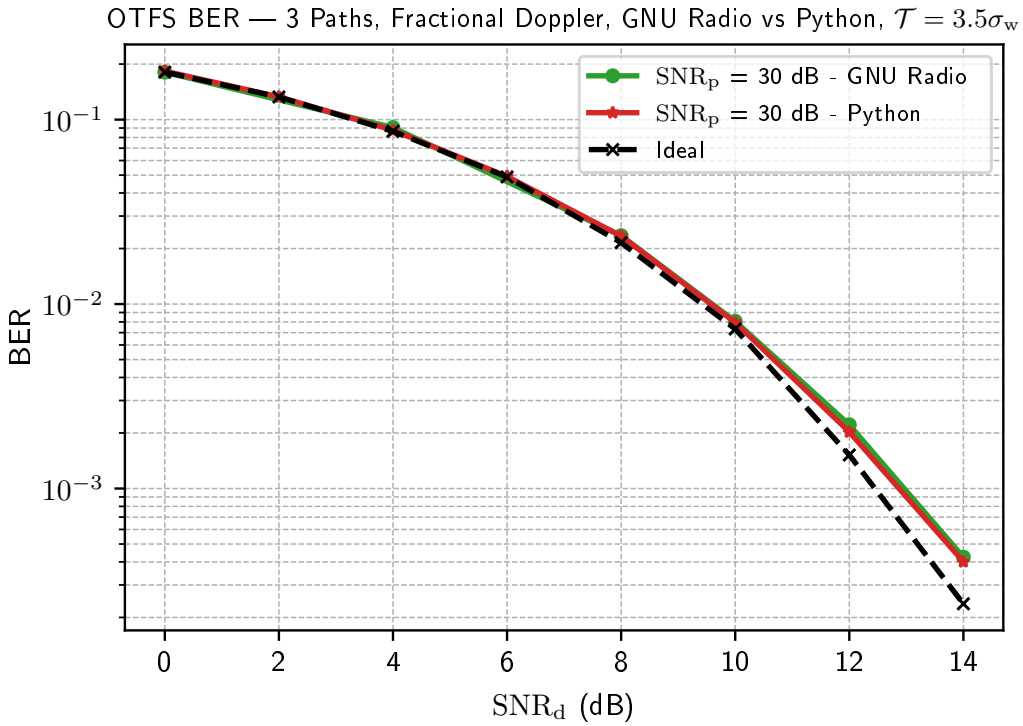


Figure 5.3: Comparison between the BER curves obtained with the standard Python and the GNU Radio flowgraph, for $P = 3$, $\mathcal{T} = 3.5\sigma_\mathrm{w}$ and $\mathrm{SNR_p} = 30$ dB.

5.2 Loop-back configuration

In this section we implement the loop-back configuration of the previously described flowgraph, employing the following hardware:

- The NVIDIA Jetson Orin NX Developer Kit (Fig. 5.4) as the computing platform: this is a compact powerful module that includes a CPU, GPU, memory, and interfaces, all on a single board.
- Universal Software Radio Peripheral (USRP) B205mini-i (Fig. 5.5): SDR transceiver designed by Ettus Research.
  Key specifications:

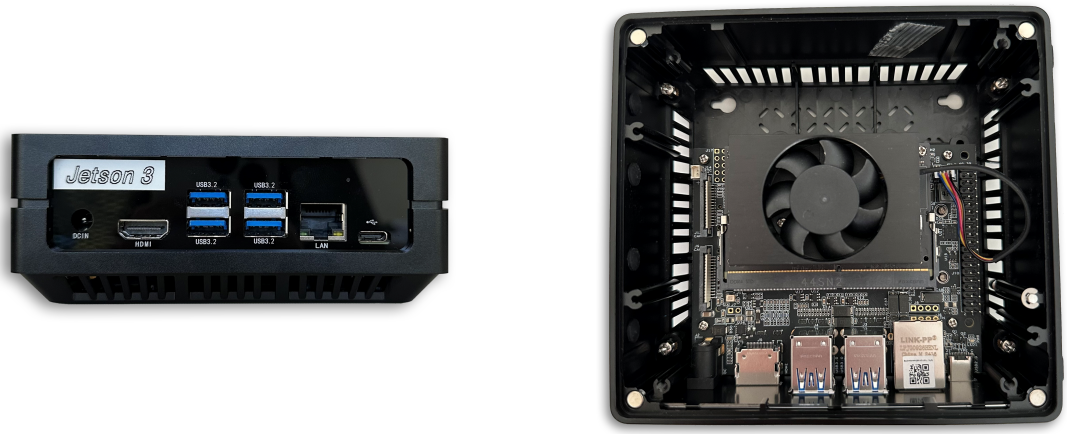


Figure 5.4: NVIDIA Jetson Orin NX Developer Kit: ports and board.

- Channels: 1 TX, 1 RX.
- Frequency range : 70 MHz - 6 GHz.
- Up to 56 MHz of instantaneous bandwidth.
- ADC/DAC resolutions: 12 bits.
- Synchronization: either with 10 MHz clock reference or pulse-per-second (PPS) time reference.



Figure 5.5: USRP B205mini-i: external view and board.

The loop-back configuration is realized by connecting the USRP RF output (transmitter) directly to the RF input (receiver), inserting a 30 dB attenuator, as shown in Fig. 5.6. This setup aims to validate the correctness of the algorithms implemented within the flowgraph (Figs. 5.1 and 5.2) and to perform BER measurements in a controlled environment, free from external channel impairments or interference. Specifically, the channel is generated and managed within the *Time domain channel* block, while the USRP hardware only contributes with white Gaussian noise. The loop-back configuration inherently ensures synchronization in terms of clock, frequency, and phase between the transmitter and

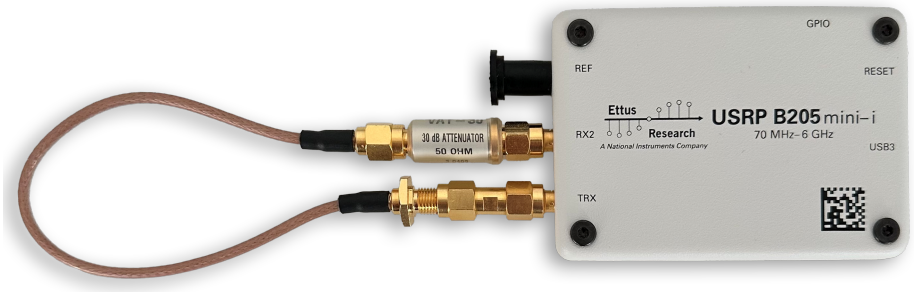


Figure 5.6: Loop-back configuration.

the receiver. However, we observed a lack of time synchronization, resulting in a delay between the beginning of the transmission and the actual start of reception. In order to implement this configuration, we add some blocks both at the transmitter and receiver side, as shown in Figs. 5.7 and 5.8. In particular, at the transmitter, we introduce:

- *Multiplying Const* block: this is used to scale down the signal amplitude and prevent saturation in the USRP's DAC. Specifically, the signal is multiplied by a factor of 3m, which was empirically determined.

- *Vector Source* of 0 samples and *Stream Mux*: these blocks are introduced to avoid underflow errors; this happens when the flowgraph is not supplying samples fast enough to the hardware. Therefore, a sequence of 4000 zeros is prepended to the transmitted signal before sending it to the *USRP Sink* block, implementing a "packetized" version of the OTFS system.

- *Rational Resampler*: this block changes the sampling rate of a signal by a rational factor. Specifically, it is used to match the sampling rate of the flowgraph $f_s = 32$ kHz with $f_s^{\text{USRP}} = 500$ kHz, used by the USRP blocks.

- *Signal Source* and *Multiply*: these blocks implement frequency upconversion. Although the DC Offset correction is enabled in the *USRP Source* block (to be introduced shortly), a residual offset is still observed and affects the SNR measurement at the receiver. To address this, we choose to shift the incoming signal by applying a frequency translation, achieved by multiplying it with $e^{j2\pi f_0 n T_s^{\text{USRP}}}$, where $n \in \mathbb{Z}$ is the sample index, $f_0 = 150$ kHz and $T_s^{\text{USRP}} = 1/f_s^{\text{USRP}}$ is the sampling period.

- *USRP Sink*: this block acts as a transmitter and streams samples to the USRP device. We set the center frequency value $f_c = 162$ MHz, the sampling rate $f_s^{\text{USRP}} = 500$ kHz and gain $G_{tx} = 50$ dB.

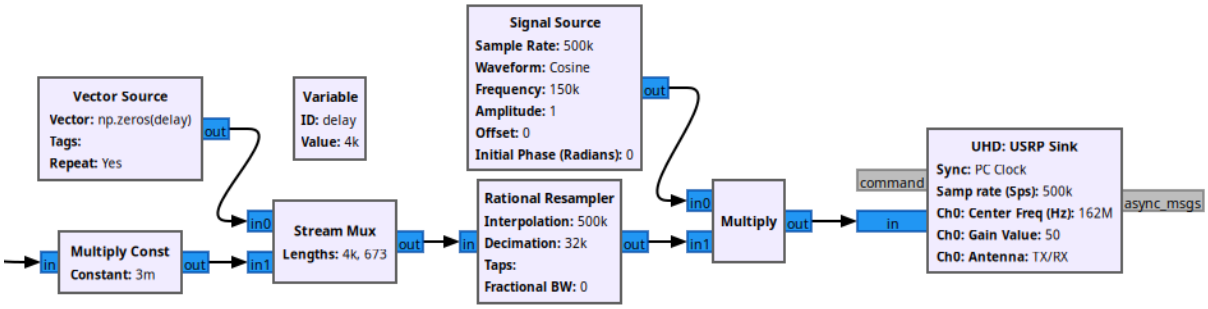


Figure 5.7: Additional blocks in transmission.

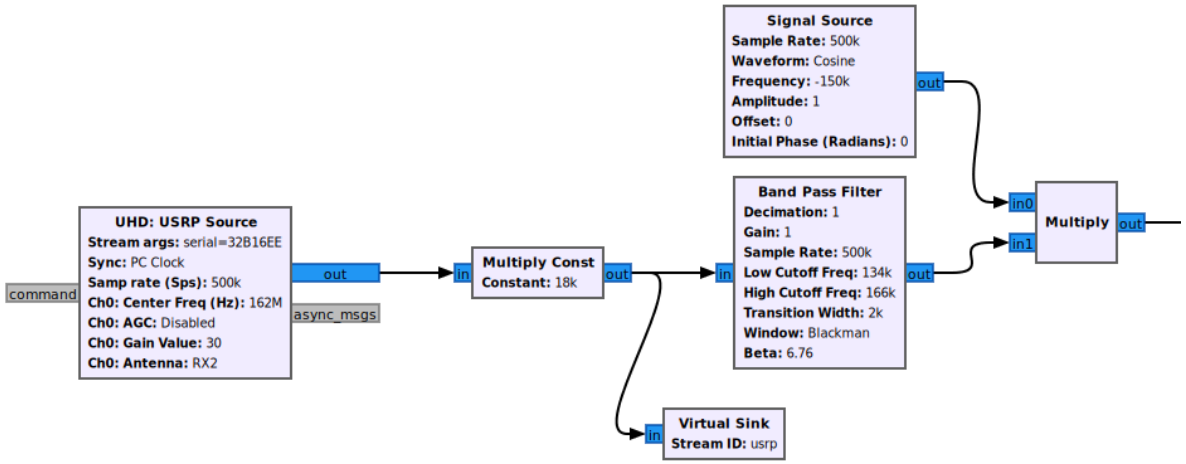


Figure 5.8: Additional blocks in reception.

At the receiver side, the following blocks are added:

- *USRP Source*: operates as the receiver and uses the same sampling rate f_s^{USRP}. The gain is set to $G_{rx} = 30$ dB and the DC Offset correction is enabled in order to remove the average value of the signal which would otherwise be very visible as a peak in the frequency domain. Since the block's center frequency is set to $f_c = 162$ MHz, the received signal appears centered at frequency f_0. This approach ensures that any residual DC offset introduced by the USRP hardware does not overlap with the OTFS signal.

- *Multiplying Const*: since the signal is attenuated at the transmitter side, it is now multiplied by the factor 18000, again empirically determined.

- *Band Pass Filter*: the Band-pass filter (BPF) is used to isolate the desired OTFS signal. This filter is centered at frequency f_0 and has bandwidth 32 kHz.

- *Signal Source* and *Multiply* blocks: to shift the signal back to bandbase, these two blocks implement frequency downconversion. In particular, the filtered signal is

Simulation parameters	
Number of delay bin	$M = 16$
Number of Doppler bin	$N = 8$
Channel delay spread	$l_{\max} = 3$
Channel Doppler spread	$k_{\max} = 3$
Pilot position along delay	$l_p = 12$
Pilot position along Doppler	$k_p = 4$
Pilot power	$P_p = 30$ dB
Decay factor	$a = 0.6$
Fractional Doppler resolution	$D = 8$

Table 5.1: System parameters for loop-back configuration.

multiplied by $e^{-j2\pi f_0 n T_s^{\text{USRP}}}$.

- *Virtual Sink* block: this is used in section 5.2.1 to estimate the SNR at the receiver side.

The system parameters adopted in this setup are different from those used in section 5.1 and are summarized in Table 5.1. This is because, to avoid overflows errors, which occur when the flowgraph is not reading samples fast enough from the hardware, the complexity of the blockchain has to be reduced. In particular, the *ChanEst and LMMSE EQ*, due to the cross-correlation operation and the inversion of N $M \times M$ matrices, is the most computationally heavy block. Therefore, the fractional Doppler resolution D is reduced to 8 and M,N, which determines the matrices dimensions, are set to 16 and 8, respectively. As a consequence, we choose $l_{\max} = 3$ and $k_{\max} = 3$, since $l_{\max} \leq \frac{M-1}{2}$ and $k_{\max} \leq N/2$.

5.2.1 SNR computation

To perform the time-domain block-wise LMMSE equalization (section 2.3.5), it is essential to accurately estimate both the received signal power of the OTFS symbols and the noise power P_{w}. In the loop-back configuration, the Gaussian noise is introduced exclusively by the USRP hardware. As illustrated in Fig. 5.9, by means of the *Virtual Source* block, a BPF is applied to the received signal, in a portion of the spectrum containing only noise. Specifically, the filter is centered at $f_{BP} = -150$ kHz and has a bandwidth of 32 kHz. To bring the filtered noise back to baseband, frequency downconversion is performed by multiplying the signal by $e^{j2\pi f_{\text{BP}} n T_s^{\text{USRP}}}$. Afterward, the noise passes through the *Rational Resampler* and *RRC Matched filter* blocks, resulting is a reduced sampling rate of $f_s' =$

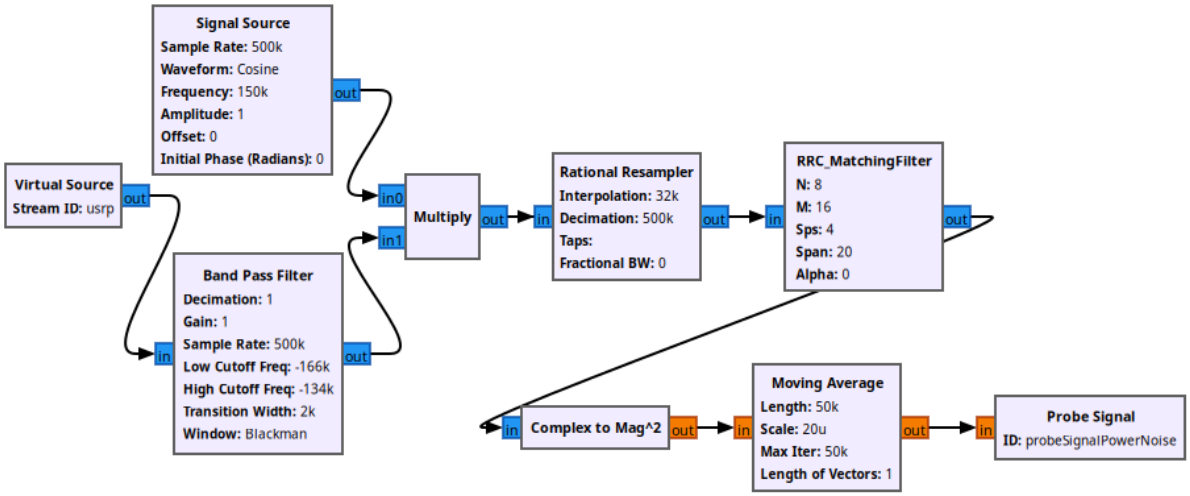


Figure 5.9: Noise power estimation flowgraph.

$f_s/sps = 8$ kHz. Finally, the *Complex to Mag2* and *Moving Average* blocks compute the average noise power at the symbol level. Once the noise power P_{w} is estimated, the power of the received symbols is computed by considering the received $M \times N$ delay-Doppler matrix, denoted as $\mathbf{Y}_{\mathrm{DD}}$ in Fig. 4.9, obtained before the equalization step. Given the known ratio between the power of the pilot and the data symbols, $P_{ratio} = P_p/P_s$, the total power of the received matrix can be expressed as:

$$\begin{aligned} P_{tot} &= \sum_{i=1}^{M}\sum_{j=1}^{N} |\mathbf{Y}_{\mathrm{DD}}[i,j]|^2 \\ &= P_{\mathrm{w}}\, MN + P_s M'N + P_p \\ &= P_{\mathrm{w}}\, MN + P_s(M'N + P_{ratio}) \end{aligned} \tag{5.1}$$

where $M' = (M - 2l_{\max} - 1)$ represents the number of rows that contain useful data. We can now isolate P_s and the received power of the symbols can be derived by inverting (5.1) as:

$$P_s = \frac{P_{tot} - P_{\mathrm{w}}\, MN}{M'N + P_{ratio}}. \tag{5.2}$$

Once P_s and P_{w} are recovered, we can estimate the SNR (dB) at the receiver as:

$$\mathrm{SNR(dB)} = 10\log_{10}P_s - 10\log_{10}P_{\mathrm{w}}. \tag{5.3}$$

Specifically, by employing the flowgraph in Fig. 5.9, the noise power is computed prior to the Monte Carlo simulations of the entire OTFS flowgraph. In this case, the estimated value is approximately -6.635 dB and is assumed constant inside the equalizer block. Once the Monte Carlo simulations of the OTFS chain are completed, the SNR point is

empirically determined. At this point, by changing the power of the transmitted symbols P_s, we compute the BER for different values of the estimated SNR (dB). In particular, the adopted powers P_s are shown in Table 5.2 along with the obtained SNR points. For this configuration, the pilot power is kept constant to $P_p = 30$ dB. The resulting BER curve is presented in Fig. 5.10. This plot shows the BER as a function of the estimated SNR points, for fixed pilot power $P_p = 30$ dB, simulated paths $P = 3$ and threshold $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$. The green curve aims to show just the qualitative trend of the BER computed for the loop-back configuration, highlighting that this setup works correctly.

P_s(dB)	SNR (dB)
0	4
3	7.5
5	9.5
8	12.7
10	14

Table 5.2: Adopted values for the data symbol power P_s (dB), together with the estimated SNR (dB) points, considering a constant pilot power $P_p = 30$ dB.

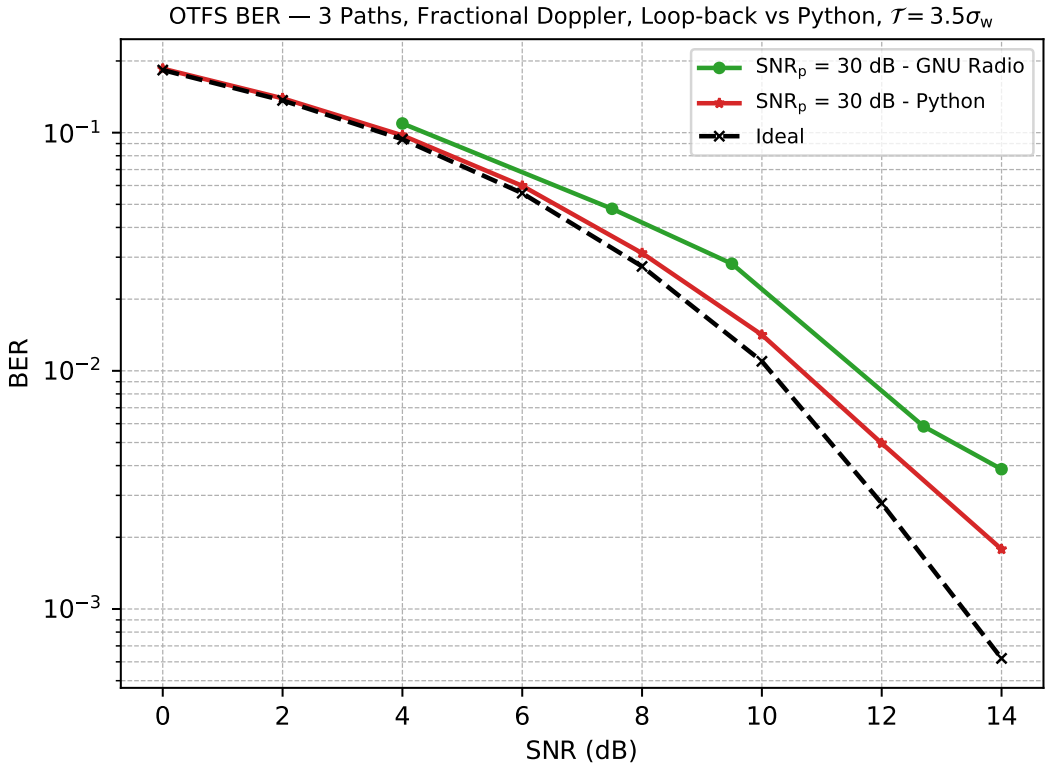


Figure 5.10: Comparison between the BER curves obtained with the standard Python and the GNU Radio loop-back configuration, for $P = 3$, $\mathcal{T} = 3.5\sigma_w$ and $\mathrm{SNR_p} = 30$ dB.

6 Conclusions

This thesis focused on the investigation and implementation of channel estimation algorithms for the OTFS modulation scheme, with the ultimate goal of integrating the full OTFS chain onto a SDR platform. To achieve this objective, an in-depth review covering multiple aspects of the OTFS system and the challenges introduced by high-mobility environments has been carried out.

The work begins by introducing the channel model that captures the doubly-selective nature of rapidly time-varying channels. Various mathematical channel representations are presented, from the delay-time domain, to the frequency-time domain, and finally to the delay-Doppler domain, which shows appealing properties. We have discussed how the frequency-time representation, adopted in the OFDM system, is highly time-variant, leading to the need of a more frequent channel estimation, thus requiring many pilot resources. Moreover, in the presence of non negligible Doppler shifts, performance degradation is exhibited and, since single-tap equalization is no longer effective, the great advantage of having a low-complexity OFDM receiver is lost.

The OTFS modulation, on the other hand, by directly embedding the symbols in the delay-Doppler domain, leverages the sparsity and the stability of the channel in this domain. In the thesis, we focused on the ZP-OTFS variant, whose implementation enables a reduced complexity equalizer and allows an efficient channel estimation in the delay-Doppler domain.

The first major contribution of this work lies in the analysis and realization of channel estimation techniques in the delay-Doppler domain. In particular, an embedded pilot approach is adopted, by transmitting a single pilot symbol within the OTFS frame, together with the data and guard symbols. Initially, we implemented a channel estimation technique for the integer Doppler case, with successive extension to the more realistic and complicated fractional scenario. The receiver, by means of a cross-correlation, detects the fractional Doppler shifts introduced by the channel and, by iteratively executing the

channel estimation algorithm, estimates the gain of each propagation path.

To validate the effectiveness of the proposed methods, the complete OTFS chain is implemented in Python and numerical results have been presented with respect to a case study. The outcome shows that the effect of fractional Doppler shifts can be effectively mitigated through relatively simple additional signal processing, and the channel estimation performance for the integer and fractional cases differs only marginally.

The second main contribution of this dissertation is the integration of the OTFS communication scheme in SDR, using GNU Radio as the framework. First, the corresponding flowgraph of the Python code is replicated and validated. Then, a loop-back configuration is implemented, by introducing the hardware. The initial results confirm that the SDR implementation works correctly in this controlled environment. However, due to time constraints, further over the air experiments could not be conducted.

6.1 Future developments

The first step for future developments in the OTFS SDR implementation involves estimating run-time the noise power P_{w} at the receiver side, which would allow for a more accurate estimation of the SNR operating points. Furthermore, the presented loop-back configuration serves as an initial SDR implementation, intended to validate the OTFS system developed in GNU Radio. This setup could be extended and made more realistic by employing actual antennas both at the transmitter and receiver sides, and introducing a wireless channel into the system. Additionally, the current OTFS chain could be extended by integrating further signal processing blocks, such as those for channel encoding and decoding.

List of Figures

1.1 Communication scheme over single-path channel. 8
1.2 Equivalent low-pass discrete-time channel. 9
1.3 Constellation points and received symbol. 10
1.4 Constellation diagram of 4-QAM. 11
1.5 Static multipath channel. 13
1.6 Time-variant multipath channel. 14
1.8 Delay-Doppler and frequency-time channel representations, with $P = 3$ paths. 16
1.7 Different representations of the channel impulse response and respective relations [15]. 16

2.1 CP of length μ. 21
2.2 ISI between data blocks in channel output. 22
2.3 OFDM transmitter [15]. 23
2.4 OFDM receiver [15]. 24
2.5 Pre-coded OFDM block diagram. 26
2.6 IDZT block using rectangular pulse shaping. 28
2.7 IDZT-based OTFS block diagram. 29
2.8 Time-domain channel matrix $\mathbf{G}$ with three delay paths ($l_{\max} = 2$) divided into $M \times M$ sub-matrices [15]. 32
2.9 Delay-Doppler channel matrix $\mathbf{H}$ with three delay paths ($l_{\max} = 2$) divided into $N \times N$ submatrices [15]. 33
2.10 ZP-OTFS delay-Doppler matrix. 34
2.11 ZP-OTFS time-domain channel matrix $\mathbf{G}$ [15]. 35

3.1 Tx symbol arrangement on the left and Rx symbol pattern on the right. . . 41
3.2 Transmitted and received pilot in delay-Doppler domain for the integer Doppler case. 42

3.3 Transmitted and received pilot in the delay-Doppler domain for the fractional Doppler case. 44
3.4 Rx symbol pattern. 45

4.1 OFDM BER plots in AWGN channel with QPSK modulation. 48
4.2 OFDM diagram block over static multipath channels. 49
4.3 ZF diagram. 50
4.4 MMSE diagram. 51
4.5 Performance comparison between ZF and MMSE equalizers in OFDM system with 3 delayed paths and decay factor $a = 0.2$. 52
4.6 Performance comparison between ZF and MMSE equalizers in OFDM system with 3 delayed paths and decay factor $a = 0.6$. 53
4.7 OFDM BER plots in high-mobility multipath channels with $a = 0.6$. 54
4.8 OTFS vs OFDM BER plots in multipath channels with delays and Doppler shifts, $a = 0.6$. 55
4.9 ZP-OTFS transceiver block diagram. 56
4.10 Channel estimation performances in terms of NMSE as a function of $\mathcal{T}$ (left) and of $\mathrm{SNR_p}$ (right), varying the number of paths $P \in [1, 3, 5, 8]$. The numbers on the curves represent the average number of estimated propagation paths $\hat{P}$ for each $\mathcal{T}$. 58
4.11 Normalized sinc functions (left) for fractional Doppler values with $D = 10$; Channel estimation performances in terms of NMSE (right) as a function of the threshold $\mathcal{T}$, varying the maximum delay tap $l_{\max}$, for $\mathrm{SNR_p} = 30$ dB. 59
4.12 Iterative channel estimation: paths $P = 2$ having same delay tap. 60
4.13 Channel estimation performances in terms of NMSE as a function of $\mathcal{T}$, for different values of the maximum delay tap $l_{\max}$, $D = 100$, $P = 3$ and $\mathrm{SNR_p} = 40$ dB. 61
4.14 NMSE comparison between the integer and fractional Doppler cases as a function of $\mathcal{T}$, with $P = 3$ paths and for different values of $\mathrm{SNR_p}$. 61
4.15 NMSE comparison between the integer and fractional Doppler cases as a function of $\mathrm{SNR_p}$, with 3 paths and for $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$. 62
4.16 OTFS BER plots for the integer and fractional Doppler cases as a function of $\mathrm{SNR_d}$, with fixed threshold $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$, $P = 3$ and for different values of $\mathrm{SNR_p}$. 63

5.1 ZP-OTFS flowgraph, transmitter side. 66
5.2 ZP-OTFS flowgraph receiver side. 67

5.3 Comparison between the BER curves obtained with the standard Python and the GNU Radio flowgraph, for $P = 3$, $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$ and $\mathrm{SNR}_{\mathrm{p}} = 30$ dB. . 68
5.4 NVIDIA Jetson Orin NX Developer Kit: ports and board. 69
5.5 USRP B205mini-i: external view and board. 69
5.6 Loop-back configuration. 70
5.7 Additional blocks in transmission. 71
5.8 Additional blocks in reception. 71
5.9 Noise power estimation flowgraph. 73
5.10 Comparison between the BER curves obtained with the standard Python and the GNU Radio loop-back configuration, for $P = 3$, $\mathcal{T} = 3.5\sigma_{\mathrm{w}}$ and $\mathrm{SNR}_{\mathrm{p}} = 30$ dB. 75

Bibliography

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

[2] S. K. Mohammed, "Derivation of otfs modulation from first principles," *IEEE transactions on vehicular technology*, vol. 70, no. 8, pp. 7619–7636, 2021.

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

[4] J. Zak, "Finite translations in solid-state physics," *Physical Review Letters*, vol. 19, no. 24, p. 1385, 1967.

[5] A. J. Janssen, "The zak transform: a signal transform for sampled time-continuous signals." *Philips Journal of Research*, vol. 43, no. 1, pp. 23–69, 1988.

[6] H. Bolcskei and F. Hlawatsch, "Discrete zak transforms, polyphase transforms, and applications," *IEEE Transactions on signal processing*, vol. 45, no. 4, pp. 851–866, 2002.

[7] F. Lampel, A. Avarado, and F. M. Willems, "On otfs using the discrete zak transform," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2022, pp. 729–734.

[8] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE transactions on Communications Systems*, vol. 11, no. 4, pp. 360–393, 1963.

[9] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for otfs in delay–doppler channels," *IEEE transactions on vehicular technology*, vol. 68, no. 5, pp. 4906–4917, 2019.

[10] H. B. Mishra, P. Singh, A. K. Prasad, and R. Budhiraja, "Otfs channel estimation and data detection designs with superimposed pilots," *IEEE transactions on wireless communications*, vol. 21, no. 4, pp. 2258–2274, 2021.

[11] W. Yuan, S. Li, Z. Wei, J. Yuan, and D. W. K. Ng, "Data-aided channel estimation for otfs systems with a superimposed pilot and data transmission scheme," *IEEE wireless communications letters*, vol. 10, no. 9, pp. 1954–1958, 2021.

[12] P. Sanoopkumar and A. Farhang, "A practical pilot for channel estimation of otfs," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 1319–1325.

[13] N. Hashimoto, N. Osawa, K. Yamazaki, and S. Ibi, "Channel estimation and equalization for cp-ofdm-based otfs in fractional doppler channels," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–7.

[14] L. Calandrino and M. Chiani, *Lezioni di Comunicazioni Elettriche*. Pitagora Editrice Bologna, 2013.

[15] Y. Hong, T. Thaj, and E. Viterbo, *Delay-Doppler Communications: Principles and Applications*. Academic Press, 2022.

[16] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[17] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "Otfs—predictability in the delay-doppler domain and its value to communication and radar sensing," *IEEE BITS the Information Theory Magazine*, vol. 3, no. 2, pp. 7–31, 2023.

[18] F. Conillera Vilar, "Implementation of zero forcing and mmse equalization techniques in ofdm," 2015.