
Towards Cellular Digital Twins

UNDERGRADUATE THESIS

*Submitted in partial fulfillment of the requirements of
BITS F421T Thesis*

By

PULAK MEHROTRA
ID No. 2021AATS0017P

Under the supervision of:

Dr. Dinesh Bharadia, Assistant Professor, UCSD

&

Dr. Sandeep Joshi, Assistant Professor, BITS Pilani



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI,
PILANI CAMPUS

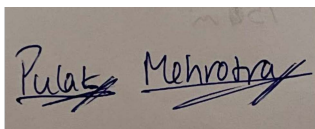
December 2024

Declaration of Authorship

I, Pulak Mehrotra (ID: 2021AATS0017P), declare that this Undergraduate Thesis titled, 'Towards Cellular Digital Twins' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

A rectangular box containing a handwritten signature in black ink. The signature appears to read 'Pulak Mehrotra' with a stylized, cursive script.

Date:

December 24, 2024

Certificate

This is to certify that the thesis entitled, “*Towards Cellular Digital Twins*” and submitted by PULAK MEHROTRA ID No. 2021AATS0017P in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.



Supervisor

Dr. Dinesh Bharadia

Assistant Professor,

University of California, San Diego

Date:

Co-Supervisor

Dr. Sandeep Joshi

Assistant Professor,

BITS Pilani

Date:

“When in doubt, always err on the side of taking action.”

JEFF BEZOS

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

Abstract

Bachelor of Engineering (Hons.)

Towards Cellular Digital Twins

by PULAK MEHROTRA

Multimedia applications are witnessing a surge in popularity and constant evolution, often surpassing the advancements in network infrastructure. For application developers, comprehending the performance of these applications on cellular networks poses a significant challenge. Although the concept of digital twins in communication and network systems research is not new, the pursuit of identifying an effective, easy-to-deploy twin or framework for efficient app testing on radio networks remain critical.

We seek to address a fundamental question regarding the PHY-layer representation of a channel: *What constitutes a significant change in the PHY-layer, and how does it affect the MAC+ layers?* We then introduce a pipeline that condenses a traditional channel representation into a more concise, software-friendly format, along with an open-source 5G framework that incorporates these representations. Our experiments illustrate that a systematic and theoretically grounded method for modifying the wireless channel representation has an almost negligible impact on overall system performance. The steps outlined in this thesis represent an initial effort toward developing a realistic, open-source digital twin of a cellular network, referred to as a "*full twin*".

Acknowledgements

I wish to express my profound gratitude to my advisor, Dr. Dinesh Bharadia, for her steadfast belief in my abilities and for his significant contributions to this research endeavor. Additionally, I express my gratitude to Ushasi Ghosh and Dr. Ish Kumar Jain for their endeavors in augmenting my understanding, generating research concepts, and participating in fruitful collaborative ideation sessions. I express my sincere appreciation for the insightful feedback and constructive criticism provided by Dr. Sandeep Joshi, my co-supervisor. The faculty and personnel of the Electrical and Electronics Engineering department at Birla Institute of Technology and Science are especially thanked for their instruction and direction. I express profound gratitude to my peers and acquaintances for their steadfast encouragement during the course of my thesis. Finally, I would like to extend my sincere appreciation to my family members for their unwavering support and solid presence throughout my voyage.

Contents

Declaration of Authorship	i
Certificate	ii
Abstract	iv
Acknowledgements	v
Contents	vi
List of Figures	viii
List of Tables	ix
Abbreviations	x
1 Introduction	1
2 Background And Related Work	4
2.1 Background	4
2.1.1 Wireless Multipath Model	4
2.1.2 Doppler Shift and Coherence Time	5
2.1.3 Orthogonal Frequency Division Multiplexing (OFDM)	6
2.2 Related Work	6
3 Wireless Channel Representation	9
3.1 Problem Formulation	9
3.1.1 Core Problem	9
3.1.2 Rationale	9
3.1.3 Challenges	11
3.2 Proposed Solution	12
3.2.1 collectTaps	12
3.2.2 selectTaps	13
3.2.3 processTaps	14
4 Experimental Framework	15

4.1	5G System Architecture	15
4.2	Channel Simulation Setup	16
5	Channel Representation Pipeline	20
5.1	Tap Reduction	20
5.1.0.1	How can you classify a path as non-dominant?	22
5.1.1	Solution Overview	22
5.1.2	Metrics-In-Use	23
5.1.2.1	Error Vector Magnitude (EVM)	23
5.1.2.2	Bit Error Rate (BER)	24
5.1.3	Experiments Performed	25
5.2	Tap Updation	27
5.2.0.1	What is a breakpoint?	28
5.2.1	Solution Overview	28
5.2.2	Metrics-In-Use	29
5.2.2.1	Channel Tap Deviation	29
5.2.2.2	OFDM Output Deviation	30
5.2.3	Experiments Performed	31
6	Conclusion	33
	 Bibliography	 34

List of Figures

3.1	Comparison of Channel Impulse Responses for Two Different Channel Models	10
3.2	Comparison of OFDM Output Symbols for Two Different Channel Models	11
3.3	OFDM System Before and After MMSE Equalization	11
3.4	Proposed Solution Workflow	12
3.5	This figure illustrates the addition of an extra signal processing block within the OAI stack in the downlink (DL) to perform real-time convolutions. .	13
3.6	Effect of Position Variation on the CIR	13
4.1	5G System Architecture	16
4.2	Comparison of RF Simulator Systems in OAI 5G	17
4.3	RSRP Variation with Varying Channel Taps. As you can see each packet received at the UE side has a different RSRP value, indicating a change in the channel representation.	18
4.4	OFDM System Used to Perform Channel Simulations in Sionna	18
4.5	Channel Tap extraction Pipeline	19
5.1	Visual Representation of Tap Reduction	20
5.2	Etoile (LoS) CIR and Channel Taps	21
5.3	Munich (NLoS) CIR and Channel Taps	21
5.4	Evaluation Metrics for Tap Reduction	23
5.5	What is EVM?	23
5.6	Comparison of Munich scenario's EVM across different modulation schemes	25
5.7	Comparison of Etoile scenario's EVM across different modulation schemes	26
5.8	BER vs. Tap Resolution	27
5.9	Percentage Change in BER vs. Tap Resolution	27
5.10	<i>Visualization of Breakpoint Selection:</i> The algorithm selects the breakpoints where the channel model changes significantly. The distances between breakpoints can be assumed to be the empirically determined coherence times of the channel.	28
5.11	Mobility Scenario in Sionna	31
5.12	Channel Tap Deviation Metrics vs. Distance	32
5.13	Comparison of OFDM Outputs	32

List of Tables

2.1	Doppler Shift Calculation for different source and observer motions . . .	5
2.2	OFDM System Parameters	6
2.3	Comparison of Different Existing Cellular Digital Twin Frameworks . . .	8

Abbreviations

OFDM	O rthogonal F requency D ivision M ultiplexing
MIMO	M ultiple I nput M ultiple O utput
CIR	C hannel I mpulse R esponse
EVM	E rror V ector M agnitude
BER	B it E rror R ate
SNR	S ignal to N oise R atio
QAM	Q uadrature A mplitude M odulation
QPSK	Q uadrature P hase S hift K eying
AWGN	A dditive W hite G aussian N oise
MCS	M odulation and C oding S cheme
RSRP	R eference S ignal R eceived P ower
UE	U ser E quipment
BS	B ase S tation
NLoS	N on- L ine of S ight
LoS	L ine of S ight
RAN	R adio A ccess N etwork
PHY	P hysical L ayer
MAC	M edia A ccess C ontrol
UL	U p L ink
DL	D own L ink

Dedicated to my parents, for their endless support.

Chapter 1

Introduction

In this chapter, we outline the primary motivation behind this study, emphasizing the urgent need for advancements in cellular wireless digital twins.

The next generation of cellular wireless networks is driven by demanding applications, including large-scale autonomy, extended reality, and Internet of Things (IoT) sensing and control [1]. These applications will rely heavily on reliable wireless access to edge computing resources, which will host critical intelligence for their operation. They require stringent guarantees [2] regarding network performance—specifically in terms of throughput, latency, and jitter—while also necessitating the network’s ability to adapt quickly to evolving requirements. Joint development of these applications alongside their complementary network-interfacing algorithms poses significant challenges, particularly when using commercial radio access networks (RAN) or large-scale testing frameworks such as Colosseum [3] or Powder [5]. Currently, the prevalent practice is to conduct initial tests in simplified, simulated environments during the development phases, with final evaluations reserved for real-world radio networks.

This generally accepted practice has its own set of hurdles, similar to those encountered in larger-scale equivalents. For instance, setting up and maintaining over-the-air connections with commercial off-the-shelf (COTS) radios is often tedious and requires specialized expertise. In contrast, real large-scale cellular networks are costly, challenging to manipulate, and, most importantly, not very accessible. Furthermore, many existing cellular network simulators [3, 5] lack the lightness and software-defined flexibility desired for practical development.

It is challenging to capture the complex dynamics of cellular networks, which profoundly affect application performance, while also ensuring that the testing environment remains user-friendly throughout the application development cycle. A lightweight digital twin of a real-world cellular network, or a *"full twin"*, would be a valuable resource for application

developers, as it would permit continuous refinement and evaluation of both application-level and network-level algorithms. Keeping the above stipulations in mind, we propose the following design requirements for a "*full twin*" and explain how we attempt to satisfy these requirements in this thesis:

1. **Lightweight and Open-Source:** The framework should be able to run on local compute (CPUs) with low memory and compute overheads. Users should be able to seamlessly conduct end-to-end evaluations on our framework, devoid of intricate 5G know-how or the intricacies of hands-on physical testing. The framework should be open-source, allowing for easy setup as well as easy customization. We achieve this by developing an open-source, containerized platform that facilitates easy deployment.
2. **Support RAN Optimization:** The pivotal challenge rests in curating an extensive, yet user-friendly catalogue of RAN parameters and functions that can be effortlessly manipulated by software developers, even those devoid of radio network expertise. Our framework, equipped with a broad array of parameters and functions, empowers users to pinpoint the configurations best suited for their applications. We do so by building on top the OpenAirInterface 5G (OAI) [14] platform, an existing widely adopted 5G emulated RAN stack, which provides a comprehensive set of RAN parameters and functions.
3. **Verifiable Real-World Accuracy:** Central to our initiative is the quest for veracity. To the best of our knowledge, no literature on digital twins presents a cogent juxtaposition of metrics at multiple levels between the physical realm and its digital counterpart. The crux of the challenge stems from identifying and replicating the optimal set of channel parameters from reality to ensure congruent network performance. We achieve this by utilizing real-world representative wireless channels from Sionna [7], a widely adopted channel modeling framework, and conducting extensive experiments with these channels before integrating them into our framework.

The overarching issue with existing Digital Twin solutions is the lack of an appropriate PHY layer. While the MAC+ layers are well-represented in existing solutions, the PHY layer is often abstracted away or represented in a manner that is not representative of real-world scenarios. Our major contribution in this work is incorporating a realistic PHY layer into existing MAC+ layer stacks while also ensuring that the framework remains realistic and standards-compliant.

This begs a further question: what level of artifacts, control or metrics at the PHY-level would attain the ideal trade-off between capabilities, accuracy, and simplicity in the higher layers? A key insight guiding this research is the understanding that **not all PHY-layer**

changes affect the MAC+ layers equally; thus, application-level performance in a cellular network is directly proportional to only the *significant* variations of the wireless channel at the PHY layer. Commercial-level 5G NR transmissions utilize OFDM, which often involves complex channel estimation and equalization methods to reduce the effect of the wireless channels on the transmitted symbols. We acknowledge this complexity, and we aim to leverage it to our advantage by designing a lightweight, yet accurate, wireless channel representation that can be directly integrated with existing solutions today. We further argue that not all PHY-layer channel variations yield the same impact on the OFDM system's output. Furthermore, we contend that not all changes are equally significant to the application layer. This argument is further developed in Chapter 3.

In this study, we aim to take significant initial steps toward representing a wireless channel in a manner that is lightweight, efficient, and accurate. Through extensive experimentation leveraging frameworks such as Sionna [7], we investigate the impact of varying channel representations on both network and application performance. By the conclusion of this thesis, we will present a comprehensive end-to-end channel pipeline designed to effectively represent the wireless channel. We will also provide details of how to modify existing RAN emulators to incorporate these channel representations, thereby enabling a fully realistic cellular network environment.

The remainder of this thesis is structured as follows:

- Chapter 2 provides a detailed background and related work, outlining the key concepts and technologies relevant to this study.
- Chapter 3 delves into the problem formulation, outlining the challenges and goals of this research.
- Chapter 4 presents the system framework, detailing the architecture and components of the proposed solution.
- Chapter 5 describes the experiments performed, including the simulation setup and results.
- Chapter 6 concludes the thesis, summarizing the key findings and outlining future research directions.

Chapter 2

Background And Related Work

2.1 Background

In this section, we provide a brief overview of the key concepts and technologies that are relevant to fully understanding the work presented in this thesis.

2.1.1 Wireless Multipath Model

Scattering, reflection, and diffraction are among the main effects in electromagnetic propagation. A general mathematical description of a wireless channel, seen as a linear time-varying system, is given by its impulse response [18, 16]. A general model can be written as [17]:

$$h(t, \Theta, \Phi) = \sum_k a_k(t) \delta(t - \tau_k(t)) \delta(\Theta - \Theta_k(t)) \delta(\Phi - \Phi_k(t)) \quad (2.1)$$

We represent this model as a channel impulse response (CIR) in the time domain, where $a_k(t)$ is the complex gain, $\tau_k(t)$ is the delay (or time-of-flight) of path k , $\Theta_k(t)$ is the azimuth and elevation angle of departure (AoD), and $\Phi_k(t)$ is the azimuth and elevation angle of arrival (AoA). Going forward, we use $\varphi_k = (\Theta_k, \Phi_k)$ as a shorthand to collectively represent all angles. Intuitively, equation (1) represents each path as a Dirac function in time-angle space. In general, the role of angle metrics is to capture the spatial diversity of the channel, which is crucial for beamforming and spatial multiplexing. However, these metrics are not as impactful when considering regular vanilla OFDM 5G-NR transmissions. Therefore, the task of channel modeling can be reduced to predicting the channel attributes $((a_k(t), \tau_k(t)))$, as shown below:

$$h(t, \Theta, \Phi) = \sum_k a_k(t) \delta(t - \tau_k(t)) \quad (2.2)$$

In the discrete domain, the channel model is typically represented as a set of channel taps, where each of the multipaths in a channel impulse response corresponds to a taps. Each tap is a complex number representing the gain and phase of the multipath components. We model the channel as a complex vector of channel taps, treating it as a linear time-invariant system. The channel taps serve as the coefficients of the *filter* representing this system. The exact method we apply for this representation is discussed in Chapter 4 of this thesis.

2.1.2 Doppler Shift and Coherence Time

Doppler shift $f_o = f_s \left(\frac{v \pm v_o}{v \mp v_s} \right)$	Stationary observer	Observer moving towards source	Observer moving away from source
Stationary source	$f_o = f_s$	$f_o = f_s \left(\frac{v+v_o}{v} \right)$	$f_o = f_s \left(\frac{v-v_o}{v} \right)$
Source moving towards observer	$f_o = f_s \left(\frac{v}{v-v_s} \right)$	$f_o = f_s \left(\frac{v+v_o}{v-v_s} \right)$	$f_o = f_s \left(\frac{v-v_o}{v-v_s} \right)$
Source moving away from observer	$f_o = f_s \left(\frac{v}{v+v_s} \right)$	$f_o = f_s \left(\frac{v+v_o}{v+v_s} \right)$	$f_o = f_s \left(\frac{v-v_o}{v+v_s} \right)$

TABLE 2.1: Doppler Shift Calculation for different source and observer motions

Doppler Shift refers to the change in frequency (or wavelength) of a signal due to the relative motion between the source and the observer. In cellular wireless communications, it is caused by the relative motion between users (User Equipments of UEs) and the base stations (or gNBs) within the communication channel. This effect leads to frequency variations that impact the quality and reliability of the received signal, especially in mobile scenarios where users are moving at high speeds. The Doppler shift can be quantified by the equation:

$$f_d = \frac{v}{\lambda} \quad (2.3)$$

where f_d is the Doppler shift frequency, v is the relative velocity between the transmitter and receiver, and λ is the wavelength of the signal.

Coherence time is a measure of the time duration over which the channel can be considered approximately invariant. It is influenced by the Doppler shift and the velocity of the mobile users. A longer coherence time implies that the channel characteristics do not change significantly over time, allowing for more reliable communication. It can be defined using the relation:

$$T_c \approx \frac{1}{f_d} \quad (2.4)$$

where T_c is the coherence time and f_d is the maximum Doppler shift frequency.

The coherence time of a channel is a metric that is often empirically determined. We tackle this issue in Chapter 5 of our thesis. In our work, we consider the coherence time as a key parameter in the channel model, as it directly affects the performance of the wireless communication system.

2.1.3 Orthogonal Frequency Division Multiplexing (OFDM)

Orthogonal Frequency Division Multiplexing (OFDM) is a technique used in wireless communications that divides the available frequency spectrum into multiple orthogonal subcarriers. Each subcarrier carries a portion of the data, allowing for high spectral efficiency and robustness against frequency selective fading. OFDM is characterized by its ability to reduce inter-symbol interference (ISI) and improve performance in multipath environments.

The key components of an OFDM system include the cyclic prefix, a guard interval inserted between OFDM symbols to mitigate inter-symbol interference, and the subcarrier spacing, which determines the frequency separation between adjacent subcarriers. OFDM is considered computationally intensive due to its use of Fast Fourier Transforms (FFTs) for modulation and demodulation, representing a significant improvement over traditional multi-oscillator systems. It is widely used in modern communication standards such as LTE and Wi-Fi due to its efficient use of bandwidth and simplicity in implementation. The OFDM system utilized in our setup is depicted in Figure 4.4. The various OFDM parameters we use, selected based on 3GPP specifications, are detailed in Table 2.2.

2.2 Related Work

The concept of creating a digital twin of a real-world cellular network has been explored in various research efforts [3, 5, 19, 10, 13] aimed at developing realistic network simulators for evaluating application and network layer metrics. These initiatives span industry

TABLE 2.2: OFDM System Parameters

Parameter	Value
Centre Frequency	2.149 GHz
Bandwidth	20 MHz
Sampling Frequency	20 MHz
Maximum Reflections Allowed	5
OFDM Symbols per TX	14
Number of Subcarriers	72
Subcarrier Spacing	30 kHz
Cyclic Prefix	6 ns

projects, such as the 5G digital twin platforms [10], and academic endeavors [3, 5] that focus on lightweight simulators for specific applications.

Key capabilities of platforms like POWDER [5] and Colosseum [3] include realistic network simulation and the creation of virtual replicas of 5G infrastructure for accurate testing and optimization. They facilitate automated testing of intricate scenarios via remote access, support Open Radio Access Network (Open RAN) technologies for advanced research, and leverage AI models and analytics to capture real-time network behavior. However, these platforms fall short of offering a lightweight digital twin solution that developers need for evaluating application performance and network-interfacing algorithms effectively. Remote access is difficult and daunting for developers, and the platforms are not designed for rapid prototyping and testing of new algorithms.

The 5G digital twin in [3] offers intricate FPGA-based processing of I/Q samples to accurately capture the wireless signal propagation in a radio network. The collection of realworld I/Q data is daunting and often one of the easiest points of failure in a signal collection framework. Such data demands vast storage capacities, often running into terabytes, and poses significant privacy issues. Moreover, procuring such data often requires commercial-grade Software-Defined Radios (SDRs), making the process challenging and costly. Platforms from NVIDIA [10] have presented extensive high-performance computing frameworks, with PHY channel characterization techniques based on ray tracing, for a 5G digital twin network. Such frameworks, including the work colO-RAN [15], are comprehensive infrastructures to drive high quality physical layer research, network optimization and planning.

However, our aspiration lies in leveraging open-source 5G stacks to create a twin framework that mirrors the software 5G within local computing environments to benchmark application level metrics. Notably, we intentionally bypass all intricate PHY layer processing, only retaining PHY artifacts that we deem significantly impact algorithmic or application testing. Simulators like ns-3 [19] and ueransim are well known for their 5G network simulation environments, but they sometimes fail to faithfully reproduce the nuances of a real-time 5G stack, underscoring the known sim2real gap in such technologies. On the other hand, network researchers often evaluate their algorithms using trace drive network emulators such as mahimahi [13], which can oversimplify and misrepresent real-world scenarios. Pioneering works on ABR algorithms [11], congestion control mechanisms, and transport protocols [8] are only evaluated using a mobile network operator’s traces. These may not always provide an accurate representation of real-world cellular environments.

There also exist several works which offer a very simplistic testing framework for evaluation of their network algorithms. They are based on a trace driven network emulator or licensed cellular network simulators. Such works fail to address the sim2real gap popular

Framework	Lightweight	Ease of data collection	Realtime Processing	Verifiable real world accuracy	Support end-to-end application testing	Support RAN optimization
Colosseum DT [3]	No	No	Yes	No	Yes	Yes
Nvidia DT [10]	No	No	Yes	No	Yes	Yes
ueransim	Yes	Yes	No	No	Yes	No
ns-3 [19]	Yes	No	No	No	No	No
Mahimahi [13]	Yes	Yes	No	No	Yes	No

TABLE 2.3: Comparison of Different Existing Cellular Digital Twin Frameworks

in simulation techniques. Our work focuses on providing a streamlined, lightweight, and easily deployable digital twin platform, designed to closely mimic real-world cellular networks for testing various Internet network algorithms and applications. Contrary to many existing models that do not delve into intricate physical (PHY) level channel representations, we argue that a simpler, compute-efficient channel abstraction is sufficient for end-to-end networked algorithm and application testing over cellular networks. Our goal is an abstraction that accurately captures the essential elements of real-world networks, like throughput and packet drops for each connected user.

Chapter 3

Wireless Channel Representation

3.1 Problem Formulation

3.1.1 Core Problem

To state the problem of this thesis more clearly: We do not aim to develop an all-encompassing channel model for every instance of the wireless channel. Instead, we focus on understanding the ramifications of changes in specific components of the channel model on the overall performance of the RAN. This is the core problem we aim to address. We collect channel models (in this case channel taps) of the system over a given time frame and convert them into a format that can be integrated into our an existing RAN emulator. By doing so, we can analyze how variations in the PHY-layer affect higher layers and overall network performance.

3.1.2 Rationale

As stated in the previous sections of the paper, the main goal of this study is to develop a realistic digital twin of a wireless cellular network that can be used to evaluate the performance of applications and network-interfacing algorithms. To achieve this goal, we need to develop a channel representation that accurately captures the key characteristics of the wireless channel, such as multipath fading, Doppler shift, and coherence time, while also creating a framework that is lightweight and easy to use. For this goal, it is very clear we can split our problem into two parts:

1. Develop a *reduced* channel representation that accurately captures the key characteristics of the wireless channel.

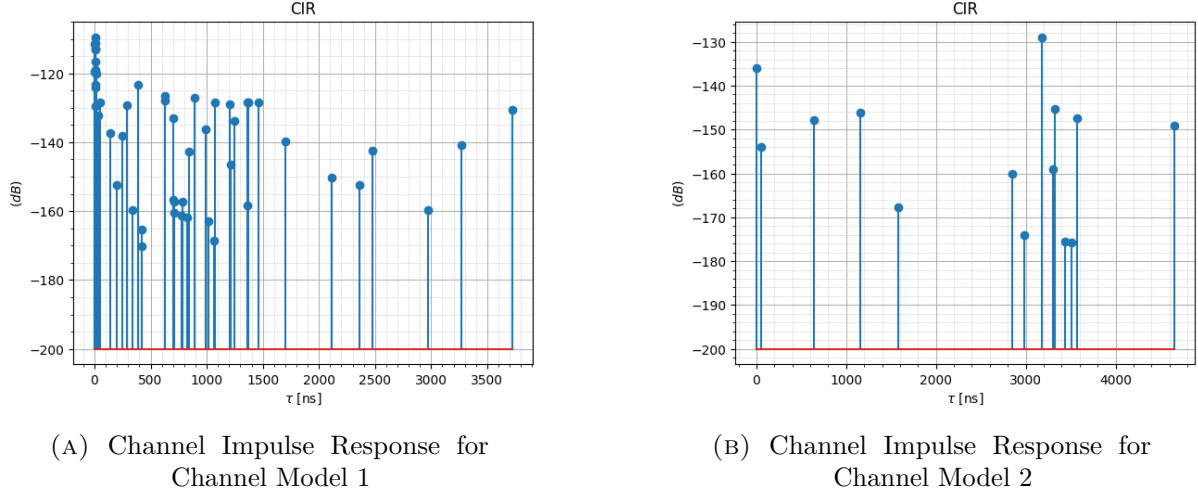


FIGURE 3.1: Comparison of Channel Impulse Responses for Two Different Channel Models

2. Create an open-source API on top of a RAN stack, that can be used to interact with the channel representation and evaluate the performance of various applications.

The key observation we leverage is that not all PHY-layer changes have an equal impact on the MAC+ layers. The main reason for such a discrepancy is that the MAC+ layers are designed to be robust to changes in the PHY-layer, through heavily optimised signal processing units like LDPC Encoding [4] and MSSE Equalization [6]. In our Sionna simulation, we employ Least Squares channel estimation [9] using Kronecker pilot symbols. This method is highly robust for estimating channel taps. This channel estimate is then used to perform MMSE equalization, leading to a system devoid of much of the randomness on the account of noise and varying channel models.

We hope to illustrate this observation by performing a simple experiment where we plot the output the OFDM output for two completely different channel models, and observe the difference in the equalized output OFDM symbols. The CIRs for the two channel models are shown in Figure 3.1, with the corresponding 16 QAM OFDM output shown in Figure 3.2. As you can notice, the output of the OFDM signal looks extremely similar for both the channel models, in the fact that all the symbols have been allotted to the quadrants their transmitted signals belong to. For further context, in Figure 3.3 we can compare how the OFDM system look before and after MMSE equalization for a high-SNR, flat-fading scenario. We can observe that a given channel imparts a constant phase shift to the transmitted symbols, a phase shift the MMSE equalizer attempts to correct for by applying an opposite of the phase shift per symbol.

Note, while the OFDM system output is *similar* for both systems, it is not *identical*. We attempt to use this reduced randomness to our favor in the following sections to design an appropriate channel representation.

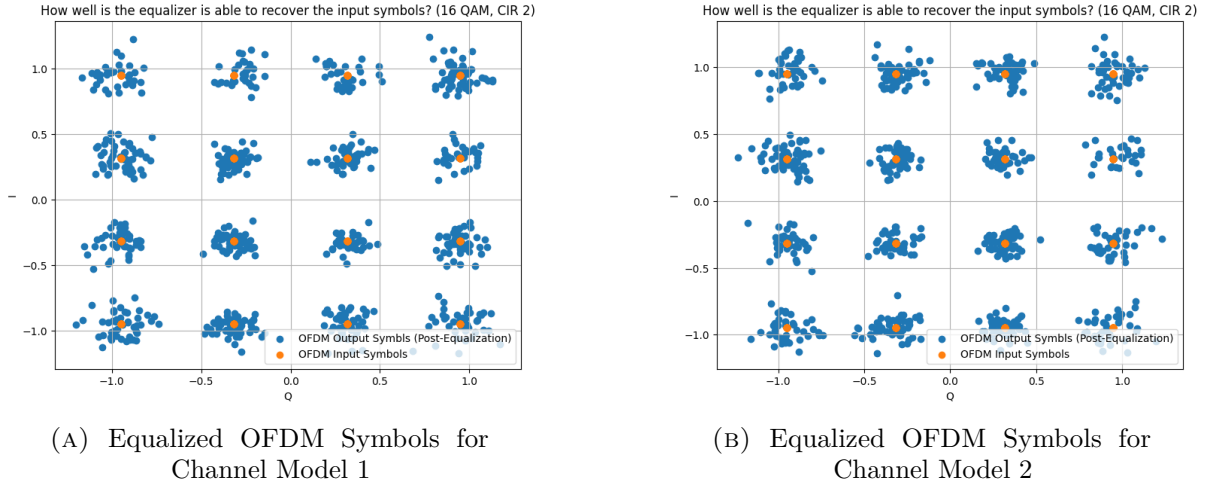


FIGURE 3.2: Comparison of OFDM Output Symbols for Two Different Channel Models

3.1.3 Challenges

To effectively integrate the channel model into the RAN-level simulator, we can represent the wireless channel as a filter with tap coefficients applied to the transmitted signal. Each channel model is defined by a set of channel taps. Historically, convolutions have been used to simulate the channel's impact on transmitted signals, but they can be computationally intensive. To create a software-only system, we require a channel representation that is lightweight and efficient. This presents one of the key challenges addressed in this thesis: **What is the optimal number of taps needed to accurately represent the channel?**

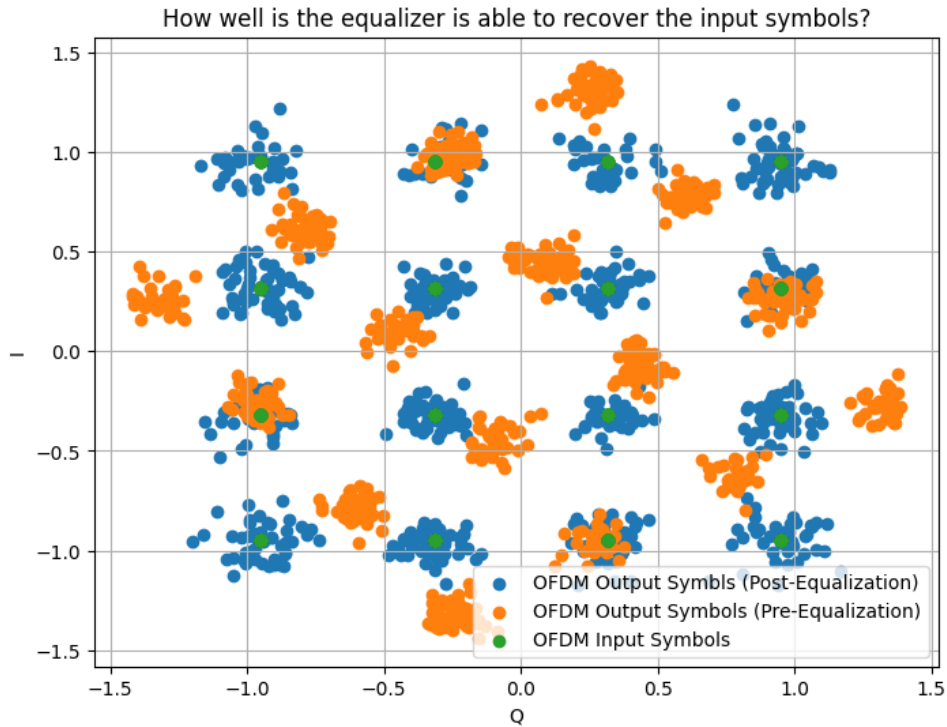


FIGURE 3.3: OFDM System Before and After MMSE Equalization

One of the primary challenges in modeling a wireless channel is its inherently time-varying nature. This variability means that channel taps can fluctuate irregularly, complicating accurate representation. While it is common to assume that channel taps remain constant (or have flat fading characteristics) over a period known as the *coherence time*, this duration is not fixed; it can range from just a few milliseconds to several seconds, depending on environmental factors. This variability leads to the second key challenge addressed in this thesis: **How do we model the time-varying nature of the channel?**

We answer both these questions in the following sections.

3.2 Proposed Solution

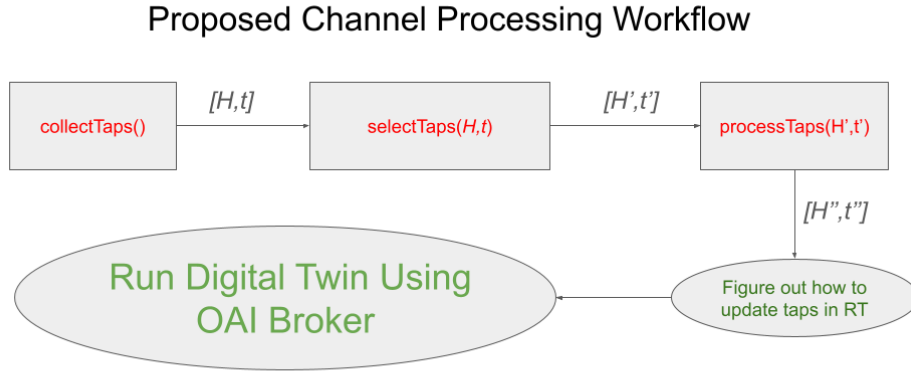


FIGURE 3.4: Proposed Solution Workflow

For the above posed problem, we propose a solution that is agnostic to our given setup, and should work across various scenarios. Our solution, as outlined in Figure 3.4, has three main steps:

3.2.1 collectTaps

The overarching idea behind our solution is to introduce abstraction into our data transmission by convolving the input with channel taps. Channel taps are simply a discretely-sampled version of the Channel Impulse Response (CIR) of a scene. We can obtain such taps from various methods such as using sniffers like SignalHound and run an accurate simulation of the environment for a given instant. For now, we use Sionna to get channel representations and will move on to over-the-air experiments after this.

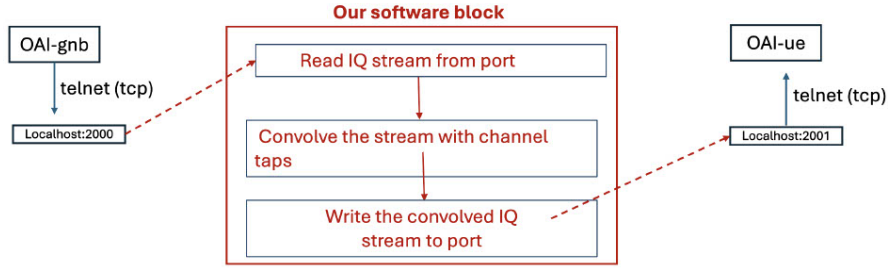


FIGURE 3.5: This figure illustrates the addition of an extra signal processing block within the OAI stack in the downlink (DL) to perform real-time convolutions.

3.2.2 selectTaps

Each system, no matter how big or small, will have a limit on the number of sliding window convolutions it can perform in real time. For most modern-day CPUs, this limit depends on the GFLOPS of the CPU and the number of taps in the channel representation. It is reasonable to assume that the channel representation will contain a higher resolution of taps than the system supports, especially if we consider moving to a real-time system and complex systems such as MIMO. Hence, using this function we attempt to find the most succinct representation of the channel. We present a simple example in Figure 3.5 to illustrate this point.

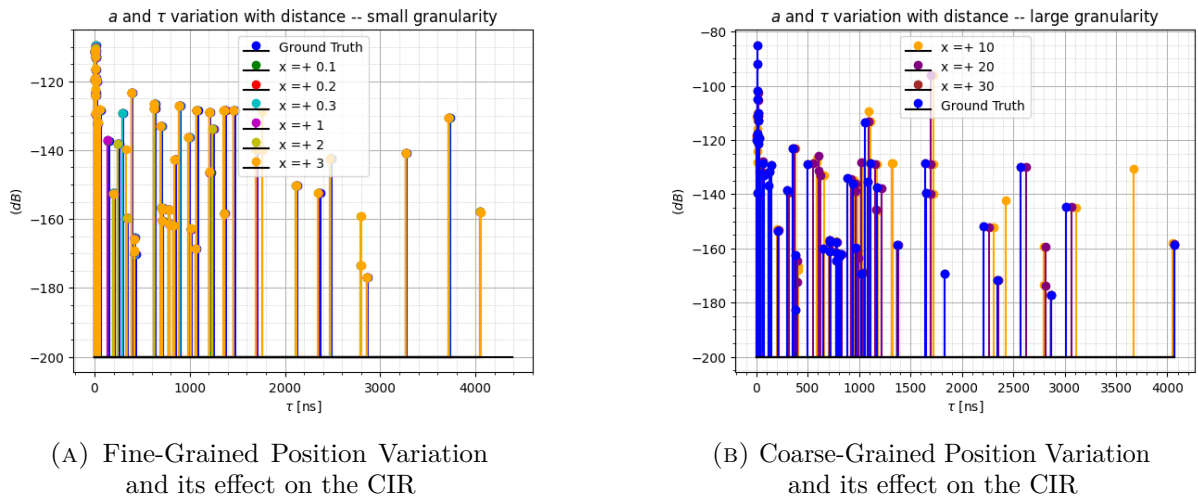


FIGURE 3.6: Effect of Position Variation on the CIR

3.2.3 processTaps

A wireless channel can be considered a spatio-temporal system, varying over both time and space. However, not all variations in the channel have an equal impact on the system. This step involves mapping significant changes in the system to the CIRs that produce them.

To demonstrate this fact, we consider the following example: we take a Sionna system and plot the Channel Impulse Response (CIR) for a given relative position between the receiver and the transmitter. We then vary the distance between the receiver and the transmitter by very small increments and plot the overlaid CIRs for the new positions. As shown in Figure ??, the CIRs are almost identical, with the only difference being a nearly negligible phase shift. The channel taps follow a similar pattern, with the phase shift being even more negligible in the channel taps.

Chapter 4

Experimental Framework

4.1 5G System Architecture

We build on top of one of the existing and widely used open-source cellular network stacks, Open Air Interface [14], to emulate the 5G network and base station to provide connectivity. This means that we utilize well-established platforms that have already been tested and proven effective in real-world applications. OAI provides an RF simulator that can be used to simulate an RF board, typically used to simulate a channel and its impact on the transmitted signal. It performs convolutions on the transmitted signal in real-time as well, which validates with our main idea. However, it has three main limitations: (1) the channels remain constant over time, except for optional constant velocity Doppler shifts, (2) the selection of available channel models is extremely limited, and (3) there is no functionality to insert custom channels.

We solve these problems by opening up an API that can be used to interact with the RF simulator and change the channel model on the fly. The channel models can be loaded from a file, and our code modifications enable the RF simulator to update them regularly. As shown in Figure 4.2b, we simply change how the channel model is being loaded into the RF simulator. Therefore, the benefit of opening up such an API is twofold (1) we can vary the number of taps in the channel model, and (2) we can vary the channel model over time. As shown in Figure 4.2a and 4.2b, the channel model updates at a packet-level granularity, which is a significant improvement over the existing RF simulator. It allows us to test the impact of different channel models on the performance of the network and of the applications running on top of it.

We input various Sionna channels into the system, which are then converted into a format that can be integrated into the RF simulator. The imaginary taps of the channel are read from one .txt file and the real taps are read from another .txt file. The channel update at

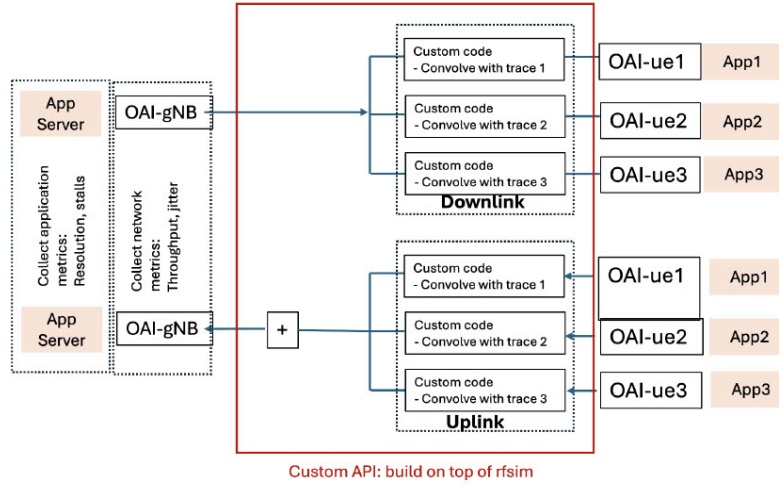


FIGURE 4.1: 5G System Architecture

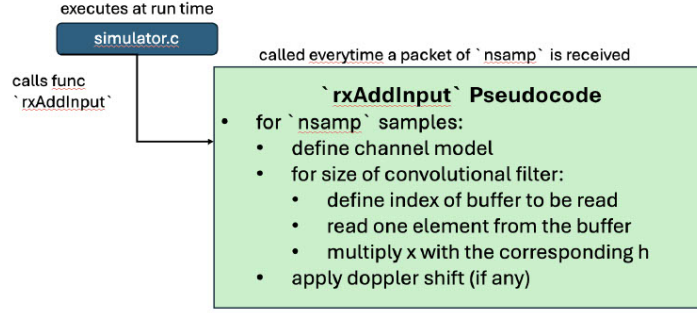
a millisecond timescale for a sampling rate of 23.04 MHz and a carrier bandwidth of 20 MHz. OAI's NR OFDM parameters are according to a numerology of 1, which means that the subcarrier spacing is 15 kHz and the symbol duration is $66.67 \mu\text{s}$.

We run a simple experiment to verify this setup. We input a varying random channel model into the RF simulator and observe the changes in the network parameters. This channel model is implemented in the DL direction, and we run 20 MBps of network traffic via the core network. Provided the system works, we should notice a change in the RSRP values at the UE side, which indicates a change in the taps which are convolved with the constant valued transmitted samples.

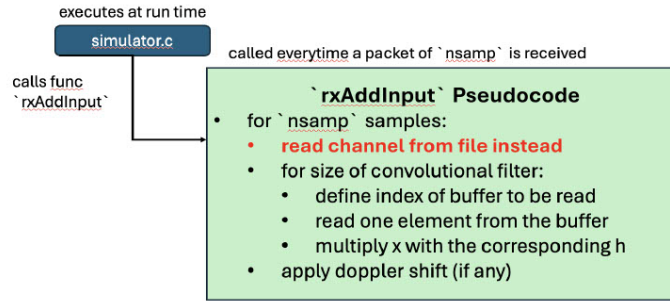
In this setup, we do observe the changes in RSRP values at the UE side, as shown in Figure 4.3. RSRP, or Reference Signal Received Power, is a key metric used to evaluate the performance of the network on the UE side. It can be thought of as the signal strength of the reference signal received at the UE. A change in RSRP values indicates a change in the channel model, which is significantly different from a constant channel model that yields a constant RSRP. This verifies our setup, and validates the capabilities of our framework.

4.2 Channel Simulation Setup

In this section, we describe the simulation setup used to evaluate the performance of various channel representations. To perform our channel simulations, we use Sionna [7], a GPU-accelerated open-source library for link-level simulations based on TensorFlow. Sionna generates attenuation and delay (Time of Flight or ToF) values for transmitter and receiver pairs. This occurs within an open environment model that includes buildings



(A) Existing RF Simulator System in OAI 5G



(B) Modified OAI system with a custom API built on top of RF Simulator

FIGURE 4.2: Comparison of RF Simulator Systems in OAI 5G

and other obstructions of various sizes and dielectric materials. Ray traced tools have been used to generate channel models for various scenarios, such as urban, suburban, and rural environments. The OFDM system used to perform channel simulations in Sionna is shown in Figure 2.2.

We obtain Channel Impulse Responses (CIRs) by utilizing the ray tracing capabilities of Sionna. In Sionna, getting the CIR consists of two main steps: (1) using ray-tracing to find the multi-path from the transmitter to the receiver, and (2) estimating the attenuation of each path using pre-existing 3GPP models. Using these capabilities, we can generate the attenuation a and delay τ models for various scenarios. We derive the channel taps from the CIR of a given scene by employing sinc interpolation on the CIR we obtained from Sionna. Sionna assumes that a sinc filter is used for pulse shaping and receive filtering, a reasonable and industry-accepted assumption. Therefore, given a channel impulse response $(a_m(t), \tau_m), 0 \leq m \leq M - 1$ the channel taps are computed as follows:

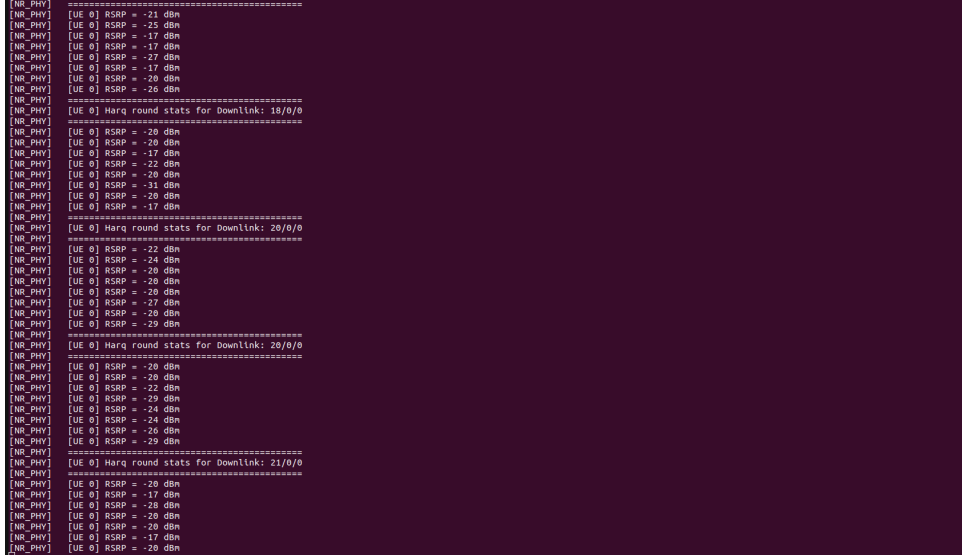


FIGURE 4.3: RSRP Variation with Varying Channel Taps. As you can see each packet received at the UE side has a different RSRP value, indicating a change in the channel representation.

$$\bar{h}_{b,\ell} = \sum_{m=0}^{M-1} a_m \left(\frac{b}{W} \right) \text{sinc}(\ell - W\tau_m) \quad (4.1)$$

Here, a_m and τ_m are the attenuation and delay of the m -th ray, respectively. The channel taps are extracted from the CIR by sampling the sinc function at intervals of $1/W$. This is done by assuming the sampling rate of the channel is W , abiding by the most basic Nyquist rate criterion. ℓ ranges from 0 to $L - 1$, where L is the number of channel taps extracted from the CIR. This process is illustrated in Figure 4.5.

A critical question that arises in this context is: How many channel taps should be extracted from the CIR to accurately represent the channel? The generally accepted answer to this question is straightforward and hinges primarily on the Delay Spread of the

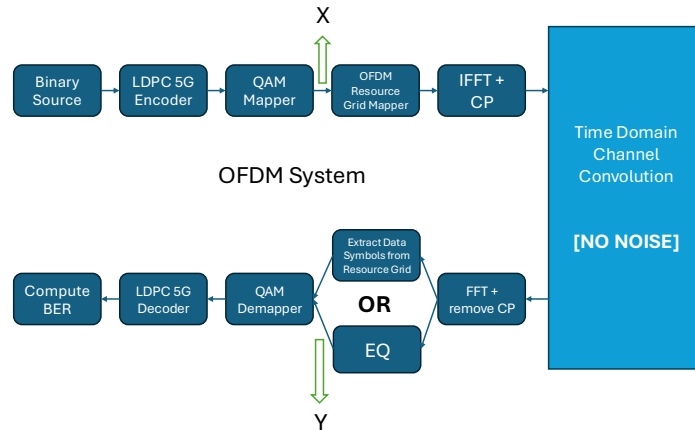


FIGURE 4.4: OFDM System Used to Perform Channel Simulations in Sionna

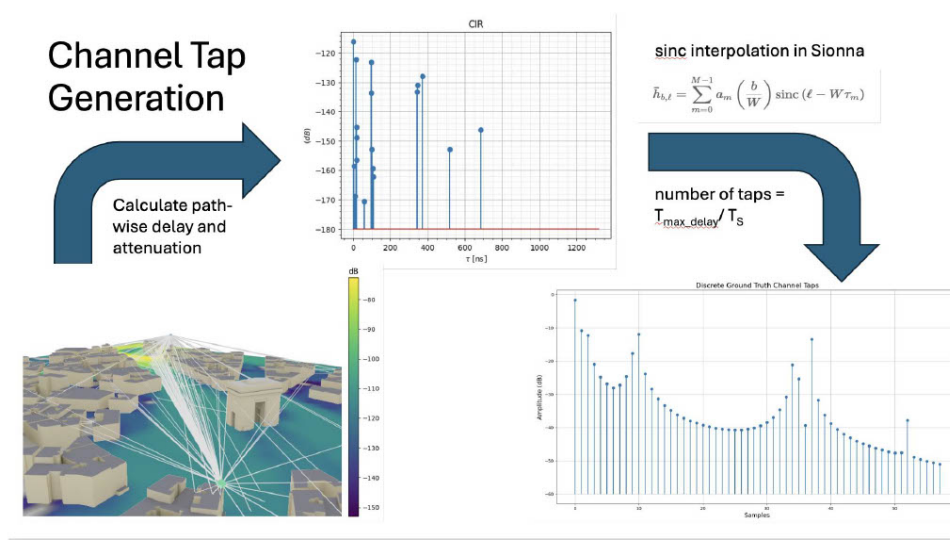


FIGURE 4.5: Channel Tap extraction Pipeline

scene. The Delay Spread indicates the time difference between the arrival of the first and the last transmitted ray from the transmitter to the receiver within a specific environment. By modeling the Delay Spread, we can ascertain the number of channel taps required for an accurate representation of the channel.

$$N = \text{number of taps} = \frac{\text{Delay Spread}}{\text{Sampling Rate}} \quad (4.2)$$

In this equation, the Sampling Rate refers to the frequency at which channel taps are extracted from the CIR. According to the Nyquist criterion, the Sampling Rate must be at least twice the maximum frequency of the channel taps and should also align with the sampling rate of the generated inputs. This ensures that we adequately capture the characteristics of the channel without introducing aliasing effects on the receiver's side.

Chapter 5

Channel Representation Pipeline

We perform a series of experiments to evaluate various channel representations in this section. We propose a pipeline to modify any existing temporal variation of the channel into a form that can be integrated into the above-described RF simulator API. We then evaluate the performance of the proposed pipeline using multiple metrics. The results demonstrate the effectiveness of the proposed pipeline in reducing the number of taps in the channel model without significantly impacting system performance.

5.1 Tap Reduction

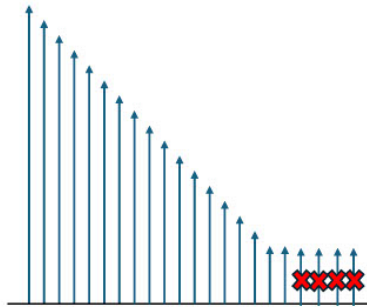


FIGURE 5.1: Visual Representation of Tap Reduction

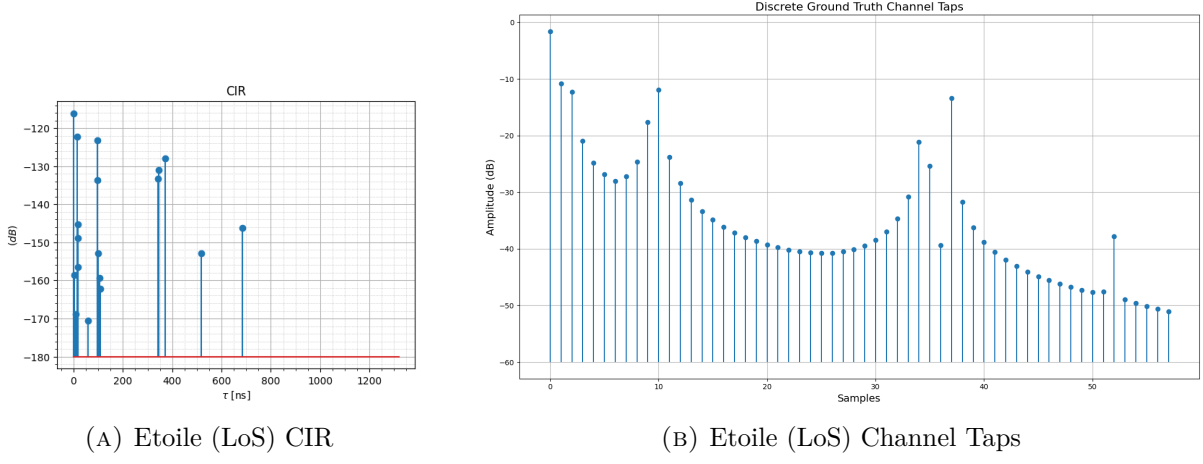


FIGURE 5.2: Etoile (LoS) CIR and Channel Taps

The primary goal of this section, and the experiments performed therein, is to develop a tap reduction algorithm that can effectively **model the multipath nature of the wireless channel in as few taps as possible**. It is common knowledge that wireless channels are characterized by multipath fading, which results in the reception of multiple copies of the transmitted signal at the receiver. An interesting observation you can draw from plotting scenario-wise CIRs in Sionna is that **not all paths contribute equally to the channel**.

This observation is consistent across different scenarios, both in the Line-of-Sight (LoS) cases and in the Non-Line-of-Sight (NLoS) cases. To aid in visualising this, we present two CIRs in Figure 5.2 and Figure 5.3. In both cases, we can observe how there exists a cluster of main paths with a much higher power than the remaining paths.

In the following sections, we further build upon on this rationale.

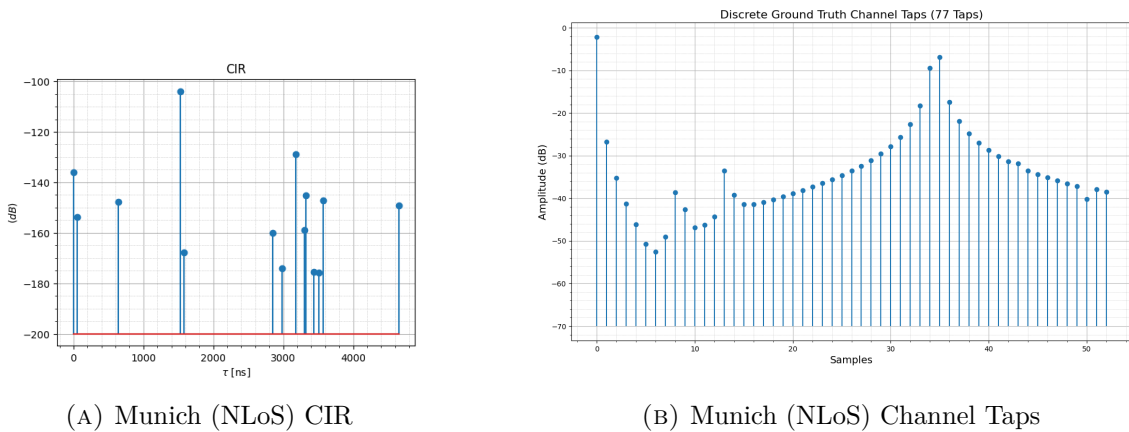


FIGURE 5.3: Munich (NLoS) CIR and Channel Taps

5.1.0.1 How can you classify a path as non-dominant?

For the purpose of this experiment, we assume that a path is non-dominant if its power is 20 dB less than the power of the most dominant path, i.e., the path with the largest tap power. This assumption is based on the fact that a 20 dB difference in power corresponds to a 100-fold difference in power levels, which is highly significant. To better understand this, let us assume we have a two path system, which has two paths: a dominant one and a non-dominant one. If the same 1W sample is transmitted over two paths, such that the dominant path outputs a signal of power 1W, the non-dominant path will output a symbol with a power of 0.01W. With this assumption, we can observe that in both Figure 5.2a and Figure 5.3a, the entire CIR can be reduced to around just 5 strongest taps, as the remaining taps are non-dominant. To err on the side of caution, we will attempt to reduce the corresponding taps from the given CIRs to the 10 strongest taps.

5.1.1 Solution Overview

We develop a simple logically backed algorithm Algorithm 1, which attempts to find the N strongest taps, out of a given set of taps.

Algorithm 1 Sparse Channel Taps Selection

```

procedure SELECTCHANNELTAPS( $h, N$ )
   $M \leftarrow \text{length}(h)$ 
  for  $i \leftarrow 1$  to  $M$  do
     $h_{abs}[i] \leftarrow |h[i]|$ 
  end for
  Find the indices of the  $N$  largest values in  $h_{abs}$  and store them in  $indices$ 
  for  $j \leftarrow 1$  to  $M$  do
    if  $j$  is not in  $indices$  then
       $h_{abs}[j] \leftarrow 0$ 
    end if
  end for
  return  $h$ 
end procedure

```

This algorithm selects the N strongest taps based on their tap wise attenuation values and sets the remaining non-dominant taps to zero. Note that a system with a higher attenuation value represents a poor path, as it results in a lower power signal received at the receiver. Therefore, selecting the N strongest taps is equivalent to selecting the N least attenuated taps in the channel model. This approach has two benefits: (1) it reduces the number of taps in the channel model, making it more computationally efficient, and (2) it allows for a more focused analysis on the dominant paths in the channel.

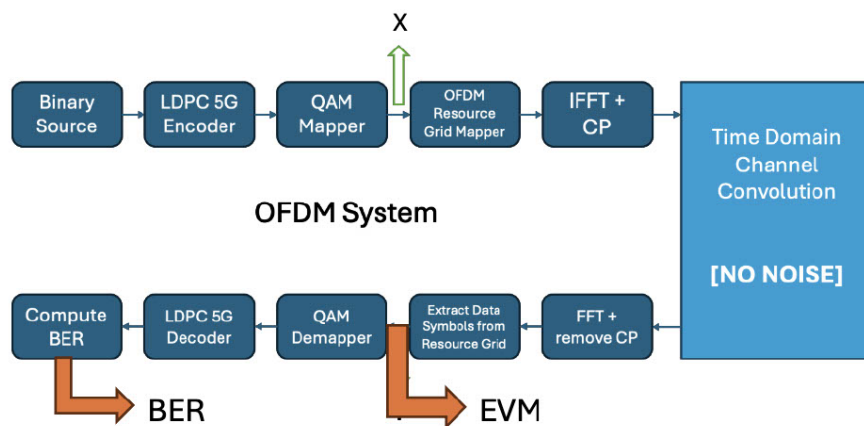


FIGURE 5.4: Evaluation Metrics for Tap Reduction

5.1.2 Metrics-In-Use

We use the following metrics to evaluate the performance of the tap reduction algorithm:

5.1.2.1 Error Vector Magnitude (EVM)

Error Vector Magnitude (EVM) is a popular system-level performance metric defined in many communication standards, such as Wireless Local Area Networks (WLAN 802.11), mobile communications (4G LTE, 5G), and many more, as a compliance test. Beyond compliance, EVM is an extremely useful system-level metric to quantify the combined impact of all potential impairments in a system through a single and easy-to-understand value.

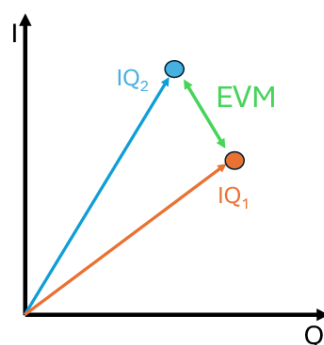


FIGURE 5.5: What is EVM?

The root mean square (rms) of all error vector magnitudes between the received symbol locations and their closest ideal constellation locations constitutes the EVM value of the device. Modern communication standards prescribe a minimum acceptable EVM level based on the transmitted or received signal characteristics, such as data rate and bandwidth.

In our case, the EVM represents the error between the ground truth symbols and the reduced tap symbols:

$$\text{EVM (dB)} = 20 \cdot \log_{10} \left(\sqrt{\frac{1}{N} \sum_{n=0}^{N-1} \left((I_{\text{actual}}[n] - I_{\text{reduced}}[n])^2 + (Q_{\text{actual}}[n] - Q_{\text{reduced}}[n])^2 \right)} \right) \quad (5.1)$$

- **Actual:** Refers to the OFDM symbols after convolution with the ground truth channel.
- **Reduced:** Refers to the OFDM symbols after convolution with a lower/reduced tap resolution of the ground truth channel.

5.1.2.2 Bit Error Rate (BER)

Bit Error Rate (BER) is a measure of the number of bit errors in the received signal compared to the transmitted signal. A lower BER indicates more reliable signal reception. BER is considered a key performance metric in digital communication systems, as it provides a higher-level view of how system changes affect the system's bottom line.

BER is calculated using the following formula:

$$\text{BER} = \frac{\text{Number of bit errors}}{\text{Total number of transmitted bits}} \quad (5.2)$$

Several factors can influence BER, including:

- **Signal-to-Noise Ratio (SNR):** Higher SNR typically results in a lower BER.
- **Modulation Scheme:** Different modulation schemes have varying levels of robustness against noise and interference.
- **Channel Representation:** Multipath fading, number of taps, and other channel impairments can cause variations in BER.

In practical systems, achieving a low BER is crucial for maintaining high data integrity and ensuring efficient communication. BER testing is often performed under various conditions to evaluate the performance of communication systems and to identify potential areas for improvement.

5.1.3 Experiments Performed

The analysis of tap variation across different scenarios reveals distinct patterns for both Non-Line of Sight (NLoS) and Line of Sight (LoS) conditions in various modulation schemes. Sionna's Ray-Tracing library comes pre-loaded with a variety of cities, each with completely different environments. Therefore, to perform our experiments, we use two completely different scenarios:

- **Munich** - We select a NLoS-dominant scenario with 93 ground truth taps
- **Etoile** - We select a LoS-dominant scenario with 58 ground truth taps

We perform our experiments on these two scenarios but across four different modulation schemes: (1) QPSK, (2) 16 QAM, (3) 64 QAM and (4) 256 QAM. We use these four

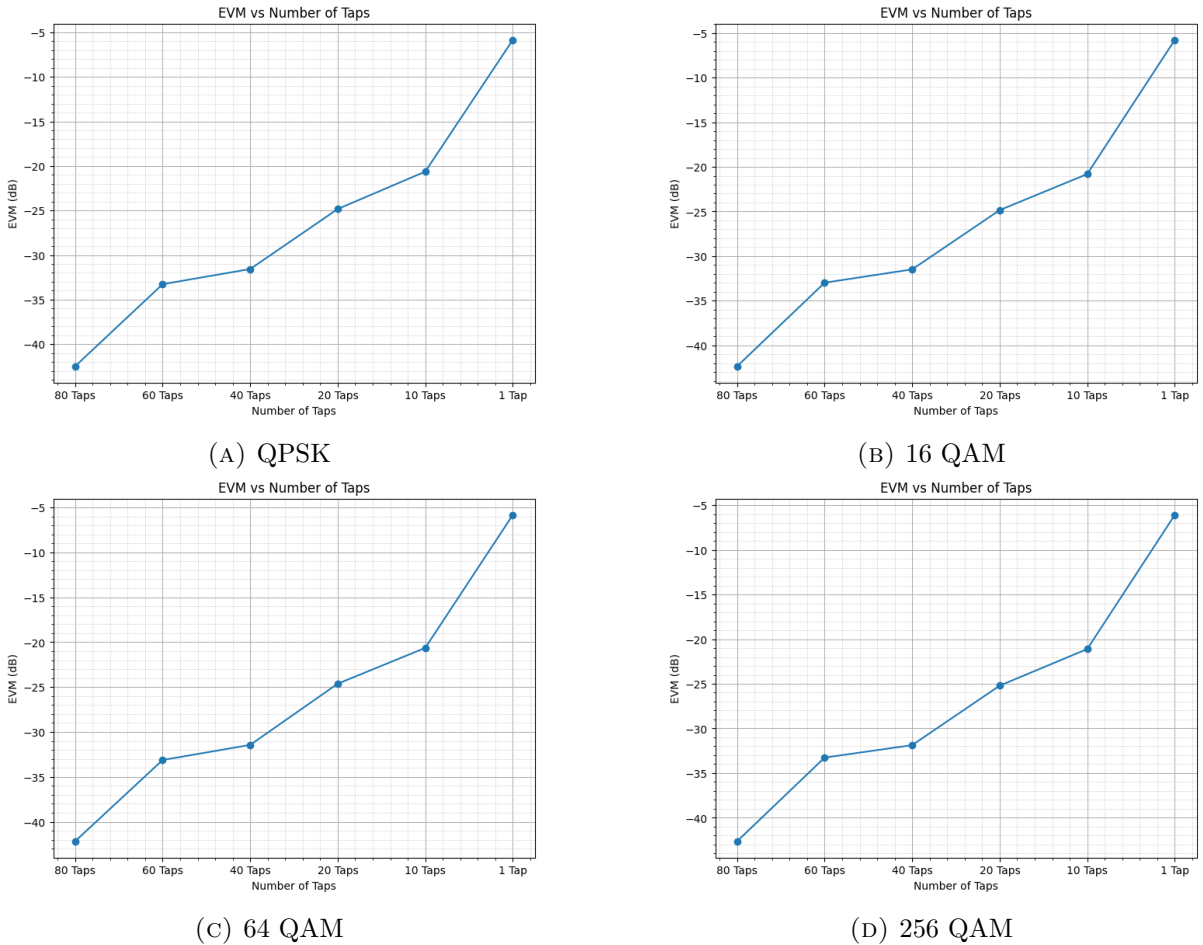


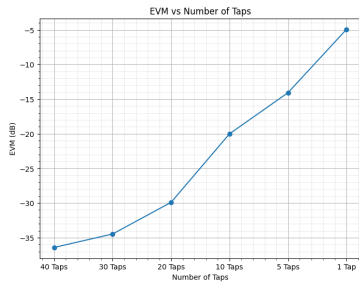
FIGURE 5.6: Comparison of Munich scenario's EVM across different modulation schemes

modulation schemes because they are the ones most commonly used in 5G NR systems. We employ a constant coding rate k of 0.5 for all modulation schemes to maintain consistency across the experiments.

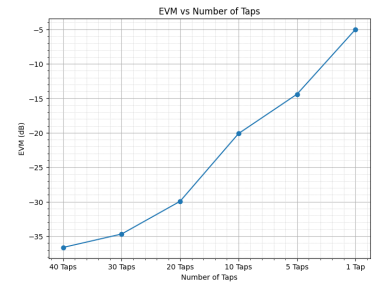
In all our experiments, we vary the number of taps from the ground truth taps (58 for Etoile and 93 for Munich) down to 10, and evaluate the EVM and BER for each scenario and modulation scheme. We vary the taps using the tap reduction algorithm described in Algorithm 1. Our goal is to determine the impact of tap reduction on system performance, and to identify the optimal number of taps that can be used to accurately represent the channel model.

In the NLoS Munich scene (see Figure 5.6), decreasing the number of taps consistently leads to an increase in the EVM power across all modulation schemes. Similarly, the LoS Etoile scene (see Figure 5.7) shows an almost identical EVM vs. number of taps relationship, allowing us to generalize that the algorithm works across both LoS and NLoS paths.

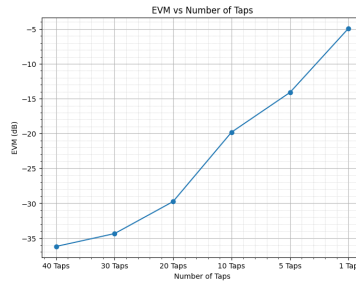
Furthermore, the variation in BER for the Etoile scene (similar to as we did for BER) in 5.9 reveals that reducing tap resolution from 58 to 10 only results in approximately a 1% change in BER, indicating that even drastic reductions in tap count has minimal negative impacts on performance. This trend is consistent across different modulation schemes, with BER remaining relatively stable even with significant reductions in tap count. Overall, these findings reinforce the significance of tap selection strategies in optimizing communication system performance across diverse scenarios. Our takeaway



(A) QPSK



(B) 16 QAM



(C) 64 QAM

FIGURE 5.7: Comparison of Etoile scenario's EVM across different modulation schemes

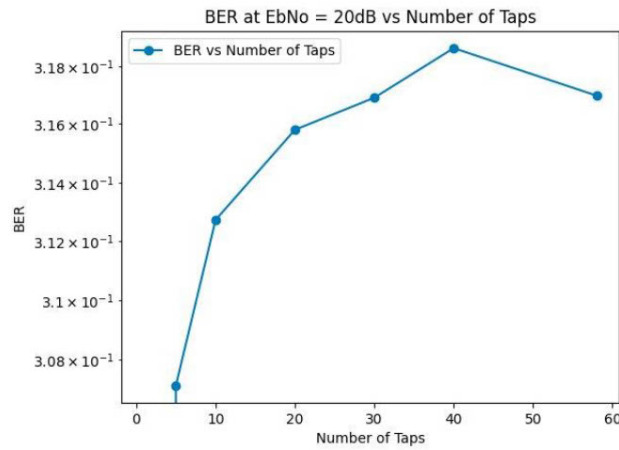


FIGURE 5.8: BER vs. Tap Resolution

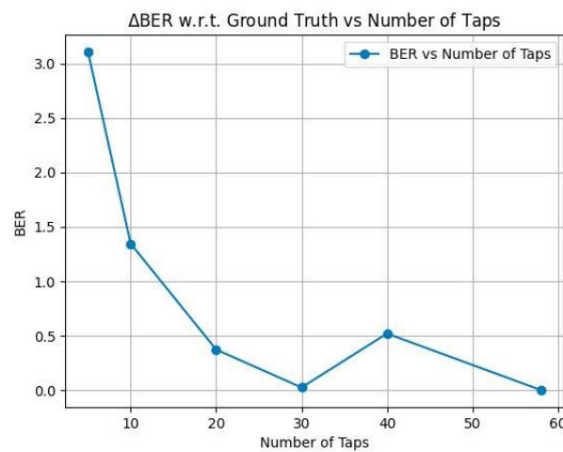


FIGURE 5.9: Percentage Change in BER vs. Tap Resolution

from this experiment is that the tap reduction algorithm can effectively reduce the number of taps in the channel model without significantly impacting system performance.

5.2 Tap Updation

The primary goal of this experiment is to develop a tap updation algorithm that can empirically determine the coherence times of a wireless channel using only the temporal variation of CIRs. We do so by identifying breakpoints in a set of varying channel models, which represent significant changes in the channel characteristics.

5.2.0.1 What is a breakpoint?

The coherence time of a wireless channel is typically defined as the duration over which the channel's characteristics, particularly phase shift and attenuation, remain constant. Therefore, finding the coherence time of a channel is equivalent to identifying the points in time where the channel characteristics change significantly. Each breakpoint corresponds to a significant change in the channel model and is considered a point where the current channel tap would not accurately represent the system. The challenge is to empirically determine these breakpoints for the entire time frame using very limited scenario context.

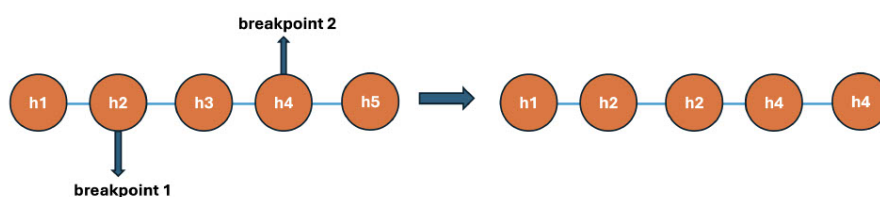


FIGURE 5.10: *Visualization of Breakpoint Selection:* The algorithm selects the breakpoints where the channel model changes significantly. The distances between breakpoints can be assumed to be the empirically determined coherence times of the channel.

We obtain a set of channel taps from Sionna over a fixed duration as a series of different CIRs. We then compare these time-stamped CIRs and channel taps to apply the algorithm and determine the breakpoints in the channel model.

5.2.1 Solution Overview

We develop a simple logically-backed algorithm, Algorithm 2, which attempts to find the system breakpoints in the channel model. The question now arises: *How do we determine the system breakpoints in the channel representation?* We modify this question to: *How do we determine significant changes in the channel representation over time?*

We attempt to do this by using the metrics defined in the next section in two main ways:

- By comparing how the **channel taps deviate** from each other over time.
- By comparing how the **OFDM output changes** as we vary the channel models in the system.

Algorithm 2 Finds the system breakpoints

```

1: function PROCESSTAPS( $\tau$ )
2:    $[H, t] \leftarrow \text{getTaps}()$ 
3:    $N \leftarrow \text{length}(H)$ 
4:    $H' \leftarrow$  initialized as an empty array
5:    $t' \leftarrow$  initialized as an empty array
6:   for  $i = 0$  to  $N - 1$  do
7:      $h_i \leftarrow H[i]$ 
8:     if  $i > 0$  then
9:        $\text{deviation} \leftarrow \text{METRIC}(h_{i-1}, h_i)$ 
10:      if  $\text{deviation} < \tau$  then
11:         $\text{val} \leftarrow h_{i-1}$ 
12:      else
13:         $\text{val} \leftarrow h_i$ 
14:         $\text{last\_change} \leftarrow i$ 
15:      end if
16:    end if
17:    if  $\text{val} \notin H'$  then
18:       $H'.\text{append}(\text{val})$ 
19:    end if
20:    if  $\text{last\_change} \notin t'$  then
21:       $t'.\text{append}(\text{last\_change})$ 
22:    end if
23:  end for
24:  return  $H', t'$ 
25: end function

```

5.2.2 Metrics-In-Use

According to Molisch [12], coherence time can be defined as the period over which the channel's time correlation function changes by 3 dB. While this definition, doesn't apply as generally as we would like, it does provide a good starting point for our analysis. Building on this concept, we propose an approach that characterizes channel variations using multiple lower-level metrics. Our goal is to establish a threshold that can effectively determine the channel's coherence time based on the Bit Error Rate (BER). We can categorise these lower-level metrics into two main categories, based on what they are measuring. These concepts are discussed below.

5.2.2.1 Channel Tap Deviation

Metrics of this category are purely concerned with how the channel representations vary from each other. For two channel representations h_1 and h_2 , we can define the following metrics:

1. **Earth Mover's Distance (EMD)**: Measures the minimum amount of work required to transform one discrete distribution into another.

$$\text{EMD}(h_1, h_2) = \sum_{i=1}^n |\text{CDF}_{h_1}(i) - \text{CDF}_{h_2}(i)| \quad (5.3)$$

2. **Euclidean Distance**: Quantifies the straight-line distance between two points in the channel space.

$$d_e(h_1, h_2) = \sqrt{\sum_{i=1}^n (h_1(i) - h_2(i))^2} \quad (5.4)$$

3. **Manhattan Distance**: Measures the distance between two points along axes at right angles (also known as L1 distance).

$$d_m(h_1, h_2) = \sum_{i=1}^n |h_1(i) - h_2(i)| \quad (5.5)$$

4. **Mean Squared Error (MSE)**: Calculates the average of the squares of the differences between the estimated and actual values.

$$\text{MSE}(h_1, h_2) = \frac{1}{n} \sum_{i=1}^n (h_1(i) - h_2(i))^2 \quad (5.6)$$

5. **Mean Channel Power**: Computes the average power per tap in the channel.

$$P_{\text{mean}}(h) = \frac{1}{n} \sum_{i=1}^n |h_i|^2 \quad (5.7)$$

5.2.2.2 OFDM Output Deviation

We again use EVM and BER in the exact same way we defined in 5.4. Another extra component we use in this section is the **Mean Symbol Phase Shift**.

This metric becomes relevant here in particular because one of the major things we attempt to model is the Doppler effect of the scene. The Mean Symbol Phase Shift is defined as the average phase shift imparted by the channel over each input OFDM symbol. In a constant Doppler scenario, a constant phase shift is applied across each of the symbols convolved by the channel. Doppler is determined by the relative motion between the transmitter and the receiver, and thus varies with changes in their relative velocity. Therefore, by measuring the change in phase shift, one can determine the change in Doppler, and consequently, the change in the channel model.

For an input of N OFDM symbols x and the corresponding stream of output symbols y , we can calculate the phase shift as below:

$$\phi = |\text{Imag}(x \cdot \text{conj}(y))| \quad (5.8)$$

Here, conj yields the conjugate of the output OFDM symbols. We then calculate the average rate of phase shift $\Delta\phi_{\text{avg}}$ as:

$$\Delta\phi_{\text{avg}} = \frac{1}{N-1} \sum_{i=1}^{N-1} (\phi[i] - \phi[i-1]) \quad (5.9)$$

5.2.3 Experiments Performed

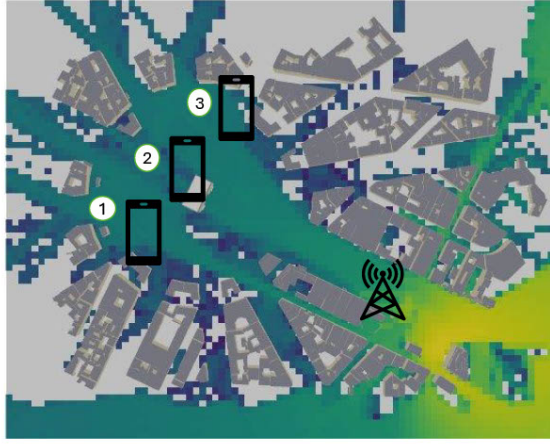


FIGURE 5.11: Mobility Scenario in Sionna

To perform our analysis, we use the same the Munich scene as defined in the previous sections. However, we now focus on the temporal variation of the channel model, rather than the tap resolution. We create a simple dataset of various channel taps for a mobility scenario in Sionna, where the receiver moves at a constant velocity.

The RX position changes from an initial position of $[-300, 200, 25]$ to a final position of $[-290, 200, 25]$ at a constant speed. This constant speed is simulated by constant spatial changes of the receiver with respect to the transmitter. The CIR and therefore channel taps are re-plotted at every significant position change, which is defined as every 0.1 meters.

We function at such a fine-grained level because the carrier frequency of the system is 2.149 GHz, which corresponds to a wavelength of around 14 cm. Therefore, a change in position of 0.1 meters corresponds to a change in phase of around 0.7 radians, which is significant enough to cause a change in the channel model.

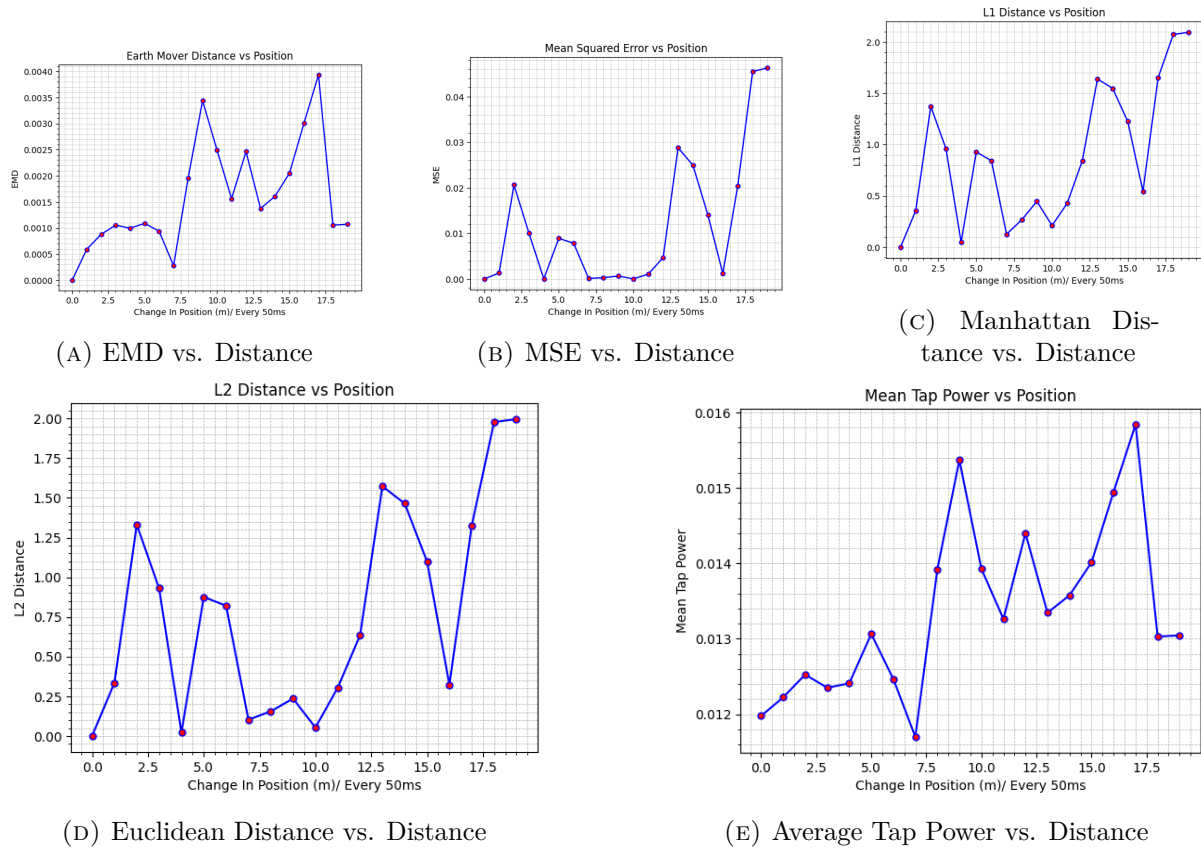


FIGURE 5.12: Channel Tap Deviation Metrics vs. Distance

We observe that the EMD, MSE, L1, and L2 distances all exhibit a similar pattern across the range of our selected time frame. To better understand the significance of the change in these metrics, we need to plot the Bit Error Rate (BER). That will be covered in future work.

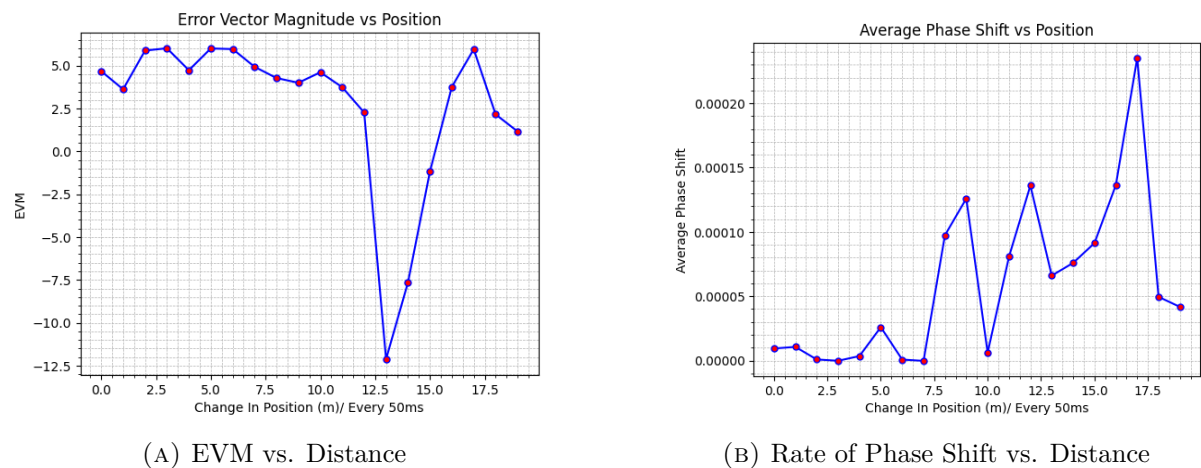


FIGURE 5.13: Comparison of OFDM Outputs

Chapter 6

Conclusion

The objective of this study was to take the initial step towards understanding how to construct a real-world cellular digital twin, or a *"full twin"*. Our efforts can be characterized into two main parts: **(1) the wireless efforts**, which involved understanding the most significant contributions of the wireless channel's multipath and how it changes over time, and **(2) the systems efforts**, which focused on building the digital twin on top of a pre-existing, open-source RAN stack. The PHY-level hypotheses were tested using Sionna [7], a ray-traced wireless channel simulator. Additionally, a framework to input these channel taps was developed on top of the 5G version of Open Air Interface [14].

By the end of this study, we developed a comprehensive channel pipeline to transform a real-world, temporally varying wireless channel into a format that can be used as input for our newly built OAI framework. The channel pipeline was tested across more than 100 different transmission scenarios and was found to generalize across various modulation schemes, LoS/NLoS conditions, and different coding rates. Similarly, we are also able to use various metrics to map out significant changes in the channel model and its output on RAN-level and application-level metrics. These results are significant as they provide a foundational understanding of how wireless channel variations impacts the RAN and application performance.

Future work on this topic involves testing the channel pipeline on a real-world setup and further refining the system framework to include more advanced features like MIMO and sub-nanosecond channel updates. We also aim to conduct end-to-end application benchmarks on the digital twin to assess the impact of the wireless channel on the application layer.

Bibliography

- [1] Jeffrey G. Andrews et al. “What Will 5G Be?” In: *IEEE Journal on Selected Areas in Communications* 32.6 (2014), pp. 1065–1082. DOI: 10.1109/JSAC.2014.2328098.
- [2] Mehdi Bennis, Mérouane Debbah, and H. Vincent Poor. “Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale”. In: *Proceedings of the IEEE* 106.10 (2018), pp. 1834–1853. DOI: 10.1109/JPROC.2018.2867029.
- [3] Leonardo Bonati et al. “Colosseum: Large-Scale Wireless Experimentation Through Hardware-in-the-Loop Network Emulation”. In: *Proc. of IEEE Intl. Symp. on Dynamic Spectrum Access Networks (DySPAN)*. Virtual Conference, 2021.
- [4] Saumya Borwankar and Dhruv Shah. “Low Density Parity Check Code (LDPC Codes) Overview”. In: *CoRR* abs/2009.08645 (2020). arXiv: 2009.08645. URL: <https://arxiv.org/abs/2009.08645>.
- [5] Joe Breen et al. “Mobile and wireless research on the POWDER platform”. In: *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services. MobiSys ’21*. Virtual Event, Wisconsin: Association for Computing Machinery, 2021, 509–510. ISBN: 9781450384438. DOI: 10.1145/3458864.3466915. URL: <https://doi.org/10.1145/3458864.3466915>.
- [6] J.M. Cioffi et al. “MMSE decision-feedback equalizers and coding. I. Equalization results”. In: *IEEE Transactions on Communications* 43.10 (1995), pp. 2582–2594. DOI: 10.1109/26.469441.
- [7] Jakob Hoydis et al. “Sionna: An Open-Source Library for Next-Generation Physical Layer Research”. In: *arXiv preprint* (2022).
- [8] Adam Langley et al. “The QUIC Transport Protocol: Design and Internet-Scale Deployment”. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication. SIGCOMM ’17*. Los Angeles, CA, USA: Association for Computing Machinery, 2017, 183–196. ISBN: 9781450346535. DOI: 10.1145/3098822.3098842. URL: <https://doi.org/10.1145/3098822.3098842>.
- [9] Jia-Chin Lin. “Least-Squares Channel Estimation for Mobile OFDM Communication on Time-Varying Frequency-Selective Fading Channels”. In: *IEEE Transactions on Vehicular Technology* 57.6 (2008), pp. 3538–3550. DOI: 10.1109/TVT.2008.919611.

- [10] Xingqin Lin et al. *6G Digital Twin Networks: From Theory to Practice*. 2022. arXiv: 2212.02032 [cs.NI]. URL: <https://arxiv.org/abs/2212.02032>.
- [11] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. “Neural Adaptive Video Streaming with Pensieve”. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM ’17. Los Angeles, CA, USA: Association for Computing Machinery, 2017, 197–210. ISBN: 9781450346535. DOI: 10.1145/3098822.3098843. URL: <https://doi.org/10.1145/3098822.3098843>.
- [12] Andreas F. Molisch. *Wireless Communications*. 2nd. Wiley Publishing, 2011. ISBN: 0470741864.
- [13] Ravi Netravali et al. “Mahimahi: accurate record-and-replay for HTTP”. In: *Proceedings of the 2015 USENIX Conference on Usenix Annual Technical Conference*. USENIX ATC ’15. Santa Clara, CA: USENIX Association, 2015, 417–429. ISBN: 9781931971225.
- [14] Navid Nikaein et al. “OpenAirInterface: A Flexible Platform for 5G Research”. In: *SIGCOMM Comput. Commun. Rev.* 44.5 (Oct. 2014), 33–38. ISSN: 0146-4833. DOI: 10.1145/2677046.2677053. URL: <https://doi.org/10.1145/2677046.2677053>.
- [15] Michele Polese et al. “ColO-RAN: Developing Machine Learning-based xApps for Open RAN Closed-loop Control on Programmable Experimental Platforms”. In: *CoRR* abs/2112.09559 (2021). arXiv: 2112.09559. URL: <https://arxiv.org/abs/2112.09559>.
- [16] Theodore S Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 1996.
- [17] Mathew K Samimi and Theodore S Rappaport. “Local multipath model parameters for generating 5G millimeter-wave 3GPP-like channel impulse response”. In: *2016 10th European Conference on Antennas and Propagation (EuCAP)* (2016), pp. 1–5.
- [18] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
- [19] Klaus Wehrle, Mesut Günes, and James Gross, eds. *Modeling and Tools for Network Simulation*. Springer, 2010. ISBN: 978-3-642-12330-6. DOI: 10.1007/978-3-642-12331-3. URL: <https://doi.org/10.1007/978-3-642-12331-3>.