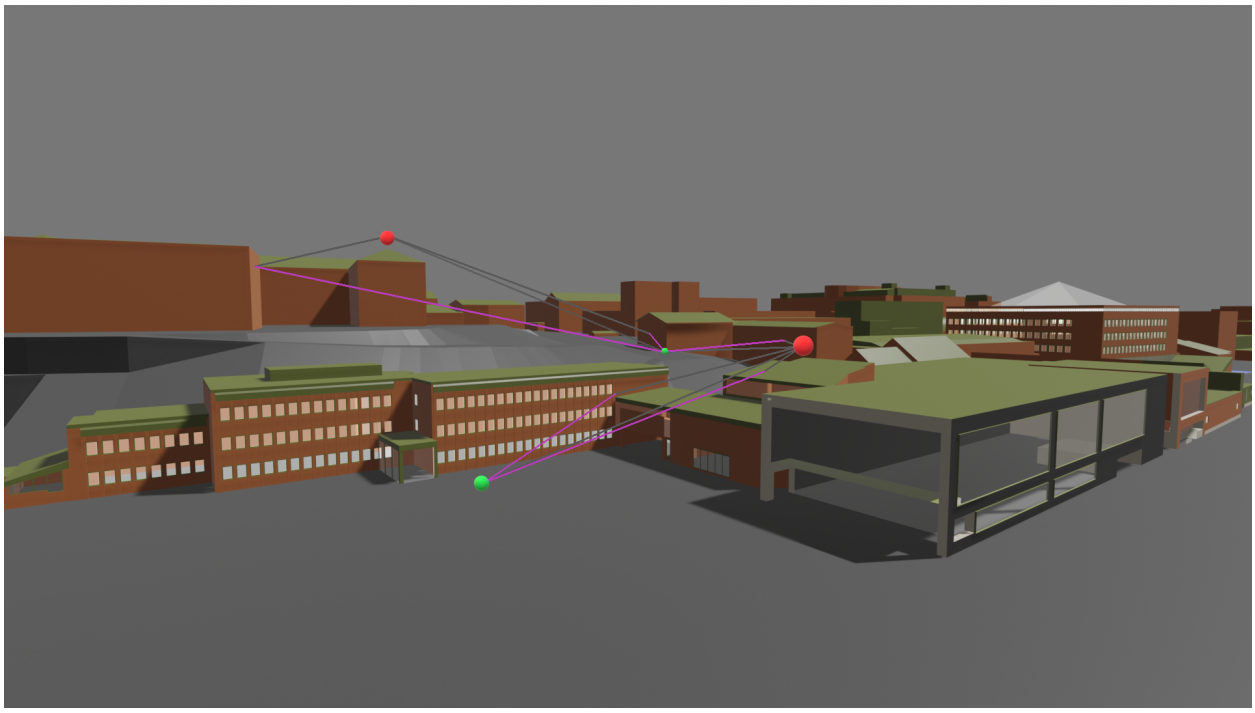
CHALMERS
UNIVERSITY OF TECHNOLOGY



Digital Radio Twin of Chalmers for 6G Integrated Sensing, Positioning, and Communication

Bachelor's Thesis in Communications, Antennas, and Optical Networks

Degree project report in Electrical/Biomedical engineering

Malte Rieglert, Abdur Arshad, Abdirahman Jamal Abdi, Antony Hamouche, Jesper Damgaard, Ramtin Rafiei Taghanaki

Departement of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Bachelor's Thesis 2025

Digital Radio Twin of Chalmers for 6G Integrated Sensing, Positioning, and Communication

Bachelor's Thesis in Communications, Antennas and Optical Networks

Malte Rieglert, Abdur Arshad, Abdirahman Jamal Abdi, Antony Hamouche, Jesper Damgaard, Ramtin Rafiei Taghanaki

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Digital Radio Twin of Chalmers for 6G Integrated Sensing, Positioning, and Communication
Bachelor's Thesis in Communications, Antennas and Optical Networks
Malte Rieglert, Abdur Arshad, Abdirahman Jamal Abdi, Antony Hamouche, Jesper Damgaard, Ramtin Rafiei Taghanaki

Supervisor: Liping Bai, Sharief Saleh, Electrical Engineering
Examiner: Henk Wymeersch, Electrical Engineering

Bachelor's Thesis 2025
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden

Cover: Digital radio twin of Chalmers University of Technology.

Abstract

This thesis presents the development of a Digital Radio Twin of Chalmers University campus, aimed at enabling advanced simulation and evaluation of 6G wireless communication systems. As future cellular technologies increasingly demand precise modeling of radio environments, this project integrates 3D modeling, ray tracing, and signal processing to create a simulation framework that reflects realistic propagation conditions. Using Blender for geometric modeling and NVIDIA's Sionna RT for ray-tracing-based channel simulations, a virtual replica of the Chalmers campus was constructed. This digital environment supports configurable transmitter and receiver setups, allowing systematic analysis of signal behavior under various parameters. The generated data was then used to compute key performance indicators (KPIs) such as channel capacity, latency, and positioning accuracy. Despite time and scope constraints, this approach demonstrates the feasibility and value of digital radio twins in exploring and designing future 6G networks. The resulting datasets and simulation tools offer a valuable foundation for further research in integrated sensing, positioning, and communication. Analyzing the simulation results provides insight into how different performance metrics affect the transmitted signal in terms of both positioning and communication. Parameters such as the number of subcarriers, orthogonal frequency-division multiplexing (OFDM) symbols, transmission power, and distance, all seem to impact the performance metrics significantly. For positioning, it became surprisingly evident that distance was not the only contributing factor in achieving low positioning bounds; the system's resolution also seemed to play a significant role.

Keywords: Digital Twin, Digital Radio Twin, 6G Communication, 6G, 5G, Ray Tracing, Wireless Propagation, Sionna RT, Sionna, Channel Modeling, Positioning, Blender 3D Modeling, Key Performance Indicators (KPIs), Integrated Sensing and Communication (ISAC).

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in alphabetical order:

AoA angle of arrival
AoD angle of departure
AWGN additive white Gaussian noise
BS base station
BW bandwidth
CFR channel frequency response
CIR channel impulse response
CRLB Cramér-Rao lower bound
FIM Fisher information matrix
GCS global coordinate system
3GPP 3rd generation partnership project
KPI key performance indicator
LCS local coordinate system
LoS line-of-sight
NF noise factor
NFig noise figure
NLoS none-line-of-sight
NPSD noise power spectral density
OFDM orthogonal frequency-division multiplexing
OSM OpenStreetMap
PEB position error bound
QPSK quadrature phase shift keying
RX receiver
SCS sub-carrier spacing
SNR signal-to-noise ratio
TX transmitter
UE user equipment

Contents

List of Figures

List of Tables

1
Introduction

Cellular communication has evolved from the ability to perform basic voice transmission over short distances to the global exchange of diverse forms of information. Throughout the evolution of cellular communication, each generation of cellular technology has introduced substantial advancements, leading to improved transmission speeds, increased network capacity, and enhanced overall functionality [1]. The most recent generation, 5G, was developed to overcome the limitations of its predecessors and is currently being deployed across various sectors of society. Concurrently, research and development efforts toward the next generation, 6G, are underway [2]. A critical aspect of designing future cellular systems lies in the optimal configuration of radio parameters, which determine the fundamental parameters and properties of both the transmitting and receiving systems (Base Stations (BSs) and User Equipments (UEs), respectively). These parameters include, but are not limited to, carrier frequency, bandwidth, antenna count, and antenna type.

The project's objective is to enable the evaluation of such communication parameters using a Digital Radio Twin of the Chalmers campus, to support the development and implementation of future generations of cellular communication systems. This objective is achieved by designing a Digital Twin of Chalmers that can be integrated into a digital simulation environment to serve as a Digital Radio Twin. The Digital Radio Twin can then be used to generate data on radio wave propagation under different parameter settings and configurations.

The generated data will be used throughout the project to evaluate positioning and communication KPIs. Additionally, the effects and implications of different sets of parameters on the KPIs will be analyzed and discussed in detail in the concluding sections of the report.

Furthermore, the developed Digital Radio Twin, along with the generated data, is intended to serve as a valuable resource for researchers in the field of communication systems. It is expected to facilitate the development and optimization of next-generation cellular technologies, before physical deployment, by eliminating the need for manual configuration and measurement processes. It also enables more complex and realistic simulation scenarios.

1.1 Background

Wireless communication is defined as the transfer of data without cables, wires, or any physical connections [3]. In telecommunication, data is transmitted to its target through a medium in the form of electromagnetic waves over a frequency range of 3000 GHz or less [4], depending on the use case. Wireless communication began in the late 19th century, initially by transmitting Morse code, which later translated into the invention of mobile devices [5]. The first handheld mobile phone was introduced in 1973 and weighed approximately 2 kilograms. Since then, mobile devices have developed drastically and become an integral part of modern life.

Evolution of Cellular Generations

A major driver of the development of wireless communication systems has been the evolution of cellular communication. This evolution has spanned multiple generations, each introducing improvements in transmission speed, network capacity, and functionality. The first generation (1G) marked the beginning of mobile telephony, enabling analog voice communication over relatively short distances [1]. It operated in frequency bands around 800 MHz and supported data transfer rates of approximately 2.4 Kbps. The second generation (2G) introduced digital transmission and was the first to support services such as Short Message Service (SMS). 2G operated primarily in the 900 MHz frequency band and allowed data rates of up to 64 Kbps. The third generation (3G) further enhanced mobile capabilities by providing access to the World Wide Web and multimedia services. It operated in the 2.1 GHz band and enabled higher data throughput compared to its predecessors. The fourth generation (4G) significantly improved speed and functionality, supporting advanced applications such as high-definition video streaming, online gaming, and **VR!** (**VR!**). The latest generation, 5G, operates across two main frequency ranges: sub-6 GHz, also known as Frequency Range 1 (FR1), and millimeter-wave bands (24–71 GHz), referred to as Frequency Range 2 (FR2). 5G offers peak data rates of up to 10 Gbps and latencies as low as 1 millisecond. Additionally, since the release of 3rd generation partnership project (3GPP) Release 16, 5G has incorporated positioning capabilities, which, in theory, enable high-accuracy localization of devices receiving 5G signals [6].

Frequency Ranges and System Design Implications

Wireless communication systems across generations utilize specific frequency ranges, influencing system performance characteristics. Higher frequencies generally experience greater path loss, necessitating more infrastructure to maintain coverage [7]. However, they offer wider bandwidths, enabling higher data rates, and allow for the deployment of more antenna elements within the same physical space, enhancing techniques like beamforming and spatial multiplexing. The selection of frequency bands for newer generations is influenced by technical capabilities and spectrum allocation policies. While some frequency bands are already in use, regulatory harmonization and refarming allow certain frequencies to be reused or shared across generations.

Overview and Expectations for 6G

6G is the upcoming sixth generation of cellular communication technology. Though not yet deployed, it is currently a central focus of research and development. Standardization is expected to be initiated by organizations such as 3GPP, with commercial deployment projected to begin around 2030 [8]. A key distinction of 6G compared to earlier generations is its anticipated operation in new radio frequency bands, including sub-terahertz (100–300 GHz) and frequency range 3 (FR3), which spans from 7 to 24 GHz [9]. It is also expected to reuse frequencies of past generations, such as 5G's FR1 and FR2.

6G aims to deliver signal transmission speeds of up to 1 Tbps and latencies in the order of microseconds, which noticeably surpass the capacity of 5G [2]. Furthermore, 6G is anticipated to improve positioning and introduce new sensing capabilities. These advancements may enable higher accuracy estimation of the device locations, orientations, and velocities, while also facilitating the detection and tracking of surrounding objects and individuals, including the assessment of their distance, size, shape, and materialistic properties [10]. While 6G remains in the early stages of development and these projections are not yet definitive, technical studies are already underway. Standardization groups such as 3GPP are expected to release initial technical specifications for 6G by the end of 2028 [8].

Once becoming publicly available, 6G technology is expected to transform several domains, including threat detection, health monitoring, and decision support for law enforcement [2]. Moreover, its anticipated high capacity, fast data rates, and low latency could enhance the performance of existing technologies such as remote surgery systems, autonomous vehicles, and vehicular safety, as well as applications in virtual reality (VR) and augmented reality (AR).

Importance of Simulation and the Role of Academia

Since 6G standards are still under development, there remain significant opportunities for research to shape the future of this technology. The quality and realism of simulation scenarios and measurements are crucial, as they directly affect the accuracy of propagation data. Realistic emulations of real-world environments are essential for telecommunication companies in the early phases of 6G implementation, especially for tasks such as BS planning and network resource management. Accurate simulations and detailed data analysis support informed decision-making, enabling optimized performance and resource allocation.

Institutions like Chalmers University can play a vital role in this evolving research landscape. By conducting high-quality simulations and analysis, Chalmers can contribute valuable insights that inform the development and deployment strategies of telecom operators and equipment manufacturers. This collaborative effort could advance academic knowledge and support the practical implementation of 6G technologies. Potential applications of 6G, such as integrated sensing and communication (ISAC), could revolutionize various sectors by enabling new technologies and enhancing existing systems. By actively participating in 6G research, Chalmers can play an important role in driving innovation and ensuring that emerging technologies are effectively integrated into future networks.

1.2 Purpose

The purpose of this project is to enable the evaluation of various radio configurations and parameters in a virtual simulation environment, with the goal of optimizing communication and positioning services on the Chalmers campus. Additionally, the project aims to provide insights into how different configuration choices influence the characteristics of the simulated radio channel data and how they in turn influence different positioning and communication KPIs. The developed simulation pipeline and the resulting datasets may support future research and development efforts related to 6G networks.

1.3 Scope

Due to time constraints and the group's limited prior experience in wireless communication and 3D simulation technologies, it was necessary to define clear boundaries to ensure the successful and timely completion of the project. These constraints influenced decisions regarding the level of detail in the modeling process, the extent of the simulation environment, the selection of parameters and metrics used for data generation and evaluation, and the complexity of the channel modeling for computations. While certain trade-offs were made to manage complexity and computational feasibility, the resulting scope was designed to maintain relevance within the limits of the available resources and expertise.

Modeling Simplifications

An initial boundary involved determining the level of detail in the Digital Twin of Chalmers, which was developed using Blender. The model primarily consisted of simplified geometric representations of buildings with larger geometrical details, while smaller objects such as benches, fences, and flagpoles were deliberately excluded. The incorporated topography of the Chalmers Campus area was manually estimated and constructed using multiple available open-source data sources. Hence, the resulting elevation model did not fully reflect the real terrain, and certain topographical details were excluded. The materials assigned to the objects in the model did not precisely replicate real-world conditions; instead, they were chosen from the materials available in the Sionna library. Specific materials not present in the library were approximated using similar alternatives available within it. In instances where walls featured multiple materials, the predominant material was typically selected, and their placements were estimated based on visual cues. Although a more detailed model would have been desirable for more accurate results, it was deemed impractical within the given time and resource limitations.

Coverage Limitations and User Equipment Grid Design

To simulate realistic radio propagation between BSs and UEs, multiple positions of UEs were required across the radio environment. Rather than covering the entire campus uniformly, the grid field was limited to preexisting roads and pedestrian pathways within the Chalmers campus. This approach concentrated the analysis on realistic and accessible UE locations, specifically in areas where users were most likely to be present. However, the OpenStreetMap (OSM) software used to generate the roads does not accurately depict the open fields surrounding the campus; instead, it generates narrow roads in place of the fields. Consequently, some regions were not covered by the grid, resulting in missing data in areas where users could possibly be located. While these regions could have been added separately, they were excluded due to time constraints. Moreover, the grid field used to generate data was limited to the campus entrance and consisted of 2816 grid points spaced by $1\text{m} \times 1\text{m}$. This was done to enable the simulation to be run on available devices within a manageable time frame. It is worth noting that a grid field of the whole campus was generated alongside, but not utilized due to limited computing power.

Propagation Modeling Constraints

The data generation script was limited to static UEs and did not incorporate user mobility. It also modeled LoS wave propagation along with specular and diffuse reflections, while excluding effects such as refraction and other propagation mechanisms. Due to time constraints during the project, these limitations were imposed to manage computational complexity.

System modeling and KPI Selection

In the system modeling process, several simplifications were made to ensure that the project remained within a manageable scope. Specifically, the channel model included in the received signal model was simplified by assuming static UEs, thereby excluding Doppler effects. Additionally, the angle of arrival (AoA) steering vector was omitted from the channel model under the assumption that the UEs consisted of single antenna elements. These assumptions reduced the model complexity while still enabling meaningful analysis. Furthermore, the received signal was assumed to follow an additive white Gaussian noise (AWGN) model. The AWGN model, due to its simplicity, is widely utilized in wireless communications for evaluating the performance of modulation schemes and communication or positioning KPIs. This made it particularly fitting for this project. Finally, the precoder included in the model was randomized, as designing a precoder would have gone beyond the scope of the project.

The evaluated KPIs were limited to capacity, latency, and PEB due to time limitations and the need for specific expertise. The stated metrics are also the most commonly used for evaluation in wireless communication, sensing, and positioning, making them particularly relevant for this project. More specifically, the latency usually includes several delays, such as transmission delay, propagation delay, processing delay, and queuing delay. However, the processing and queuing delays were

not included in the calculations of the overall latency as they depend on specific hardware characteristics that were unknown in the simulation context. Moreover, when calculating the Cramér-Rao lower bound (CRLB) to get a lower bound for the PEB usually involves implementing an algorithm to meet those bounds. However, such an algorithm was not developed as it exceeded the scope of the project. Additionally, in the calculations of the CRLB, only the LoS path parameters were further transformed into positioning parameters while the none-line-of-sight (NLoS) parameters remained unaltered. Accounting for the NLoS paths would have required analysis of the positioning of each scatterer and reflector, which would have proved time-consuming.

1.4 Outline

The remainder of this report is structured as follows:

- **Chapter 2** describes the process of building the Digital Radio Twin, including 3D modeling in Blender, scene export, and ray-tracing simulation using Sionna RT.
- **Chapter 3** defines the setup and signal model. It also presents the methodology for computing positioning and communication KPIs.
- **Chapter 4** presents and discusses the results, including visualizations of positioning and communication performance under different scenarios and parameter settings.
- **Chapter 5** concludes the report by summarizing the findings and suggesting directions for future work.

2

Building the Digital Radio Twin

This chapter outlines the process of creating a digital radio twin that serves as the foundation for realistic wireless communication simulations. The digital twin replicates a real-world urban environment in 3D, enabling accurate modeling of wave propagation and radio channel behavior. The process involves multiple stages, including the import and adjustment of building structures from OSM data, refining architectural features, incorporating elevation through manual topography modeling, the assignment of physical material properties for electromagnetic compatibility, and generating a spatial UE grid. The final 3D environment was exported from Blender to a format compatible with the Mitsuba renderer and integrated into the Sionna RT simulation framework. After being imported into Sionna RT, the scene was configured for radio simulation, which involved defining the carrier frequencies, antenna arrays, and precise placement of BSs and UEs. With the simulation environment prepared, ray tracing was conducted using the Sionna RT `PathSolver` to compute channel parameters such as delay, Doppler shift, and reflection paths. These outputs were organized and exported as structured data for subsequent analysis and visualization. Both the modeling methodology and simulation workflow will be presented in detail in the following sections and serve as the pipeline for building a digital radio twin. A simplified block diagram of the pipeline can be seen in Figure 2.1.

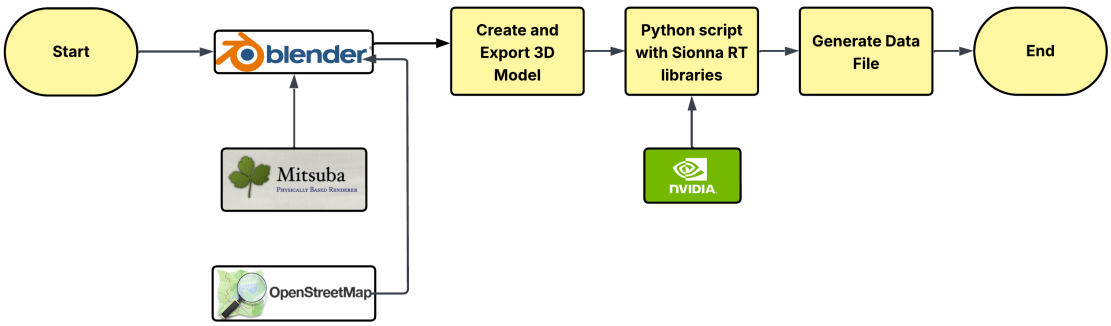


Figure 2.1: Overview of the tools used for building the Digital Radio Twin.

2.1 Introduction to Blender

Blender is an open-source tool that supports the full pipeline of 3D content creation. This project used Blender to design complex, realistic 3D outdoor environments that serve as the geometric input to the ray tracing simulation engine (Sionna RT). It enables precise placement of materials and objects in the scene and supports scripting via Python, making it highly compatible with automated workflows. The following sections highlight the core features relevant to this project. More information on Blender's capabilities can be found in the Blender documentation [11].

3D Modeling and Material Definition

Blender provides advanced tools for mesh editing, sculpting, and solid modeling. These tools allow for precise control over geometry, enabling the design of rooms, walls, objects, and surface details critical for accurate ray tracing simulations. Boolean operations, modifiers, and snapping tools enhance the ability to generate realistic and physically accurate scenes. Materials can be assigned to structures and surfaces, and can be named and colored to distinguish them from one another.

Scripting and Extensibility

Through the built-in Python Programming Interface, Blender can be fully scripted, allowing users to automate scene generation, parameter sweeps, or data export processes. The scripting interface integrates well with the simulations needed for this project and played a crucial role in defining the positions of the UEs. Since Blender's scripting language is Python, transferring and integrating data created within Blender into other Python-based environments, such as Sionna RT, becomes straightforward.

Other Features

Beyond modeling and scripting, Blender also supports features such as rendering, animation, compositing, and simulation. Although these were not core to the project, they enable detailed and realistic visualization of the simulation scenes. In this project, Blender's abilities were limited to 3D modeling, material definition, and scripting tools, which form the backbone of the digital environment and the UE-grid fed into the Sionna simulation pipeline.

2.1.1 Blender add-ons

Blender offers a variety of add-ons that can be used to enhance and extend its features and capabilities. The ones used in this project were OSM add-on for Blender (BLOSM) and the Mitsuba Blender add-on. OSM is an open-source geographical database that provides data on natural features, infrastructure, and buildings [12]. The BLOSM add-on for Blender is a tool that enables users to import OSM data into Blender. The imported data can then be utilized to create the foundational elements of a 3D model that represents a real-world area. The Mitsuba Blender add-on is a tool that converts and exports the digital 3D model created in Blender, including its geometry and material properties, into an XML file format [13]. This format is compatible with the Mitsuba 3 rendering engine, which serves as the ray-tracing backend for Sionna RT in simulating electromagnetic wave propagation [14]. As a result, the exported model can be used directly in Sionna RT.

2.2 Introduction to Sionna

Sionna is an open-source library developed by NVIDIA for research on wireless communication systems. It is designed to leverage GPU acceleration and comprises three main modules:

- Sionna RT: A ray-tracing engine for modeling radio wave propagation.
- Sionna PHY: Used for link-level simulations between a single BS and UE.
- Sionna SYS: Used for system-level simulations across entire networks.

The descriptions of Sionna and its functionalities in this section are based on the official documentation provided by NVIDIA [14].

The module primarily used in the data generation pipeline is Sionna RT. It is used to facilitate the simulation of radio wave propagation, which is achieved through the integration of the Mitsuba 3 rendering framework and the Dr.Jit differentiable just-in-time compiler. Utilizing these tools, Sionna RT performs ray tracing to simulate and compute channel-related outputs such as channel frequency response (CFR) and average received power. The ray tracing engine enables simulations of radio wave propagation in various 3D settings by incorporating electromagnetic material characteristics, the structure of the 3D scene, and antenna configurations.

In summary, Sionna RT is a differentiable and GPU-accelerated radio channel simulator, suitable for modeling and analyzing modern wireless communication systems. The rationale behind employing Sionna RT as a tool becomes clearer when considering the broader advantages of the computational method it is based on, that is, ray tracing, and how it is integrated within Sionna RT.

2.2.1 Advantages with Ray-Tracing and Sionna RT

Ray tracing is a computational method utilized for simulating the propagation of electromagnetic waves, particularly effective in complex environments where phenomena such as direct paths, reflection, diffraction, and scattering are considered. This technique systematically traces the paths of rays as they interact with physical objects within the environment, thereby accurately modeling wave behavior. Ray tracing holds significant relevance in communication system research, notably in the study of next-generation networks such as 6G, where precise simulation of multipath effects, spatial diversity, and localization is essential. Unlike approaches relying on simplified wave propagation approximations, ray tracing incorporates detailed scene geometry to deliver precise simulation outcomes.

Among the available ray tracing tools, Sionna RT was selected because it offers the advantages mentioned above while being an open-source library. It facilitates the simulation of radio wave propagation by leveraging 3D scene geometry and electromagnetic material properties, thereby eliminating the need to manually implement wave interaction models. This enhances research efficiency and enables the simulation of complex environments and wave behaviors that would be impractical to model manually. Furthermore, Sionna RT offers a range of functionalities relevant to wireless communications research, which will be further elaborated in the following sections.

2.2.2 Core Features of Sionna RT

As previously mentioned, the primary capability of Sionna RT is its ability to simulate radio wave propagation through ray tracing in 3D environments. It accounts for various types of interactions, including direct paths, specular (mirror-like) reflections, and diffuse scattering from rough surfaces. It also accounts for the electromagnetic properties of materials, the geometry of the environment, and antenna configurations, among other factors.

This functionality enables the computation of several channel characteristics, such as time delays, Doppler shifts, and spatial angles. Subsequently, these parameters are used by Sionna RT to derive radio channel outputs, including channel impulse responses (CIRs) and CFRs.

Sionna RT is also capable of simulating not only static scenes, but also dynamic scenarios such as environments involving moving UEs. Other features of Sionna RT include the ability to design various antenna array configurations, define BS and UE configurations, and generate data for use in machine learning applications.

To implement these capabilities, Sionna RT provides a structured set of classes and functions that facilitate these key simulation functionalities.

2.2.3 Overview of the Functional Components of Sionna RT

The Sionna RT module consists of a set of classes and functions that together enable the simulation of wave propagation, the creation of scenes, the generation of radio maps, and the calculation of channel responses, among other tasks. In Figure 2.2 are the functionalities found in Sionna RT and an overview of the classes and functions utilized in the data generation pipeline.

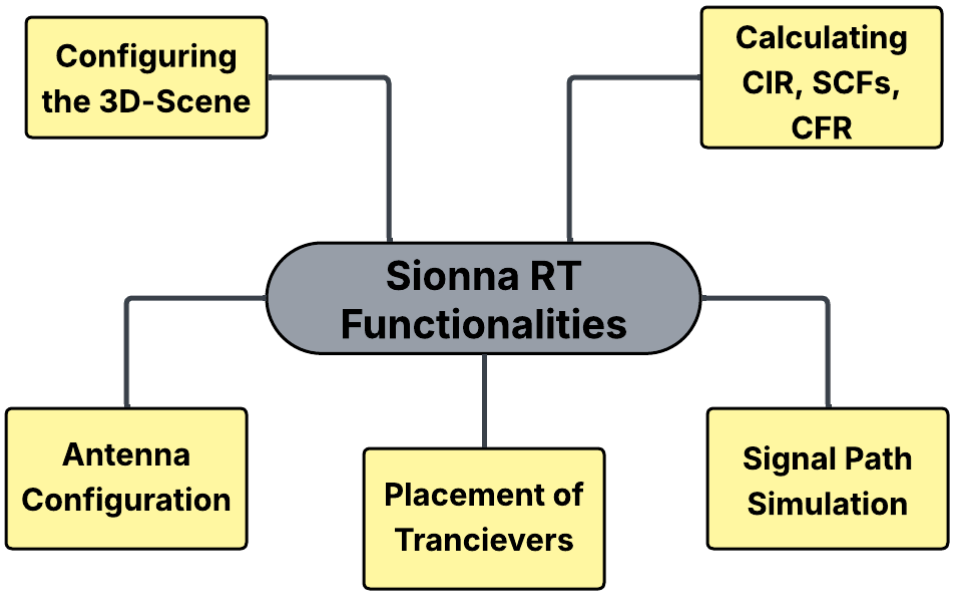


Figure 2.2: Block diagram of the functionalities of the Sionna RT module.

Configuring the 3D-Scene

The `load_scene()` function is used to import 3D models described in the Mitsuba 3 XML file format. When called with the imported filename, `load_scene()` returns a scene object that includes the scene geometry, material definitions, and other elements required for wireless propagation simulation. This object contains various scene components, such as buildings, surfaces with defined electromagnetic properties, and radio devices such as BSs and UEs.

If the filename argument is set to "None", the function instead returns an empty scene that can later be modified directly in Python within the data generation code. Scenes already created of different cities around the world, such as Munich, Germany, are also available for immediate use by calling on their specified argument name.

Once a scene has been created, it can be added to a simulation instance using the `scene.add()` method, which sets the scene as the simulation environment. The `load_scene()` function and the `scene.add()` method together allows for scene management and provides access to all geometrical, spatial, and material data necessary to perform the wave propagation simulation.

Antenna Configuration

The `PlanarArray` class is used to define antenna arrays for use in simulations. A planar array is specified by the number of rows and columns of antenna elements, as well as the spacing between the elements, which is defined in terms of wavelengths.

The array is placed in the y–z plane, with the antenna elements arranged on a regular grid. The antennas are indexed column-first, starting from the top-left corner. Each antenna can be assigned a configurable radiation pattern, such as isotropic or dipole, and a polarization setting. These arrays can then be used to model multiantenna configurations and can be attached to either BSs or UEs within the simulation environment.

These configurations can be assigned to `scene.tx_array` object for the BSs and to `scene.rx_array` object for the UEs. Parameters such as the number of rows and columns, element spacing, polarization, and radiation pattern can be defined during the instantiation of the `PlanarArray` class with the above-mentioned objects.

Base Stations and User Equipments

After defining the antenna arrays, individual BS and UE objects can be created using the `transmitter` and `receiver` classes as mentioned above. These devices serve as signal sources (BSs) and targets (UEs) within a simulation scene. They act as the nodes between which radio wave propagation is simulated.

Each UE and BS is initialized with a unique name, position, and orientation. Setting the orientation to (0, 0, 0) will use the default orientation in Sionna RT, which is directed along the positive X-axis and Z-axis. These devices can then be added to the scene using the `scene.add()` method, which registers them for use in subsequent ray-tracing simulations. Once added, their spatial configurations, that is, their location and orientation, can be accessed and modified as needed.

The `receiver` class also supports the addition of velocity parameters, enabling the modeling of moving UEs, as well as visualization settings. The `transmitter` class includes a `power_dbm`, which specifies the transmission power to be used for each BS in the simulation.

Together, the `transmitter` and `receiver` classes encapsulate all necessary parameters, such as position, orientation, and transmission and reception properties, required for modeling the sources and targets used in the simulation.

Signal Path Simulation

The `PathSolver` class in the Sionna RT library is used to compute signal propagation paths between BSs and UEs within a loaded 3D scene. It implements a ray tracing algorithm that simulates how electromagnetic waves travel through the environment by modeling both direct and indirect propagation effects, such as refraction, scattering, and reflections. These include LoS paths, specular reflections, diffuse scattering, and refraction. The type of interaction considered can be configured in the simulation by specifying appropriate input parameters when calling the `path_solver` object. Multiple instances of `path_solver` objects can be initialized with different input parameters, allowing for flexible configuration of which wave

interactions to consider for different simulations.

For instance, the maximum number of reflections that a ray can experience can be controlled via a depth parameter, which limits the number of bounces a ray may undergo before reaching a UE. Refraction and reflection effects can also be configured individually through dedicated parameters when calling the `path_solver` object. For each valid propagation path, the solver computes a set of channel parameters. These include complex gain (consisting of both magnitude and phase), propagation delay, AoA, angle of departure (AoD), and the Doppler shift resulting from relative motion between the BS and UE. These will be explained in more detail in Subsection 2.2.4.

Additionally, `PathSolver` supports different methods for modeling antenna arrays. Arrays can be modeled explicitly by including individual antenna elements in the scene or synthetically by representing the array as a single source and applying spatial phase corrections afterward.

Extracting Ray-Tracing Simulation Results

Once the path computation is complete, the `PathSolver` class returns a `Paths` object that contains all simulated paths between each BS–UE pair, along with the corresponding computed information for each path.

For every path, several physical and geometric parameters are stored. The `Paths` object includes the coordinates of the reflection points (vertices) that define the geometric trajectory of each path. This object can then be used to extract relevant channel characteristics, which will be explained in more detail below.

`CIR` and `CFR` Computation from Ray-Tracing Data

The `Paths` object includes methods for using the parameters collected from the ray-tracing simulation to calculate outputs used for communication system analysis. Among these are:

- `cir()`: Computes the baseband-equivalent CIR, which describes how a transmitted signal propagates through the environment and arrives at a UE over time. It summarizes contributions from all propagation paths between each BS and UE antenna pair and incorporates relevant path parameters. The result is a multidimensional array that represents the behavior of the channel in the time domain.
- `cfr()`: Uses the output of `cir()` to compute the CFR by evaluating the CIR at defined subcarrier frequencies. This is useful in systems with OFDM, where frequency-dependent channel gains are required for modulation and demodulation. It outputs the frequency response across all antenna pairs.

The outputs from both methods can be saved and visualized in formats such as NumPy, making them more readable for further analysis. Together, `cir()` and `cfr()` use data from the ray-tracing simulation to characterize the wireless channels in both the time and frequency domains.

Generating Subcarrier Frequencies

The `subcarrier_frequencies()` function computes the baseband subcarrier frequencies used in an OFDM system. It takes the number of subcarriers and their spacing as inputs and returns an array of evenly spaced frequencies centered around 0 Hz, extending symmetrically from negative to positive values.

These frequencies are then used to evaluate the CFR at each subcarrier in OFDM-based simulations.

Visualizing the Simulation

The `Camera` class and the `scene.render()` function use the output of the solver to compute and visualize radio signal characteristics across a 3D scene. It can be used for, among other things, visualizing the received signal strength at many positions in the scene with heat maps and visualizing the propagation paths simulated in the environment, which can be useful for verifying that the simulated paths reflect the intended physical behavior and considerations.

2.2.4 Path Data Parameters from Sionna RT

The following path-level parameters can be computed for each ray, traced between a BS and a UE using Sionna RT. These quantities are crucial for physical-layer modeling and post-processing, including CIR construction and propagation analysis. A full description of each parameter is available in the official Sionna RT documentation [14].

- **UE Positions (x, y, z):** Each UE's spatial coordinates in 3D space, used as input to the simulation. They directly affect path calculations, such as delays and angles.
- **Channel frequency response (h_frequency_data):** The CFR for each link between BS and UE antennas across all subcarriers. It is a complex-valued tensor where each element corresponds to a specific subcarrier and antenna pair.
- **Average channel power (h_avg_power):** The average channel power over frequency and space. It provides a scalar value representing the channel gain for a BS-UE pair and is useful for link budget analysis.
- **Path gain coefficients (_a_complex):** Complex-valued path gain coefficients for each ray, with real and imaginary parts stored separately. These are used to reconstruct the channel in the time or frequency domain.
- **Doppler (_doppler):** Doppler shift of each path due to relative motion, measured in Hz. Essential for modeling time-varying channels.
- **Azimuth, angle of arrival (_phi_r):** The azimuth AoA at the UE, measured from the local X-axis. It describes the horizontal direction of the incoming ray.

- **Azimuth, angle of departure (_phi_t):** The azimuth AoD from the BS, also measured from the local X-axis. It describes the horizontal departure direction.
- **Propagation delay (_tau):** Path delay (time-of-flight) in seconds, calculated based on the ray's total traveled distance including bounces and refractions.
- **Zenith, angle of arrival (_theta_r):** The zenith AoA at the UE, measured from the horizontal plane. Together with `_phi_r`, it forms the AoA tuple that defines the direction of the transmitted path.
- **Zenith, angle of departure (_theta_t):** The zenith AoD from the BS. Together with `_phi_t`, it forms the AoD tuple that defines the direction of the transmitted path.
- **Interaction positions (_vertices):** 3D coordinates of all interaction points (reflections, diffractions, transmissions) along each path, used to trace physical propagation mechanisms.

2.3 Building the Digital Twin of Chalmers

This section describes the process of constructing the digital model of the Chalmers campus using Blender, generating a UE grid, and exporting the scene as well as the grid for use in Sionna. The model, along with the UE grid, was developed through the integration of geographic data, manual modeling, and scripting. Key steps include the import of OSM data, refinement and incorporation of exterior features to architectural structures, implementation of topography, assignment of materials compatible with Sionna RT, and the generation of a UE grid. This process can be seen in a simplified block diagram in Figure 2.3.

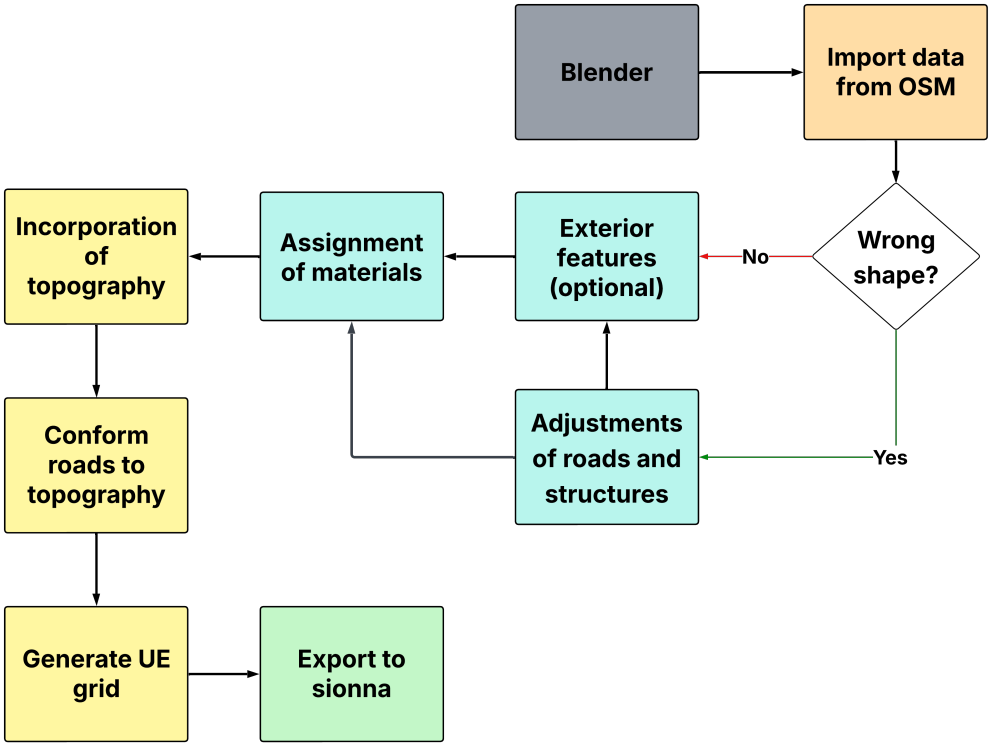


Figure 2.3: Block diagram on the construction of the Digital Twin of Chalmers using Blender.

2.3.1 Open Street Map

The OSM data for the Chalmers campus area was first imported into Blender using the OSM add-on. Only the road and building data were included in the import. The add-on generated the foundational model by creating block representations of buildings and flat planes for roads. In some cases, buildings and roads were imported with accurate shapes and dimensions based on available OSM data. For buildings lacking specific information, default preset sizes defined by the add-on were applied. However, the preset sizes had to be manually adjusted to accurately reflect real-world dimensions. All elements were correctly positioned relative to one another.

2.3.2 Structures

The buildings in the model are represented as planar surfaces with no thickness. Their dimensions were determined and imported based on available OSM data. In cases where such data were missing, buildings were visually compared to nearby structures with known dimensions (where data was available). Using 3D satellite imagery and in-person visual references, these buildings were manually adjusted to approximate their real-world scale. For buildings with incorrect shapes or architectural details, manual corrections were applied by either modifying the existing geometry or adding new objects to accurately reflect the actual structure.

2.3.3 Façade Elements and Roof Structures

After the buildings were appropriately scaled and shaped, exterior features such as windows, doors, pilasters, sills, ventilation systems, and other roof structures were added; an example of the façade elements can be seen in Figure 2.4. These elements were incorporated either by attaching new objects to the building's exterior or by directly modifying the geometry of the exterior walls and roofs. For instance, windows and pilasters were sometimes modeled as separate objects and placed onto the building surfaces, while in other cases, parts of the walls or roofs were subdivided and modified with different materials, extruded, or intruded to represent architectural features like doors, windows, and recessed windows. Exterior features were applied to only a subset of the buildings. Additional detailing may be incorporated in future stages of the work.

The choice of modeling method depended on the frequency and repetition of the architectural details. In cases where numerous identical elements were present, such as regularly spaced windows or pilasters, a new object resembling the detail was modeled and then duplicated across the surface to maintain consistency and efficiency. For less common details, direct modification of the surface itself was preferred.

For walls featuring recessed windows, one approach was to give the wall a thickness with openings at the window positions, allowing the windows to be placed within these holes. Another approach involved modifying an area of the wall by recessing it to simulate a window with depth. The method that was used depended on the frequency of the detail on the specific wall.

Where different sections of the walls required different materials, the walls were subdivided accordingly to enable the assignment of distinct materials to each segment. Doors were similarly modeled using this approach due to their infrequent occurrence on specific walls, resulting in a material resemblance rather than a detailed geometric representation.

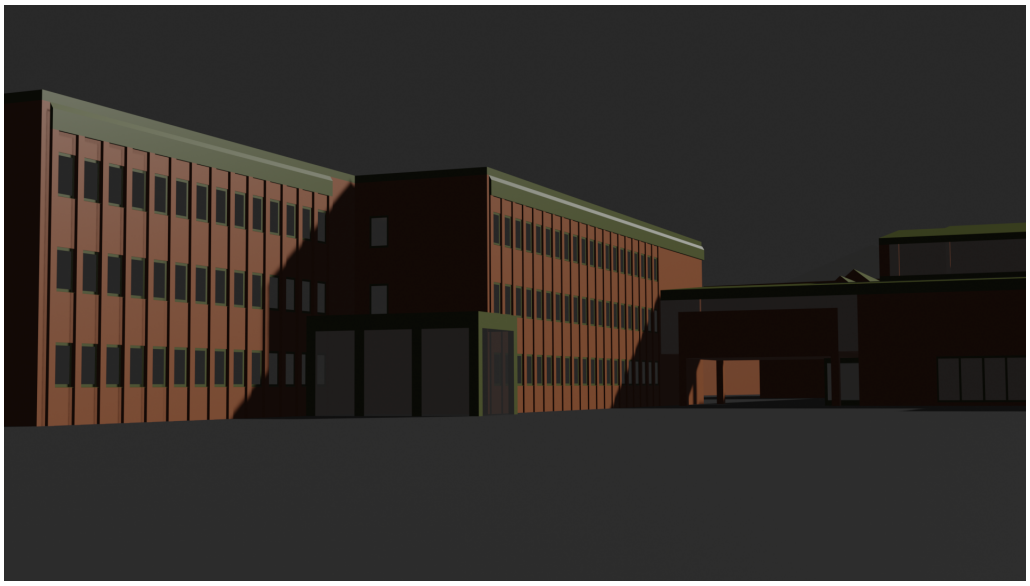


Figure 2.4: Image capturing the Façade Elements on the entrance of the Digital Twin of Chalmers.

2.3.4 Topography

Topographical data was not included in the OSM import. As a result, all elements were initially generated at the same elevation. To improve the model's accuracy and generate more accurate data, topography was manually incorporated. Using a combination of elevation-above-sea-level data and visual references from satellite imagery, the terrain was approximated by dividing the area into five distinct elevation levels, each represented by a separate plane. These levels were then interconnected using additional or already existing geometry, such as buildings that naturally contributed to the topographical transitions. The additional geometry represented features such as stairs, slopes, and natural terrain. These connecting elements were modeled using planes and cubes. In some areas, level planes were slightly tilted to mimic the gradual slope that characterizes the campus. Smaller hills surrounding the area were added to enhance the model's accuracy. Once the elevation model was complete, structures were adjusted to align with the topography, ensuring continuous contact with the connected elevation planes. The roads were conformed to the terrain using Blender's "Shrinkwrap" modifier, which projected the road geometry along the z-axis to follow the contours of the underlying elevation surface.

2.3.5 Materials

Materials were selected to approximate the physical properties of real-world surfaces. The materials used in the 3D environment included brick, concrete, glass, metal, wood, and marble. Each object within the scene was assigned one or more materials, either uniformly across the entire object, selectively to specific parts, or smaller areas through mesh subdivision. To ensure compatibility with Sionna RT, all materials were named according to a standardized convention recognizable by

Sionna RT. For example, itu_brick, itu_glass, and similar identifiers. All of the materials incorporated within Sionna RT and their identifiers can be found in the Sionna RT documentation [14]. This naming scheme allows Sionna RT to apply appropriate electromagnetic properties during the ray tracing simulation.

2.3.6 UE Grid

A spatial grid was generated using a custom Python script executed within Blender's scripting environment. The script placed small spherical objects at regular intervals as grid points, forming a 1m × 1m grid as shown in Figure 2.5. The grid was constructed directly on top of the selected road geometries, which were aligned with the terrain, including any elevation. As a result, the generated grid naturally followed the contour and elevation of the underlying ground.

The implementation includes adjustable parameters such as "road width" and "height above road", allowing the grid to be adapted to various road sizes and ensuring that the UE positions can be raised to a desired height above the surface. This enables the simulation of different UE scenarios.

The world-space coordinates of each grid point were extracted and exported to a CSV file. These coordinates were subsequently used in the Sionna wave propagation simulation by iteratively assigning each position to a UE.

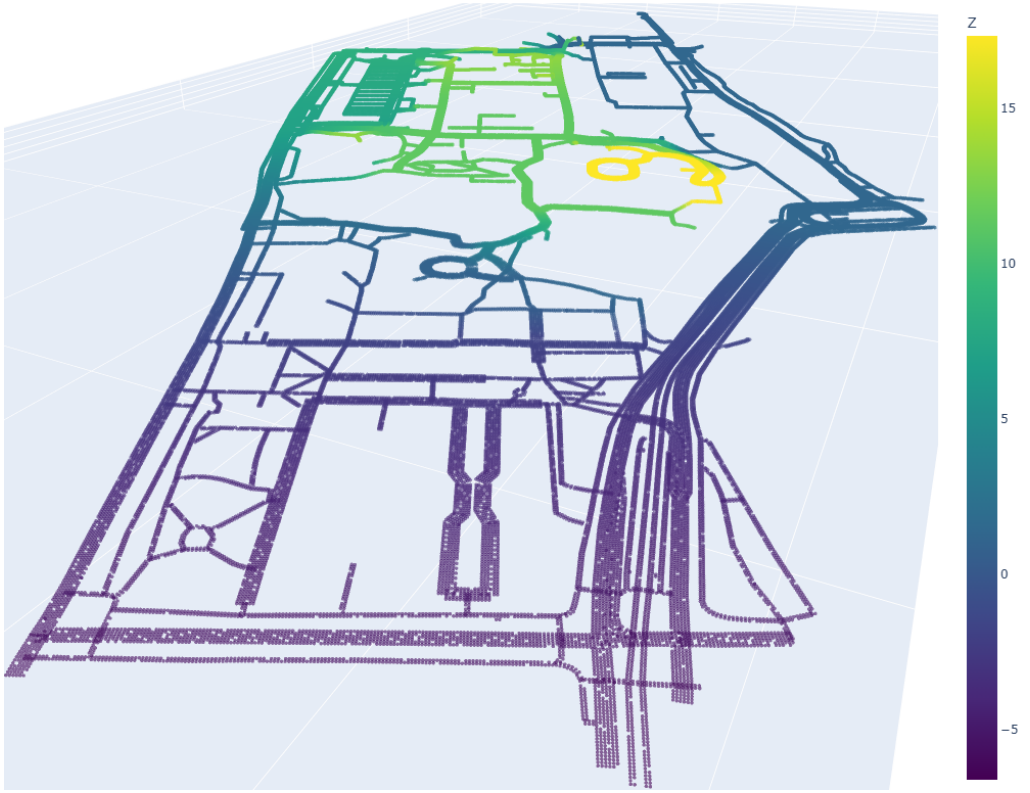


Figure 2.5: Plot of the 3-dimensional grid points for the Digital Radio Twin of Chalmers Campus (z-axis in meters).

2.3.7 Export scene to Sionna RT

The final model, which can be seen in the Figures 2.6, 2.7, 2.8, and 2.9, was exported as an XML file using the Mitsuba Blender add-on, which facilitates the integration of Blender scenes into the Mitsuba rendering engine. This export process converted the geometric data, materials, and scene configuration into a format compatible with Mitsuba, and by extension, with Sionna RT, enabling simulation based on the scene's physical properties.

2.3.8 The Digital Twin

Images of the Digital twin are shown in this section.

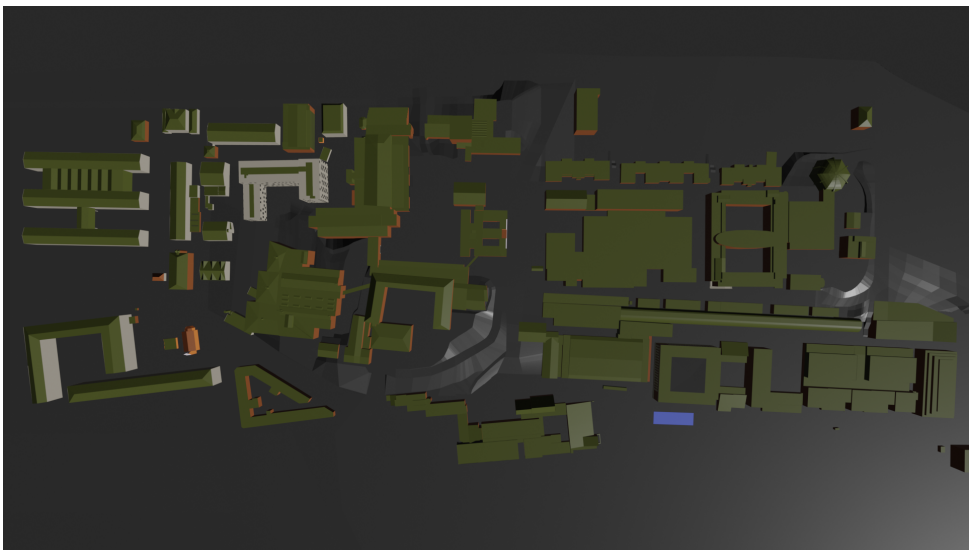


Figure 2.6: Image on the Digital Twin taken from above.

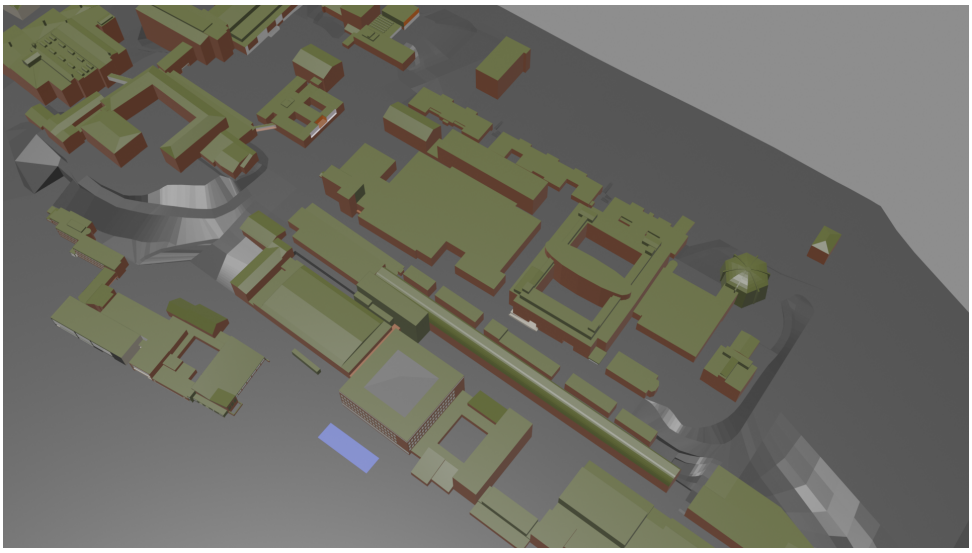


Figure 2.7: Image on the east side of the Digital Twin.

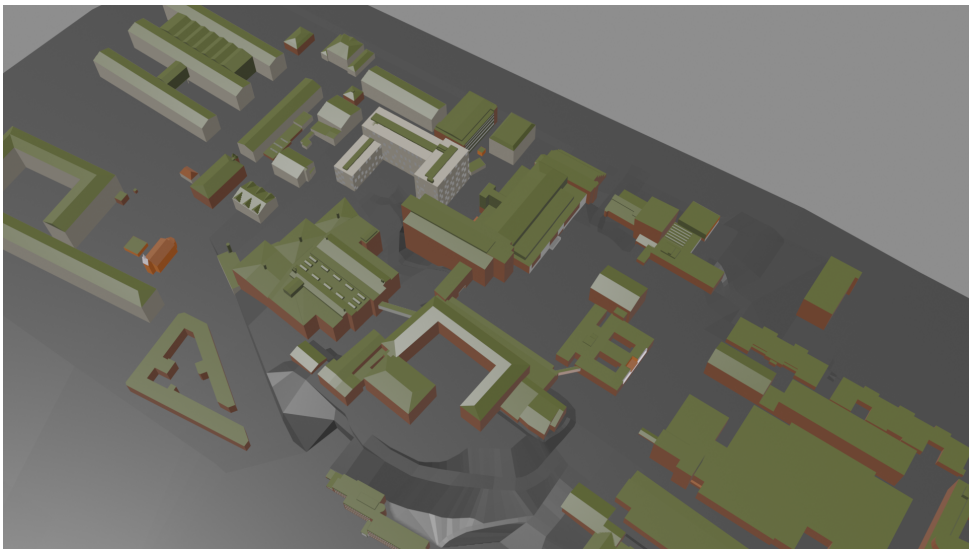


Figure 2.8: Image on the west side of the Digital Twin.

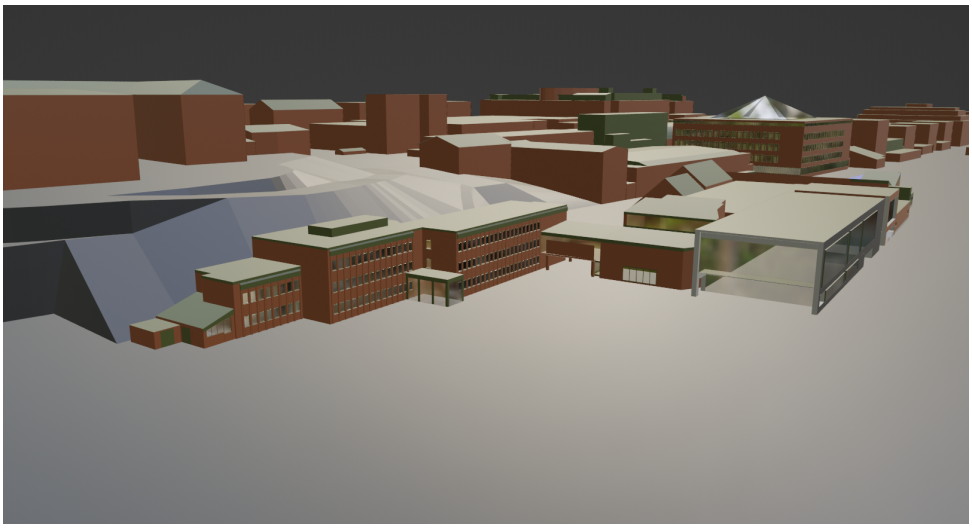


Figure 2.9: Close-up image of the Chalmers entrance in the Digital Twin.

2.4 Data Generation using Sionna RT

With the 3D scene exported, the data generation was feasible. This section describes the process of generating path data using the Sionna RT library. The process was divided into two main phases:

1. Preparation of the simulation environment.
2. Running the simulations and collecting the path data.

Each phase is described below to provide a description of the simulation setup, and the data generation and collection process. A visualization of the process is shown in Figure 2.10.

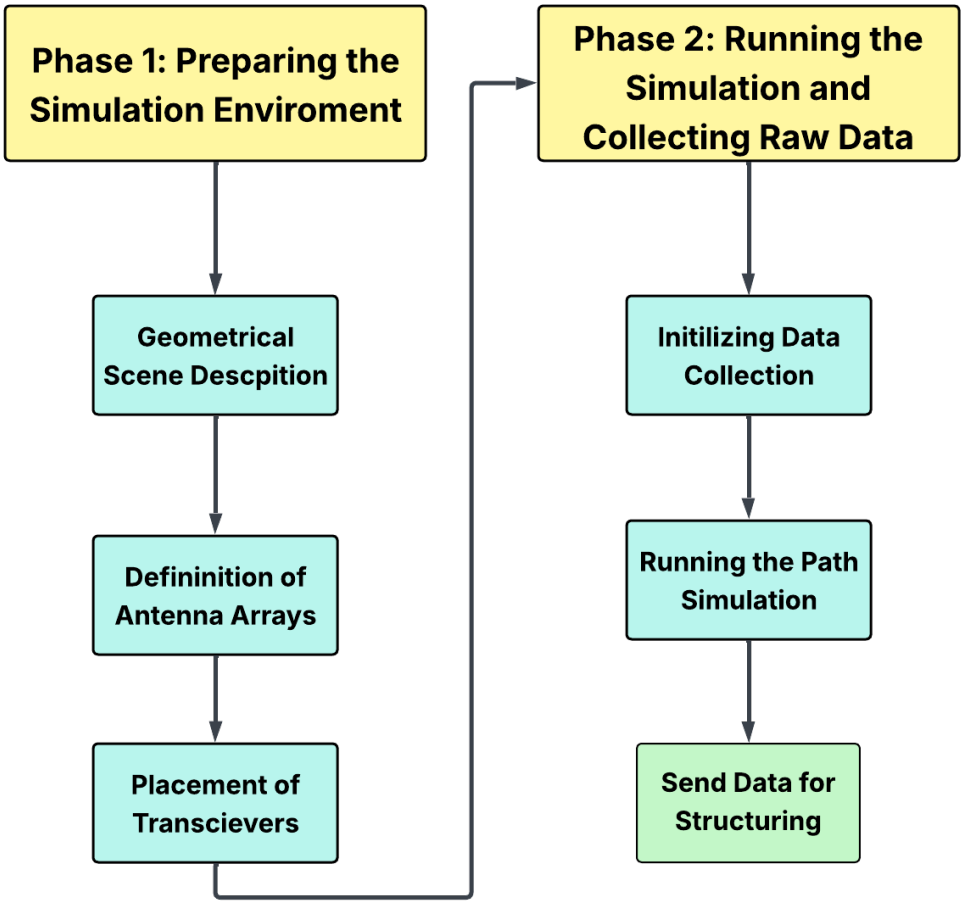


Figure 2.10: Block diagram on the data generation using Sionna RT.

2.4.1 Simulation Environment Configuration

The initial phase focused on setting up the simulation environment. This involved importing the geometric model of the physical area, specifying frequencies, configuring antenna parameters, and placing BSs and UEs within the scene.

Importing Geometrical Scene Descriptions

The preparation of the simulation began with importing a 3D scene into Sionna RT using the `load_scene()` function, which loaded an XML file containing all geometric and material definitions of the Digital Twin designed in Blender. This function then returned a scene object that contained all the geometrical and material information from the loaded XML file.

A carrier frequency, f_c, of 3.5 GHz was then assigned to the scene, corresponding to a typical sub-6 GHz band used in 5G New Radio (NR) systems according to 3GPP standards. Setting this frequency ensured that every frequency-dependent material property in the scene was evaluated at the 3.5 GHz band.

The free-space wavelength corresponding to the previously mentioned carrier frequency was then set with ($\lambda = \frac{c}{f_c}$) and half of this wavelength, ($\lambda/2$), was set as the spacing between elements in the antenna arrays, (`antenna_dist`).

Definition of Base Staion and User Equipment Antenna Arrays

Next, BS and UE antenna arrays were defined using the `PlanarArray` class. For the BSs, a 2×2 planar array with vertical and horizontal spacings equal to the previously defined `antenna_dist`, an isotropic element pattern, and vertical polarization were selected. The array was then assigned to the `scene.tx_array` object.

In the same manner, but with a single antenna element, the array for the UEs was configured and instantiated with `scene.rx_array`. By attaching these arrays to these objects, every BS and UE added thereafter automatically adopted the specified configurations.

Placement and Configuration of Base Stations and User Equipments

With the antenna arrays defined, BSs were added to the scene. Each BS was instantiated using the `transmitter` class, assigned a unique name, and positioned at fixed coordinates within the 3D scene.

Each BS was also configured with orientation angles $(\theta, \phi, \psi) = (0, 0, 0)$, corresponding to pitch, roll, and yaw, respectively. This configuration aligned the local coordinate system (LCS) of the BS with the global coordinate system (GCS), such that the antennas faced along the positive X-axis with the Z-axis oriented upward. BSs were then added to the scene with `scene.add()` method.

For UEs, a set of positions were extracted from a CSV file containing a grid of coordinates. This file included (X, Y, and Z) coordinates corresponding to a uniformly spaced 1 m $\times$ 1 m grid aligned on the pathways and roads of the Digital Twin.

The coordinates were then saved into a list, where each entry represented a single UE position. These positions define the set of UE locations that were used in the data generation phase and would be iterated over in the path simulation loop and loaded into the scene with the `scene.add()` method.

With this done, phase one, that is, preparing the simulation environment, was completed.

2.4.2 Data Generation and Collection

In the second phase, the simulation environment prepared in the previous steps was used to run a ray-tracing path computation. For each UE position, propagation paths were simulated, and the resulting channel parameters were collected and stored.

Initializing Data Collection

Before starting the path simulation loop, four different data structures were prepared:

- `data_combined`: A list that holds one dictionary per UE position, with each dictionary containing all of the UE's channel data.
- `desired_parameters`: A list of the parameter names to extract from each `paths_object`, including complex path gains, propagation delays, Doppler shifts, AoAs, AoDs, and reflection vertices.
- `h_frequency_data`: A dictionary mapping each UE coordinate position to its full CFR array.
- `h_avg_power_data`: A dictionary mapping each UE coordinate to its average received signal power, computed from the CFR as the mean squared magnitude.

These containers allowed for the accumulation of all simulation outputs for each UE in memory, before the outputs were written to a final CSV file. They were then overwritten for the next UE after the data had been stored in the final CSV file. This enabled efficient data handling and use of memory.

Configuration of the Path Simulation

The class used for simulating wave propagation in the scene was the `PathSolver` class. A single `PathSolver` instance was created before the main loop and configured to consider and ignore certain types of propagation data. It considered LoS paths, specular reflections, and diffuse reflections, and ignored refractions. The maximum path depth was set to `max_depth=2`, meaning that up to two reflections or bounces of rays were allowed per path. The same `PathSolver` object was reused and iterated over for every UE position.

Simulating for Each User Equipment Position

For each UE position, the following steps were performed in sequence and are visualized in Figure 2.11:

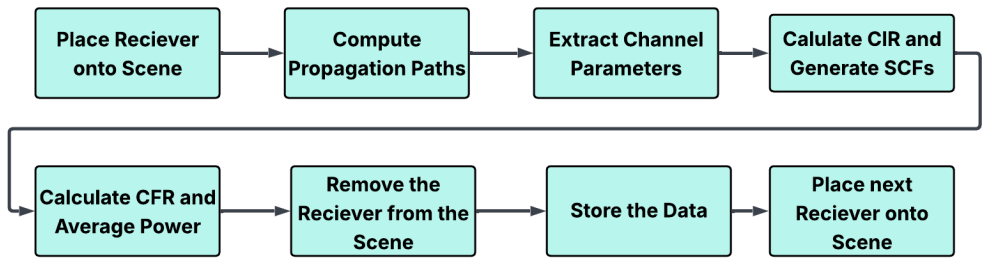


Figure 2.11: Block diagram of the path simulation loop.

1. **Place the UE into the scene:** A new `Receiver` object was created and assigned the current position from the UE grid. This UE was then added to the scene object to initiate the simulation.
2. **Compute propagation paths:** Called on the `PathSolver` for the current scene to compute all possible wave propagation paths between the BSs and the current UE. These included LoS- and reflected paths up to the configured maximum path depth.
3. **Extract channel parameters:** From the resulting `Paths` object, the following parameters were collected: the complex path gains, delays, Doppler shifts, AoA, AoD, and the reflection vertices (positions of the bounces).
4. **Compute channel impulse and frequency responses, and the average power:** The CIR was computed first using the `paths.cir()` method, which provides the complex path gains and corresponding delays. A frequency-domain representation was then obtained by evaluating the CIR over a set of subcarrier frequencies defined by the `subcarrier_frequencies()` function. Finally, the average received power was calculated as the mean squared magnitude of the CFRs across all subcarriers.
5. **Remove the UE from the scene:** After the required data of the UE was collected, the corresponding `Receiver` object was removed from the scene to make space for the next UE. This step ensured that only one active UE was present in the simulation environment at a time.
6. **Storing the data:** All collected data for the current UE was compiled into a dictionary, using the UE's coordinates (position) as key. This dictionary was then appended to `data_combined`. The CFR and average power values were saved in their respective dictionaries, still using the UE's coordinates as key.
7. **Repeating the process:** These steps were repeated for each UE position on the grid.

Raw Data Collection

After all grid points (UEs) had been processed, the outcome was represented as a list of dictionaries, `data_combined`, with each dictionary corresponding to a distinct UE position and its associated simulation data. This list served as the starting point for the next stage of the pipeline, where the raw simulation data was structured into an organized format suitable for further analysis. This process will be elaborated on in the upcoming section.

2.4.3 Visualizing the scene

To validate the simulation setup and gain an intuitive understanding of the computed propagation paths, a series of visualizations were generated using the `Camera` class and the `scene.render()` function. The visualizations illustrate how radio signals travel from the BSs to the UEs within the Digital Radio Twin of Chalmers. They served to validate that the scene configuration and propagation parameters had been correctly applied and assessed whether the simulated paths behaved as expected based on the scene environment and the selected path configurations.

In the visualizations of the propagation paths, as seen in the Figures 2.12 and 2.13, BSs are depicted as red markers positioned on top of buildings, while the UEs are shown as green markers located on the pathways. The radio propagation paths are illustrated using two distinct colors: gray lines represent LoS paths, and purple lines represent a reflected path.

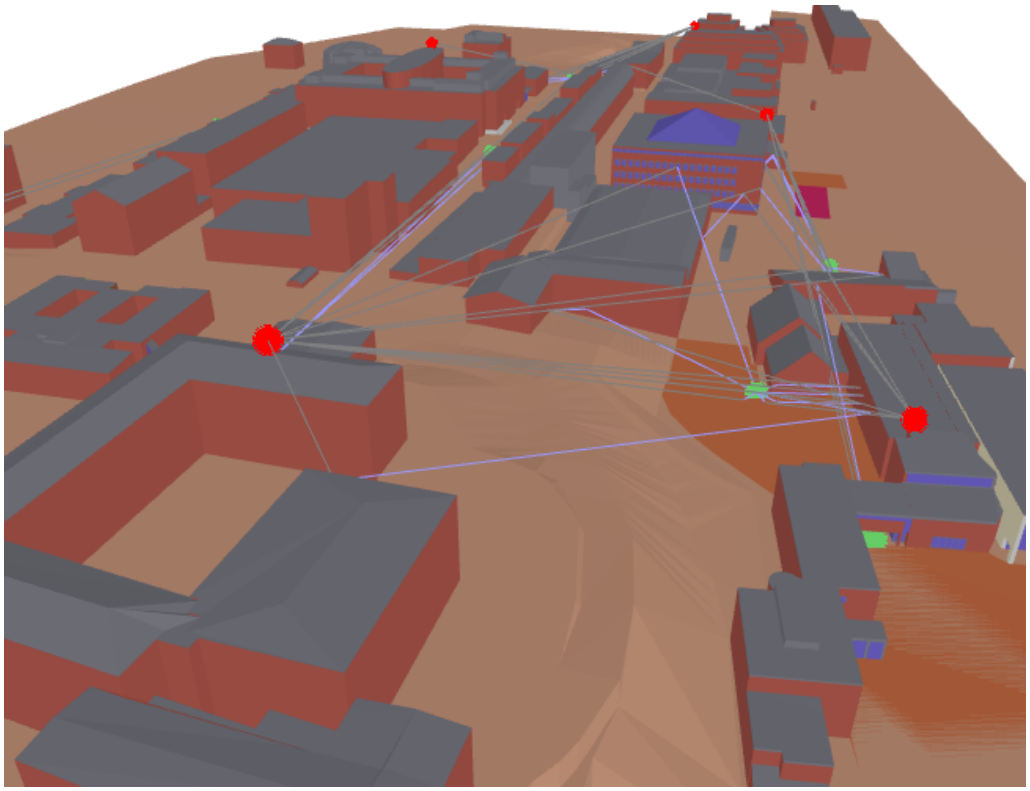


Figure 2.12: Radio propagation paths: red = BSs, green = UEs, gray = LoS, purple = reflections.

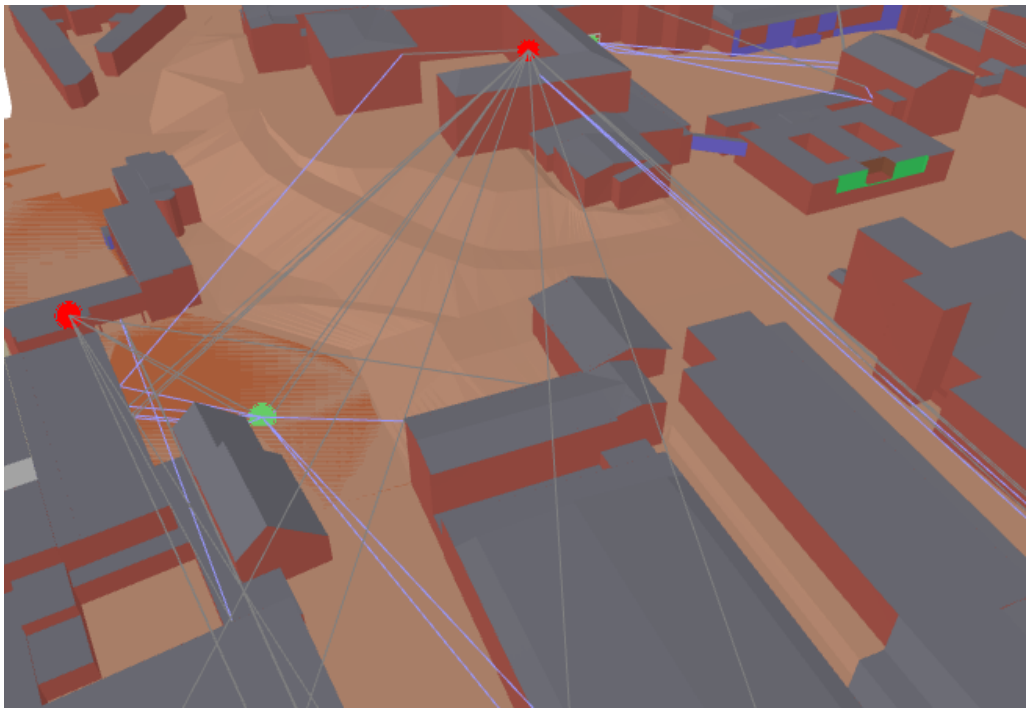


Figure 2.13: Radio propagation paths: red = BSs, green = UEs, gray = LoS, purple = reflections.

2.5 Data Handling and Storage

This section outlines the systematic process for organizing, validating, and exporting the ray tracing simulation results. The goal was to generate structured and MATLAB-compatible CSV files for every BS-UE pair while maintaining a consistent and scalable folder hierarchy. The overall structure is illustrated in Figure 2.14.

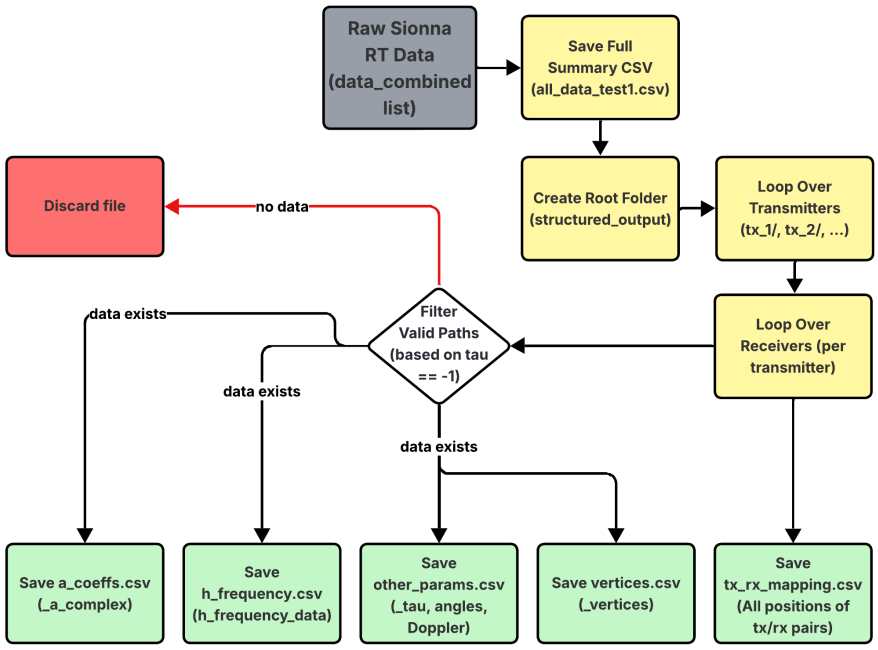


Figure 2.14: Overview of the data handling and storage pipeline.

Combined Summary CSV

Initially, as mentioned above, all simulation data was aggregated into a Python list of dictionaries, where each dictionary contains the relevant parameters for each UE. This collection was then converted into a structured table format using the Pandas library. The resulting dataset was then exported as a CSV file.

Creating the Output Folder Structure

The simulation results were exported into a structured directory hierarchy. A root output folder was created, named `structured_output`. Within this, a folder was created for each BS, labeled as `tx_1`, `tx_2`, etc. Inside each BS folder, separate subfolders were created for each UE, such as `rx_1`, `rx_2`, and so on. This design supports both single- and multi-BS scenarios, and any number of UEs. It ensures that the data associated with each BS-UE pair is isolated, making it easily accessible. The structure was automatically created regardless of the number of active BS or UE units.

Identifying Valid Paths

For each BS-UE pair, valid propagation paths were identified based on the delay (`_tau`) values extracted from the ray tracing results. A path was invalid if its delay was equal to -1. This was then used as an indicator for non-existent or invalid paths. Only valid paths were retained for storage, ensuring data quality and relevance.

Exporting Multi-path Fading Coefficients (`a_coeffs.csv`)

The complex channel coefficients for each valid propagation path were saved in the `a_coeffs.csv` file, located within each BS-UE pair folder. Each row corresponds to one propagation path, and the columns alternate between the real and imaginary parts of the fading coefficients. The number of columns depends on the total number of BS and UE antennas, calculated as:

$$2 \times (\texttt{num\_tx\_antennas} \times \texttt{num\_rx\_antennas}) \text{ per path.}$$

The current implementation handles any antenna configuration defined by the user. A file is only generated if at least one valid path exists for the corresponding BS-UE pair.

Exporting Channel Frequency Responses (`h_frequency.csv`)

The CFR (`h_frequency_data`) computed for each subcarrier was stored in the `h_frequency.csv` file. Each row represents one subcarrier, and the columns provide the real and imaginary parts of the CFR for each BS-UE antenna pair. The total number of columns is:

$$2 \times (\texttt{num\_tx\_antennas} \times \texttt{num\_rx\_antennas}).$$

This format allows a detailed view of how the channel varies between frequencies. The structure supports single and multiple BSs and UEs. If the number of subcarriers changes, the file dimensions will adjust accordingly.

Exporting Path Parameters (`other_params.csv`)

Additional path-specific parameters, such as the time delay (`_tau`), Doppler shift (`_doppler`), azimuth AoA/AoD (`_phi_r`/`_phi_t`), and zenith AoA/AoD (`_theta_r`/`_theta_t`) were saved into the `other_params.csv` file. Each row corresponds to a single path, and each column represents one of the previously mentioned parameters in their respective order.

Exporting Reflection and Scattering Points (`vertices.csv`)

The physical locations of the interaction points (including reflections and scatterings) along each valid propagation path were exported to the `vertices.csv` file. Each row corresponds to one interaction point per path and includes the 3D coordinates (x, y, z), prefixed with a label indicating the reflection order (e.g., first bounce, second bounce, etc.). For example, in a two-bounce scenario, each path will contribute two rows, one for each interaction point.

The matrix was padded to match the path with the highest number of reflections observed in the simulation. As such, those with fewer bounces would be represented with the same number of rows, maintaining consistent dimensions across all entries. For instance, in a case with three propagation paths and a maximum of two reflections, the resulting file would contain six rows (two per path) and four columns. The first column identifies the reflection order, while the remaining three columns contain the corresponding 3D coordinates.

Exporting Base Staion/User Equipment Mapping (`tx_rx_mapping.csv`)

A file named `tx_rx_mapping.csv` was created to provide an overview of the 3D coordinates of all BSs and UEs used in the simulation. Each row includes a unique identifier followed by the (x, y, z) coordinates. BSs are listed first, followed by UEs. This mapping enables traceability between the folder structure and the physical layout of the digital environment. It supports any number of BSs and UEs. However, in the current implementation, the identifiers are numeric (e.g., `1`, `2`, ...) rather than string names (e.g., `TX1`, `TX2`, ...). Therefore, the dimensions vary, depending on the number of BSs and UEs.

3

Channel Modeling and KPI Computation

The generated data from Sionna enables the evaluation of channel performance properties, known as KPIs. These metrics can be analyzed to understand how different configuration settings affect channel performance. This chapter will initially cover system modeling, where the setup and utilized signal models are defined, and subsequently detail the methodology for the computation of positioning and communication KPIs.

3.1 System Modeling

Before computing different positioning and communication KPIs, it is essential to initially define the system environment, including the positioning, orientation, and antenna configurations of both the BSs and UEs. Additionally, it is important to account for the transmission system, including its structure and modulation scheme, as well as the characteristics of the received signal and the channel model. An overview of the process can be observed in Figure 3.1.

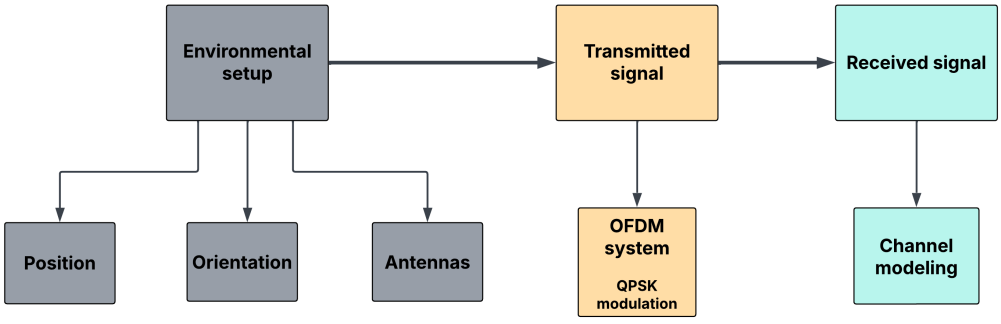


Figure 3.1: Block diagram of the system modeling process.

3.1.1 Setup

Consider a single stationary BS at a known position $\boldsymbol{p}_{\mathrm{BS}} = [x_{\mathrm{BS}}, y_{\mathrm{BS}}, z_{\mathrm{BS}}]^\top$ and with known orientation $\boldsymbol{\phi}_{\mathrm{BS}} = [\phi_{x_{\mathrm{BS}}}, \phi_{y_{\mathrm{BS}}}, \phi_{z_{\mathrm{BS}}}]$, for the pitch ϕ_x, roll ϕ_y, and yaw ϕ_z angles with respect to the GCS. Similarly, the UE on the generated grid field is located at $\boldsymbol{p}_{\mathrm{UE}} = [x_{\mathrm{UE}}, y_{\mathrm{UE}}, z_{\mathrm{UE}}]^\top$, and is assumed to be static. The distance between the BS and the UE is denoted as $d = \|\boldsymbol{p}_{\mathrm{BS}} - \boldsymbol{p}_{\mathrm{UE}}\|$, see Figure 3.2 for clarifications regarding the intended setup. Note that in this setup, the x-axis is used as the reference for the horizontal plane and the z-axis for the elevation (vertical) plane.

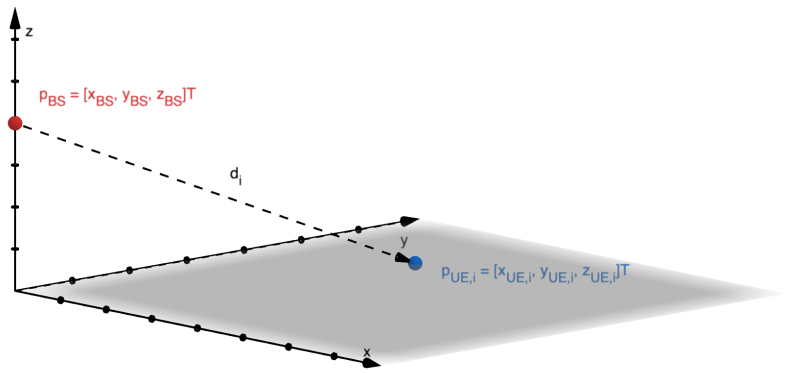


Figure 3.2: 3D coordinate system illustrating the setup.

The BS comprises $\mathrm{N}_{\mathrm{BS}} = \mathrm{N}_y \times \mathrm{N}_z$ omnidirectional antenna elements, with N_y and N_z representing the number of antennas along the BS local y and z-axis. The antenna elements are equally spaced by $\frac{\lambda}{2}$, where $\lambda = \frac{c}{f_c}$ is the wavelength, which is the speed of light c over the carrier frequency f_c. The n^{th} antenna is located at $\boldsymbol{p}_n = [0, y'_n, z'_n]^\top$ and is expressed in the LCS of the BS. Conversely, the UE is configured with a single omnidirectional antenna element.

The rotation matrix, used to transform the positioning parameters of the BS from the LCS to the GCS, is denoted as $\boldsymbol{R}_{\mathrm{BS}} = \boldsymbol{R}_z(\phi_{z_{\mathrm{BS}}})\boldsymbol{R}_x(\phi_{x_{\mathrm{BS}}})\boldsymbol{R}_y(\phi_{y_{\mathrm{BS}}})$, with $\boldsymbol{R}_z(\phi_{z_{\mathrm{BS}}})$, $\boldsymbol{R}_x(\phi_{x_{\mathrm{BS}}})$, and $\boldsymbol{R}_y(\phi_{y_{\mathrm{BS}}})$ defined as

$$\boldsymbol{R}_z(\phi_{z_{\mathrm{BS}}}) = \begin{bmatrix} \cos(\phi_{z_{\mathrm{BS}}}) & -\sin(\phi_{z_{\mathrm{BS}}}) & 0 \\ \sin(\phi_{z_{\mathrm{BS}}}) & \cos(\phi_{z_{\mathrm{BS}}}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{R}_x(\phi_{x_{\mathrm{BS}}}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_{x_{\mathrm{BS}}}) & -\sin(\phi_{x_{\mathrm{BS}}}) \\ 0 & \sin(\phi_{x_{\mathrm{BS}}}) & \cos(\phi_{x_{\mathrm{BS}}}) \end{bmatrix},$$
$$\boldsymbol{R}_y(\phi_{y_{\mathrm{BS}}}) = \begin{bmatrix} \cos(\phi_{y_{\mathrm{BS}}}) & 0 & \sin(\phi_{y_{\mathrm{BS}}}) \\ 0 & 1 & 0 \\ -\sin(\phi_{y_{\mathrm{BS}}}) & 0 & \cos(\phi_{y_{\mathrm{BS}}}) \end{bmatrix}. \tag{3.1}$$

With the use of the rotation matrix, it is possible to calculate the local LoS angles θ' and φ' of the BS (i.e AoDs) using the following formula

$$\begin{bmatrix} \theta'_{\mathrm{BS}} \\ \varphi'_{\mathrm{BS}} \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{\left[\boldsymbol{R}_{\mathrm{BS}}^{\top}(\boldsymbol{p}_{\mathrm{UE}}-\boldsymbol{p}_{\mathrm{BS}})\right]_2}{\left[\boldsymbol{R}_{\mathrm{BS}}^{\top}(\boldsymbol{p}_{\mathrm{UE}}-\boldsymbol{p}_{\mathrm{BS}})\right]_1}\right) \\ \cos^{-1}\left(\frac{\left[\boldsymbol{R}_{\mathrm{BS}}^{\top}(\boldsymbol{p}_{\mathrm{UE}}-\boldsymbol{p}_{\mathrm{BS}})\right]_3}{d}\right) \end{bmatrix}. \tag{3.2}$$

The computation of the AoDs is essential, as they are important for analyzing the behavior of the AoD steering vector (see section 3.1.3 for details), and for evaluating positioning performance bounds, which will be addressed in section 3.2. Note that since Sionna RT defines the horizontal plane along the x-axis and the elevation plane along the z-axis, the formulas have been adjusted to align with Sionna's path data generation framework.

3.1.2 Transmitted Signal Model

Communication between BSs and UEs involves the transmission of radio signals through the channel. In this case, it is assumed that the BS transmits OFDM signals, which is a transmission scheme where data is divided across a total of K subcarriers and L OFDM symbols [15]. The subcarriers are spaced equally by a subcarrier spacing (SCS) of Δ_f. Each OFDM symbol, thus, represents a snapshot in the time domain with an elementary symbol duration of $\mathrm{T}_{\mathrm{sym}} = \frac{1}{\Delta_f}$, accounting for multiple parallel subcarrier transmissions. It is also assumed that each subcarrier is modulated individually using quadrature phase shift keying (QPSK) with a modulation rate of 2 bits per symbol [16]. QPSK is a widely used modulation scheme in communication systems, where data is encoded by shifting the phase of a carrier signal among four distinct values. This allows each symbol to represent two bits of information and will be of importance when evaluating the latency in Section 3.3.

The complex-valued transmitted signal during the l^{th} transmission on the k^{th} subcarrier is denoted $s_{k,l}$ and its magnitude is expressed as

$$|s_{k,l}| = \sqrt{\frac{\mathrm{P}_{\mathrm{tot}}}{\mathrm{KN}_{\mathrm{BS}}}}, \tag{3.3}$$

where $\mathrm{P}_{\mathrm{tot}}$ is the total transmitted power per OFDM symbol.

3.1.3 Received Signal Model and Channel Modeling

The transmitted signals departing from the BS experience varying amounts of path loss due to factors such as the distance between the communicating systems, the presence of obstacles, and the level of noise present within the channel. Hence, the signals will arrive at the UE with attenuated power and lower signal-to-noise ratio (SNR). The received signal, at subcarrier k and time instance l, can be modeled as $y_{k,l} = \tilde{y}_{k,l} + n_{k,l}$, where $\tilde{y}_{k,l}$ is the received noise-free signal and $n_{k,l}$ is the AWGN caused by the thermal noise of the electronics at the UE.

Received Noise-Free Signal

The received noise-free signal model is formulated as $\tilde{y}_{k,l} = \boldsymbol{h}_{k,l}\boldsymbol{f}_l s_{k,l}$, where $\boldsymbol{f}_l$ denotes the precoder and $\boldsymbol{h}_{k,l}$ represents the channel model. Precoders apply specific phase shifts to the transmitted signals from each antenna at the BS, and when these phase shifts are appropriately configured, the transmitted energy is concentrated into a directional beam. Conversely, if the phase shifts are randomly assigned, the resulting beams will lack directional focus. Here, such randomization is intentionally applied, as stated in the Scope 1.3, by defining the precoder as $f_{n,l} = e^{-j2\pi\tau_n}$, where τ_n describes the randomized delay of the n^{th} antenna element, drawn from a uniform distribution.

Channel models are essential for simulating wireless propagation environments in both research and optimization contexts. They provide a structured representation of the radio channel and enable the derivation of different positioning and communication KPIs, which is discussed in greater detail in Section 3.2. The channel model used for this project is defined as

$$\mathbf{h}_{k,l} = \sum_{g=0}^{\mathrm{G}-1} \alpha_g e^{-j2\pi k \Delta_f \tau_g} \mathbf{a}_{\mathrm{BS}}^{\top}\left(\theta'_{\mathrm{BS}_g}, \varphi'_{\mathrm{BS}_g}\right), \tag{3.4}$$

where $\alpha_g = \frac{\lambda}{4\pi d} e^{-j2\pi f_c \tau_g}$ represents the complex channel gain of the g^{th} path, which depends on the distance d between the BS and UE, as well as the propagation delay $\tau_g = \frac{d}{c}$. The AoD steering vector is expressed as

$$\begin{aligned} \mathbf{a}_{\mathrm{BS}}^{\top}\left(\theta'_{\mathrm{BS}_g}, \varphi'_{\mathrm{BS}_g}\right) &= e^{-j2\pi f_c \boldsymbol{p}_n^{\top} \boldsymbol{u}(\theta'_{\mathrm{BS}_g}, \varphi'_{\mathrm{BS}_g})/c}, \\ \boldsymbol{u}(\theta'_{\mathrm{BS}_g}, \varphi'_{\mathrm{BS}_g}) &= \begin{bmatrix} \cos(\theta'_{\mathrm{BS}_g})\sin(\varphi'_{\mathrm{BS}_g}) \\ \sin(\theta'_{\mathrm{BS}_g})\sin(\varphi'_{\mathrm{BS}_g}) \\ \cos(\varphi'_{\mathrm{BS}_g}) \end{bmatrix}. \end{aligned} \tag{3.5}$$

It is worth noting that the complex channel gain generated by Sionna RT is expressed as $\boldsymbol{\alpha}_{\text{sionna}} = \alpha \mathbf{a}_{\mathrm{BS}}$, indicating that Sionna combines the AoD steering vector with the channel gain. Hence, to extract the AoD steering vector from Sionna's combined channel gain, the magnitude of the channel gain is initially computed as $\alpha_{\text{mag}} = |\alpha_{\text{sionna}}| = \text{mean}(|\alpha \mathbf{a}_{\mathrm{BS}}|)$. Subsequently, the phase component of the channel gain, $e^{-j2\pi f_c \tau}$, is used to finally extract the AoD steering vector from Sionna's mixed channel gain by the following relation

$$\mathbf{a}_{\mathrm{BS}} = \frac{\boldsymbol{\alpha}_{\text{sionna}}}{\alpha} = \frac{\boldsymbol{\alpha}_{\text{sionna}}}{\alpha_{\text{mag}} e^{-j2\pi f_c \tau}}. \tag{3.6}$$

Noise Model - AWGN

The noise model used to describe the received signal is the AWGN. This category of noise has three fundamental characteristics [17]:

- **Additive:** The observed signal is the sum of the true signal and the noise.
- **White:** The noise has a constant power spectral density (PSD) across all frequency bands.
- **Gaussian:** The noise is drawn from a Gaussian distribution.

For this project, the noise is assumed to follow a complex Gaussian distribution $n \sim \mathcal{CN}(0, \sigma^2)$ with a mean $\mu = 0$ and variance $\sigma^2 = \mathrm{K_B T N_f}$, where $\mathrm{K_B}$ is the Boltzmann constant, T is the temperature in unit Kelvin, and $\mathrm{N_f}$ is the noise figure on a linear scale.

3.2 Positioning KPIs

With the use of the system modeling to define the setup, along with appropriate signal models, it becomes possible to evaluate the positioning performance bounds. This section, therefore, aims to describe the methodology for computing the CRLB in the context of an OFDM system under the assumption of an AWGN channel. This could later be integrated with Sionna RT's generated path data to assess the positioning performance for multiple UE positions.

3.2.1 Cramér-Rao Lower Bound

The CRLB offers a benchmark for evaluating the performance of unbiased estimators by setting a lower bound on their covariance matrix [18] [19]. It is particularly useful in determining the minimum achievable variance for parameter estimates, thus assessing the efficiency of different estimation techniques [17] [20]. In this context, it provides the lowest possible variance, achievable by any estimator, for the positioning of the UEs. The variance is based on the known transmitted signals and is expressed as a value of PEB. To calculate the CRLB, it is required to first compute the Fisher information matrix (FIM), which measures the amount of information that an observable random variable carries about unknown parameters of the model. The CRLB relates as the inverse of the FIM, indicating that the FIM forms the basis for computing the CRLB. The necessary computation steps are presented in the pseudocode below.

Algorithm 1 Compute the Cramér-Rao Lower Bound

1: **procedure** COMPUTE CRLB FOR AWGN CASE
2: **Input:** Measurement model $\mathbf{y} = \mathbf{m}(x) + \mathbf{n}$, $\mathbf{n} \sim \mathcal{CN}(0, \mathbf{\Sigma})$, $\mathbf{\Sigma} = \frac{\sigma^2}{2}\boldsymbol{I}$
3: Determine unknown parameters: $\boldsymbol{\eta}_1$
4: Compute partial derivatives: $\nabla_{\eta_1}\mathbf{m} \leftarrow \frac{\partial \mathbf{m}(\boldsymbol{\eta}_1)}{\partial \boldsymbol{\eta}_1}$
5: Compute FIM: $\boldsymbol{J}(\boldsymbol{\eta}_1) = \Re\left\{\nabla_{\eta_1}\mathbf{m}^{\mathrm{H}}\mathbf{\Sigma}^{-1}\nabla_{\eta_1}\mathbf{m}\right\}$
6: Compute CRLB $\rightarrow \boldsymbol{J}^{-1}(\boldsymbol{\eta_1})$
7: **Output:** CRLB matrix: $\mathrm{var}\left(\hat{\boldsymbol{\eta}}_{1_i}\right) \geq \left[\boldsymbol{J}^{-1}(\boldsymbol{\eta}_1)\right]_{i,i}$
8: **if** parameters in $\boldsymbol{\eta}_1$ can be related to other parameters **then**
9: **if** wish to unpack parameter **then**
10: Add unpacked parameter to $\boldsymbol{\eta}_2$
11: Form equations for parameters in $\boldsymbol{\eta}_1$ relating to parameters in $\boldsymbol{\eta}_2$
12: Compute partial derivatives: $\frac{\partial \boldsymbol{\eta}_1}{\partial \boldsymbol{\eta}_2}$
13: Construct Jacobian matrix $\boldsymbol{T}$ with computed partial derivatives
14: Compute FIM of $\boldsymbol{\eta}_2$: $\boldsymbol{J}(\boldsymbol{\eta}_2) = \boldsymbol{T}^{\top}\boldsymbol{J}(\boldsymbol{\eta}_1)\boldsymbol{T}$
15: Compute CRLB $\rightarrow \boldsymbol{J}^{-1}(\boldsymbol{\eta_2})$
16: **Output:** CRLB matrix: $\mathrm{var}\left(\hat{\boldsymbol{\eta}}_{2_i}\right) \geq \left[\boldsymbol{J}^{-1}(\boldsymbol{\eta}_2)\right]_{i,i}$
17: **if** wish to transform further **then**
18: Redo all steps from step 8
19: **else**
20: **end if**
21: **else**
22: **end if**
23: **else**
24: **end if**
25: **end procedure**

3.2.2 Computation process

Following the pseudocode for the computation of the CRLB, the measurement model for this communication channel is the previously defined received signal $y_{k,l} = \tilde{y}_{k,l} + n_{k,l}$. The unknown set of parameters is defined as

$$\boldsymbol{\eta}_1 = \left[\alpha_{\Re_0}, \alpha_{\Im_0}, \tau_0, \theta'_{\mathrm{BS}_0}, \varphi'_{\mathrm{BS}_0}, \ldots, \alpha_{\Re_\mathrm{G}}, \alpha_{\Im_\mathrm{G}}, \tau_\mathrm{G}, \theta'_{\mathrm{BS}_\mathrm{G}}, \varphi'_{\mathrm{BS}_\mathrm{G}}\right] .$$

Note that α has been divided up into a real and imaginary part, as it represents the complex channel gain which is complex-valued. Also, note that since multipath components are present in the channel, $\boldsymbol{\eta}_1$ must account for the unknown parameters of each path individually. However, as mentioned in the Scope 1.3, only the LoS path parameters ($g = 0$) will be further transformed in the upcoming procedure when relating the channel parameters to positioning parameters.

The next step is to compute the partial derivatives of the noise-free received signal, for every subcarrier and OFDM symbol, with respect to every parameter in $\boldsymbol{\eta}_1$, i.e.

$$\nabla_{\boldsymbol{\eta}_1} \tilde{y}_{k,l} = \left[\frac{\partial \tilde{y}_{k,l}}{\partial \alpha_{\Re_0}}, \frac{\partial \tilde{y}_{k,l}}{\partial \alpha_{\Im_0}}, \frac{\partial \tilde{y}_{k,l}}{\partial \tau_0}, \frac{\partial \tilde{y}_{k,l}}{\partial \theta'_{\mathrm{BS}_0}}, \frac{\partial \tilde{y}_{k,l}}{\partial \varphi'_{\mathrm{BS}_0}}, \ldots, \frac{\partial \tilde{y}_{k,l}}{\partial \alpha_{\Re_\mathrm{G}}}, \frac{\partial \tilde{y}_{k,l}}{\partial \alpha_{\Im_\mathrm{G}}}, \frac{\partial \tilde{y}_{k,l}}{\partial \tau_\mathrm{G}}, \frac{\partial \tilde{y}_{k,l}}{\partial \theta'_{\mathrm{BS}_\mathrm{G}}}, \frac{\partial \tilde{y}_{k,l}}{\partial \varphi'_{\mathrm{BS}_\mathrm{G}}}\right] .$$

The respective partial derivatives are listed in Appendix A.

From the calculated partial derivatives, the summation of each individually constructed FIM, over all subcarriers and OFDM symbols, is performed to obtain the complete FIM of $\boldsymbol{\eta}_1$. At this stage, the CRLB could be computed directly by applying the inverse operation on the FIM. However, the unknown set of parameters in $\boldsymbol{\eta}_1$ can be transformed further into parameters of interest, i.e. positioning parameters. Therefore, the LoS parameters except for α are transformed as follows

$$\tau(x, y, z) = \frac{\sqrt{\Delta x^2_{\mathrm{UE,BS}} + \Delta y^2_{\mathrm{UE,BS}} + \Delta z^2_{\mathrm{UE,BS}}}}{c} , \tag{3.7}$$

$$\theta'_{\mathrm{BS}}(x, y, z) = \tan^{-1}\left(\frac{\left[\boldsymbol{R}^\top_{\mathrm{BS}}\left([\Delta x_{\mathrm{UE,BS}}, \Delta y_{\mathrm{UE,BS}}, \Delta z_{\mathrm{UE,BS}}]^\top\right)\right]_2}{\left[\boldsymbol{R}^\top_{\mathrm{BS}}\left([\Delta x_{\mathrm{UE,BS}}, \Delta y_{\mathrm{UE,BS}}, \Delta z_{\mathrm{UE,BS}}]^\top\right)\right]_1}\right) , \tag{3.8}$$

$$\varphi'_{\mathrm{BS}}(x, y, z) = \cos^{-1}\left(\frac{\left[\boldsymbol{R}^\top_{\mathrm{BS}}\left([\Delta x_{\mathrm{UE,BS}}, \Delta y_{\mathrm{UE,BS}}, \Delta z_{\mathrm{UE,BS}}]^\top\right)\right]_3}{\sqrt{\Delta x^2_{\mathrm{UE,BS}} + \Delta y^2_{\mathrm{UE,BS}} + \Delta z^2_{\mathrm{UE,BS}}}}\right) . \tag{3.9}$$

From $\boldsymbol{\eta}_1$, the parameters $\alpha_\Re$ and $\alpha_\Im$ are directly inserted into $\boldsymbol{\eta}_2$. Thus, $\boldsymbol{\eta}_2 = [\alpha_{\Re_0}, \alpha_{\Im_0}, x_0, y_0, z_0, \ldots, \alpha_{\Re_\mathrm{G}}, \alpha_{\Im_\mathrm{G}}, \tau_\mathrm{G}, \theta'_{\mathrm{BS}_\mathrm{G}}, \varphi'_{\mathrm{BS}_\mathrm{G}}]$ is formed. As a result of this operation, the Jacobian matrix can be constructed with the following formula:

$$\boldsymbol{T} = \mathrm{blkdiag}(\boldsymbol{T}_1, \boldsymbol{T}_2) , \tag{3.10}$$

where $\boldsymbol{T}_1$ describes the Jacobian of the LoS parameters, and $\boldsymbol{T}_2$ the Jacobian of the rest of the unaltered channel parameters.

The LoS Jacobian, $\boldsymbol{T}_1$, can be expressed as

$$\boldsymbol{T}_1 = \begin{bmatrix} \frac{\partial \alpha_{\Re_0}}{\partial \alpha_{\Re_0}} & \frac{\partial \alpha_{\Re_0}}{\partial \alpha_{\Im_0}} & \frac{\partial \alpha_{\Re_0}}{\partial x_0} & \frac{\partial \alpha_{\Re_0}}{\partial y_0} & \frac{\partial \alpha_{\Re_0}}{\partial z_0} \\ \frac{\partial \alpha_{\Im_0}}{\partial \alpha_{\Re_0}} & \frac{\partial \alpha_{\Im_0}}{\partial \alpha_{\Im_0}} & \frac{\partial \alpha_{\Im_0}}{\partial x_0} & \frac{\partial \alpha_{\Im_0}}{\partial y_0} & \frac{\partial \alpha_{\Im_0}}{\partial z_0} \\ \frac{\partial \tau_0}{\partial \alpha_{\Re_0}} & \frac{\partial \tau_0}{\partial \alpha_{\Im_0}} & \frac{\partial \tau_0}{\partial x_0} & \frac{\partial \tau_0}{\partial y_0} & \frac{\partial \tau_0}{\partial z_0} \\ \frac{\partial \theta'_{\mathrm{BS}_0}}{\partial \alpha_{\Re_0}} & \frac{\partial \theta'_{\mathrm{BS}_0}}{\partial \alpha_{\Im_0}} & \frac{\partial \theta'_{\mathrm{BS}_0}}{\partial x_0} & \frac{\partial \theta'_{\mathrm{BS}_0}}{\partial y_0} & \frac{\partial \theta'_{\mathrm{BS}_0}}{\partial z_0} \\ \frac{\partial \varphi'_{\mathrm{BS}_0}}{\partial \alpha_{\Re_0}} & \frac{\partial \varphi'_{\mathrm{BS}_0}}{\partial \alpha_{\Im_0}} & \frac{\partial \varphi'_{\mathrm{BS}_0}}{\partial x_0} & \frac{\partial \varphi'_{\mathrm{BS}_0}}{\partial y_0} & \frac{\partial \varphi'_{\mathrm{BS}_0}}{\partial z_0} \end{bmatrix}, \tag{3.11}$$

where the partial derivatives are listed in Appendix A. The Jacobian of the unaltered parameters, $\boldsymbol{T}_2$, is a unit matrix with dimensions $\boldsymbol{T}_2 \in \Re^{5(\mathrm{G}-1)\times 5(\mathrm{G}-1)}$, as they are simply derivatives with respect to themselves and are unrelated to the other parameters listed in $\boldsymbol{\eta}_2$. Based on these computations, the final FIM can be formulated as

$$\boldsymbol{J}(\boldsymbol{\eta}_2) = \boldsymbol{T}^\top \boldsymbol{J}(\boldsymbol{\eta}_1)\boldsymbol{T}\,, \tag{3.12}$$

and thus, the CRLB can be computed by $\boldsymbol{J}^{-1}(\boldsymbol{\eta}_2)$. The PEB is then obtained by the following operation

$$\mathrm{PEB} \geq \sqrt{\mathrm{trace}\left[\boldsymbol{J}^{-1}\left(\boldsymbol{\eta_2}\right)\right]_{3:5,3:5}}\,, \tag{3.13}$$

which is the summation of the errors in the x, y, and z from the specified diagonal elements of the CRLB. The term PEB refers to a fundamental limit of localization accuracy and is a measure used to quantify the maximum possible error that can ever be achieved by any unbiased estimator, in the estimated position of a target [21]. This concept is particularly relevant in fields such as wireless sensor networks and navigation systems as it helps ensure the accuracy and reliability of positioning systems.

3.3 Communication KPIs

In addition to the positioning KPI, several essential communication KPIs can be evaluated as well. These performance metrics determine the reliability of cellular networks, which is critical for ensuring consistent and reliable service [16] [22]. The KPIs are also used to optimize network performance by minimizing drive tests and leveraging user feedback to enable self-organization of the network [23] [15]. This approach reduces costs associated with manpower and equipment, which accentuates the importance of communication KPIs. In this section, two of the most commonly evaluated communication KPIs, in the field of communication systems, will be derived and discussed.

3.3.1 Latency

Latency can be described as the time required for the transmission of a packet over a channel, including the time for decoding the packet [24]. It is computed by combining various types of delays experienced by a signal, which, by the scope of this project, include the propagation delay $\mathrm{D_{PR}}$ and the transmission delay $\mathrm{D_T}$. The propagation delay represents the time required for a packet to cross the transmission medium between a BS and a UE [25]. This delay depends solely on the distance between the communicating systems and is expressed by

$$\mathrm{D_{PR}} = \frac{d}{c} = \tau \,. \tag{3.14}$$

The transmission delay, on the other hand, refers to the time required to push all the data into the transmission medium [25], in this case, the air. It depends on several factors and is formulated as

$$\mathrm{D_T} = \left\lceil \frac{\mathrm{X}}{\mathrm{Q\ K}} \right\rceil \frac{1}{\Delta_f} \,, \tag{3.15}$$

where X denotes the number of transmitted bits, and Q represents the digital modulation rate. The ceiling function is applied because even a non-integer ratio would necessitate a full transmission interval. Given the transmission and propagation delays, the total latency, $\mathrm{D_{TOT}}$ can be computed by

$$\mathrm{D_{TOT}} = \mathrm{D_{PR}} + \mathrm{D_T} \,. \tag{3.16}$$

3.3.2 Capacity

In communication systems, capacity is defined as the maximum achievable rate of information exchange over a communication channel, considering all possible transmission and detection techniques [26]. It describes the amount of data that can be transmitted between devices. Under the assumption of an AWGN channel, the maximum capacity is given by the Shannon-Hartley theorem [27] expressed as follows

$$C = \sum_{k=1}^{\mathrm{K}} \Delta_f \log_2(1 + \mathrm{SNR}_k) \,, \tag{3.17}$$

where SNR_k denotes the SNR of the k^{th} subcarrier. Each subcarrier experiences a unique SNR, which implies the summation of their respective contributions to the total capacity. The SNR quantifies the strength of the received signal relative to the background noise power [27], and is defined by the following expression

$$\mathrm{SNR}_k = \frac{|H(f_k)|^2 \, \mathrm{P_{avg}}}{\sigma^2 \Delta_f} \,, \tag{3.18}$$

with H representing the CFR of the k^{th} subcarrier frequency, and $\mathrm{P_{avg}} = \frac{\mathrm{P_{tot}}}{\mathrm{K}}$ the average transmission power.

4
Results and Discussion

The positioning and communication KPIs were evaluated based on the generated data from Sionna RT and the methodology formulated in Chapter 3. For positioning performance, heatmaps of the computed PEBs are visualized, leaving room for analysis of how factors such as distance and complexity of the radio environment could affect wave propagation and positioning bounds. Regarding communication performance, MATLAB plots are shown to illustrate how different configuration parameters affect each performance metric individually.

4.1 Positioning Performance

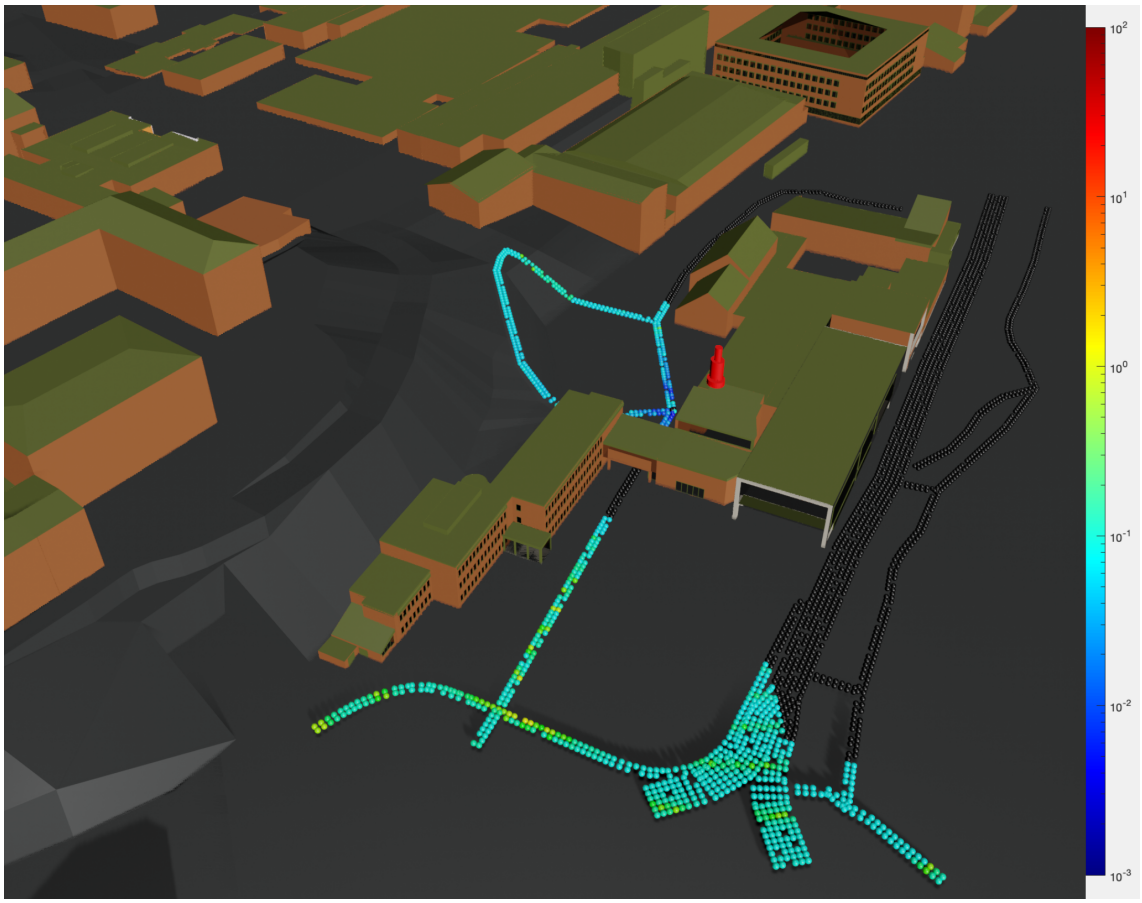
For the evaluation of positioning performance, the path data provided by Sionna RT was used in the computation of the PEBs for a set of grid points at the Chalmers entrance. The respective PEBs were computed under fixed conditions specified in Table 4.1. From this, two distinct scenarios were visualized:

- Scenario 1: The BS placed on top of the Chalmers entrance, see Figure 4.1.
- Scenario 2: The BS placed on top of "Fysikhuset", see Figure 4.2.

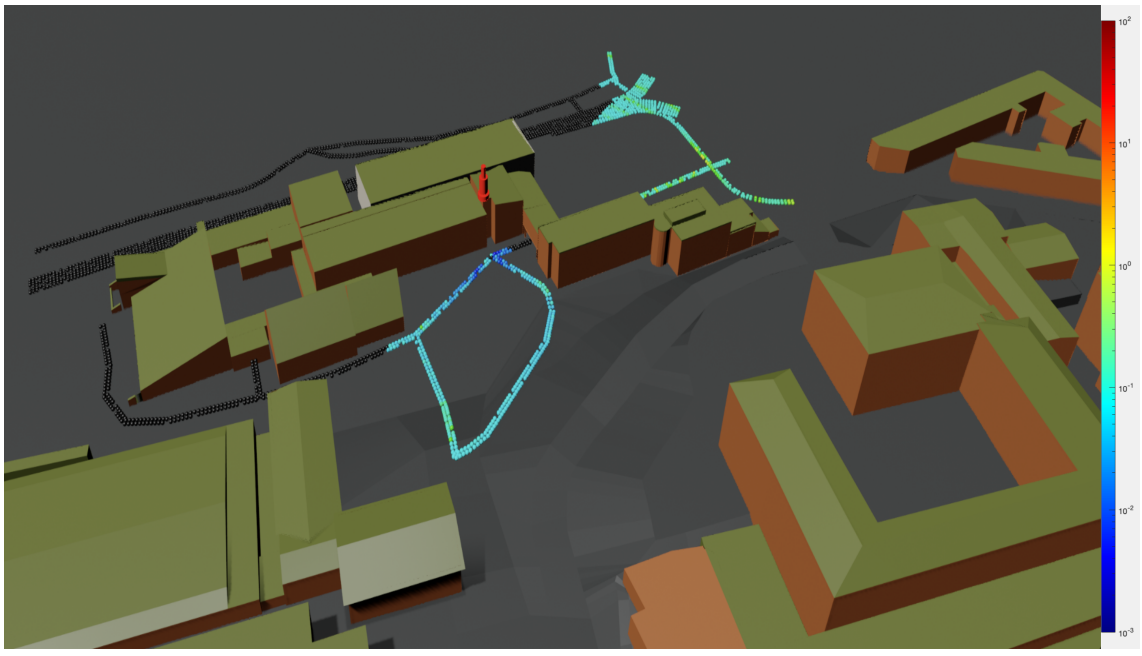
In both scenarios, the PEB values are observed from two different angles to provide a clearer overview of the spatial variations in the heatmaps. The BSs are represented by the red tower-shaped structures placed on top of the buildings, and the UEs are indicated by colored dots distributed along the roads of the digital twin. The colorbar visualizes the PEBs for the different UE points on a logarithmic scale. The red points correspond to greater PEB, and the blue points correspond to lower PEB. The black UE points represent NLoS paths and out-of-range signal, and therefore, have no computed PEB. To clarify, the black points could still be receiving signal from the BS, similar to the other points; however, they are not in LoS.

Table 4.1: Default configuration of the scenarios created in Blender.

Parameter	Sub-carriers	OFDM symbols	Transmission power [dBm]	Carrier frequency [MHz]	SCS [kHz]	Total number of UEs
Value	3300	14	10	3500	15	2816

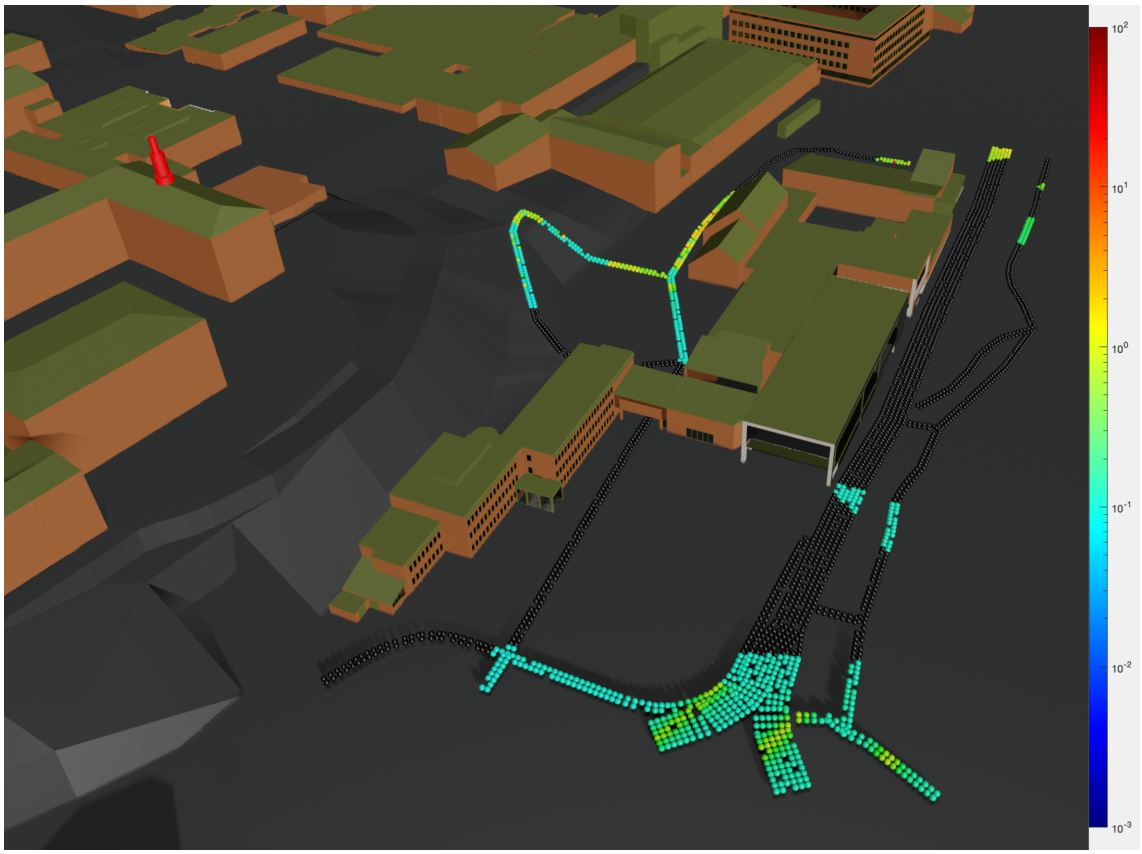


(a) Angle from outside of the Chalmers entrance.

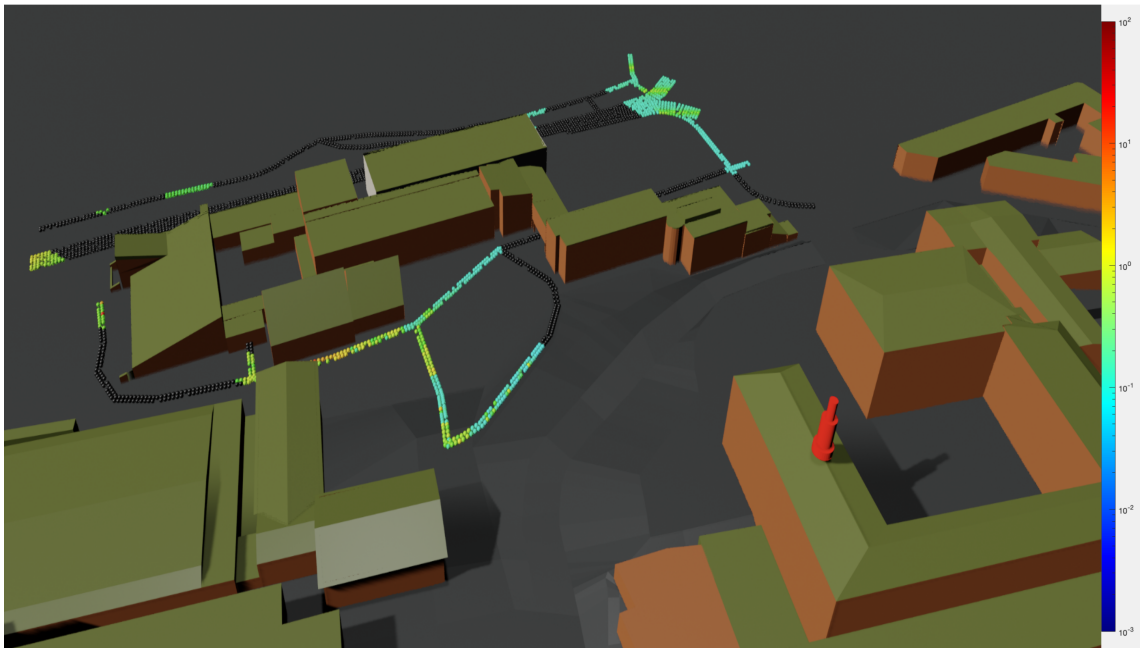


(b) Angle from inside of the Chalmers entrance.

Figure 4.1: Visualization of scenario 1 where the BS is positioned on top of the Chalmers entrance.



(a) Angle from outside of the Chalmers entrance.



(b) Angle from inside of the Chalmers entrance.

Figure 4.2: Visualization of scenario 2 where the BS is positioned on top of "Fysikhuset".

The figures for both scenarios demonstrate that the placement of the BS has a significant impact on the resulting PEBs. In general, UEs situated closer to the BSs exhibit lower PEB values, which suggests that a shorter distance between the communicating devices improves localization accuracy. This observation was further supported by Figure 4.3, where it is shown that in 90% of the cases in Scenario 1, when the BS was placed in close proximity to the UE, the PEB was 0.31 meters or lower. In contrast, for Scenario 2, when the BS was on average situated farther away from the UEs, the PEB reached up to 1.86 meters for the same percentage of cases. These results, therefore, accentuate the impact of distance on localization accuracy.

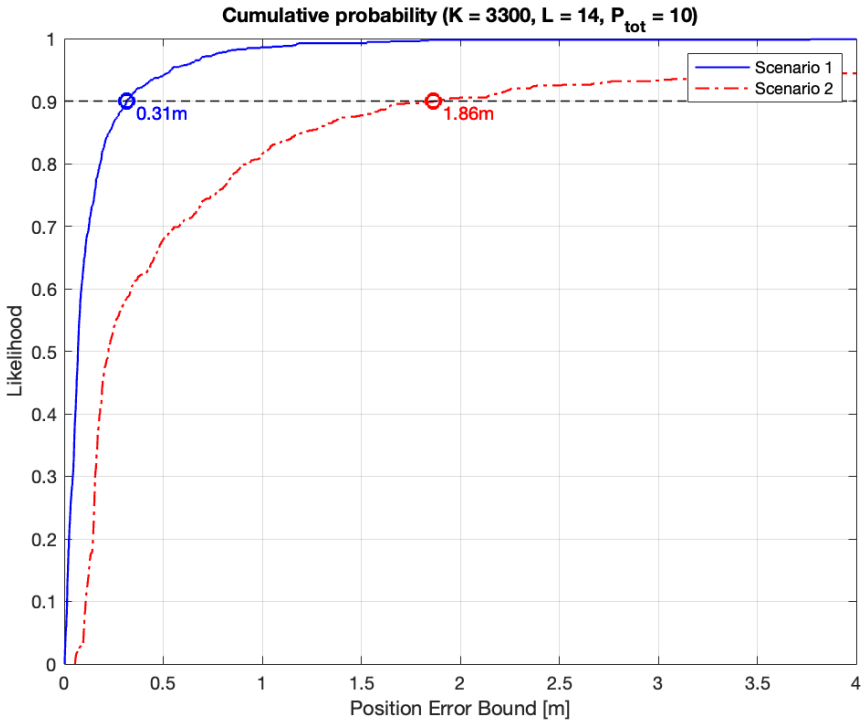


Figure 4.3: Cumulative probability for both scenarios.

However, distance did not seem to be the only contributing factor to the observed variations in PEBs. In Figure 4.4, where a randomly selected UE was investigated, it is evident that a number of subcarriers below 2000 contribute to a lower PEB for scenario 2, although the distance between the BS and the specific UEs is approximately 173.3 meters compared to scenario 1's 96.4 meters. This observation could be a result of the complex radio environment in which the communicating systems are situated, where multiple reflections of walls, roads, and other structures could all be possibilities. To confirm this, the propagation paths for both scenarios were simulated in Sionna RT in the presence of the specific UE, as illustrated in Figure 4.5. From the figure, it is evident that the BS in scenario 1 exhibits multipath signals with multiple reflections of the buildings and the roads, whereas the BS in scenario 2 only exhibits a LoS signal. This could, therefore, cause resolvability issues, which in turn could lead to challenges in distinguishing between individual propagation paths and hence degrade the localization performance. Nevertheless, when exceeding 2000 subcarriers, this trend is reversed, and the PEB for Scenario 1 surpasses that of Scenario 2, which again is confirmed by Figure 4.4. This behaviour may be attributed to the increased resolution in the system, leading to resolvable propagation paths for Scenario 1 and thus lower PEB. These observations, therefore, support the idea that positioning performance is not solely confined to distance but also depends on the system's resolution, which provides useful insight into how distance is only meaningful to address when there is enough resolution in the system.

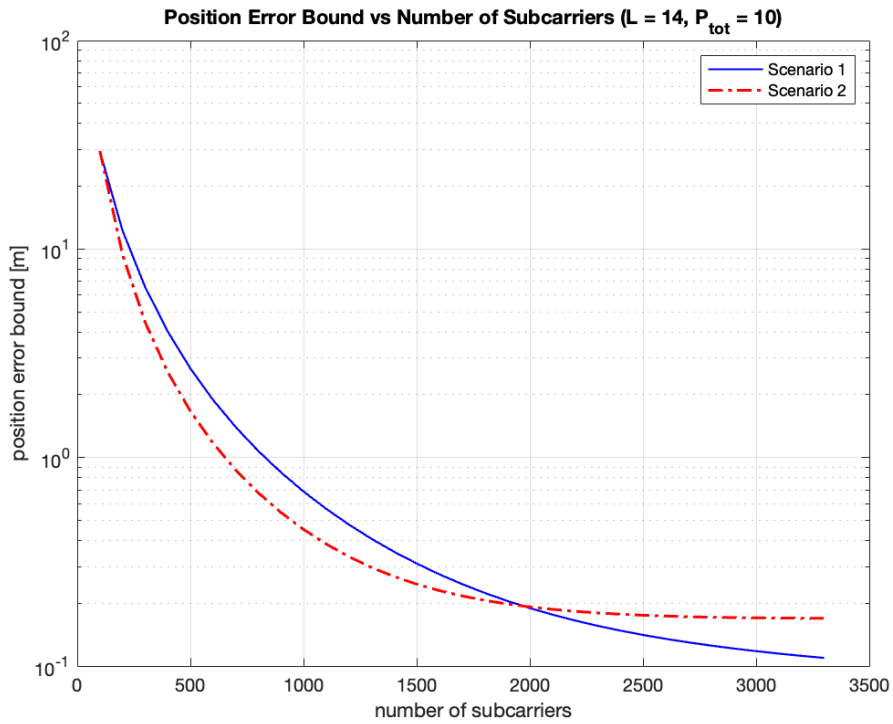


Figure 4.4: PEB in relation to the number of subcarriers for the same UE for both scenarios.

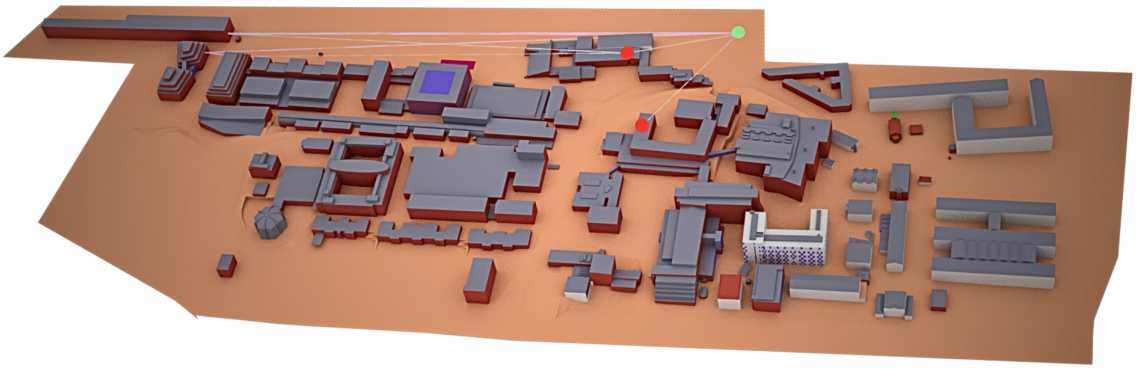


Figure 4.5: Visualization of the environmental complexity with BSs (red points) for both scenarios and a single UE (green point). The paths of the signals are the white lines connecting the BSs to the UE.

Another interesting observation made was the relation between coverage and PEB. In Scenario 2, although the more elevated BS generally contributed to increased PEBs, it still managed to cover more UEs, resulting in a total of 2772 covered UEs, compared to Scenario 1's 1754, which can be seen in Table 4.2. This observation highlights the trade-off between localization precision and coverage area when determining optimal BS placements. Elevating the BS, assuming it remains unobstructed, tends to lead to worse localization performance due to the increased distance between the transmitting and receiving devices. However, such action seems to improve coverage by extending the number of UEs within the coverage area, and hence still turns out to be beneficial. To achieve wider coverage and maintain reasonable localization accuracy, it may be necessary to deploy more BSs in the area to cover a larger set of UE points. On the other hand, this approach would require spatial and financial resources, and would possibly cause side effects in terms of interference between the different BSs, among other effects.

Table 4.2: Difference between the number of UEs covered by the BS for both scenarios.

Variable	Number of LoS UEs	Total number of UEs	Average distance [m]
Scenario 1	920	1754	75.95
Scenario 2	879	2772	169.64

When examining the effects of transmission power and number of OFDM symbols on the resulting PEB for the randomly selected UE, it was evident that an increase in both parameters resulted in a lower PEB, see Figure 4.6 and Figure 4.7. This seems to hold for both scenarios, which is expected since a greater power within the signal yields a higher SNR, and the more times the signal is transmitted, the higher its reliability. Furthermore, it is evident that for the given number of subcarriers, the BS positioned closer to the UE consistently achieves a lower PEB. This could be attributed to the larger number of subcarriers providing resolvability of the multi-path signal components. Reducing the number of subcarriers would likely preserve the descending trend in PEB, however, the scenario exhibiting lower PEB in the figures would presumably be reversed due to resolvability challenges arising within the complex radio environment.

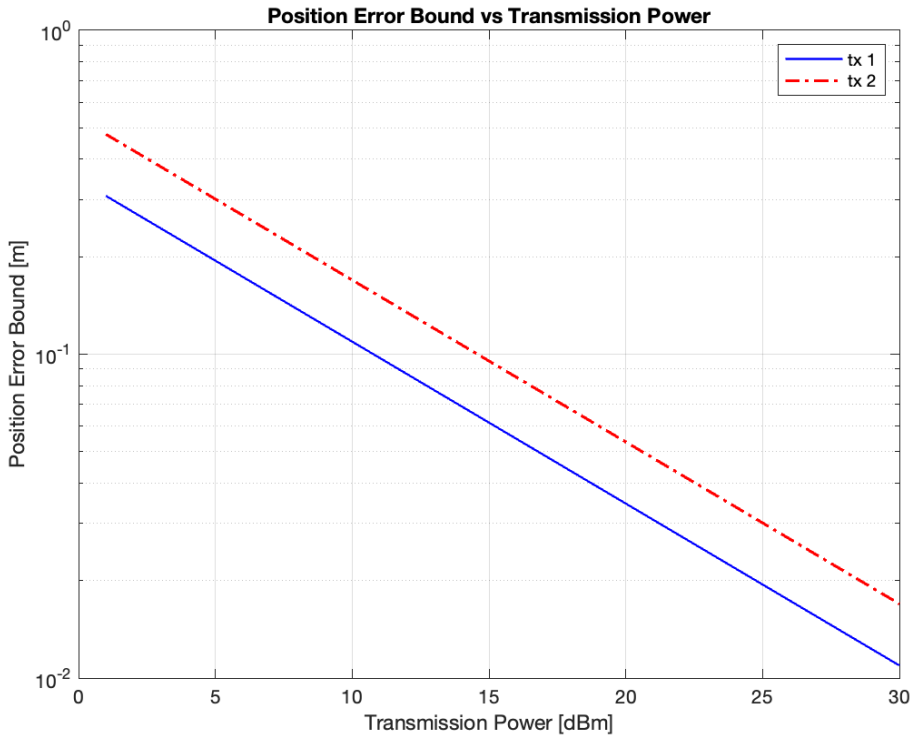


Figure 4.6: PEB in relation to the transmission power for the same UE for both scenarios.

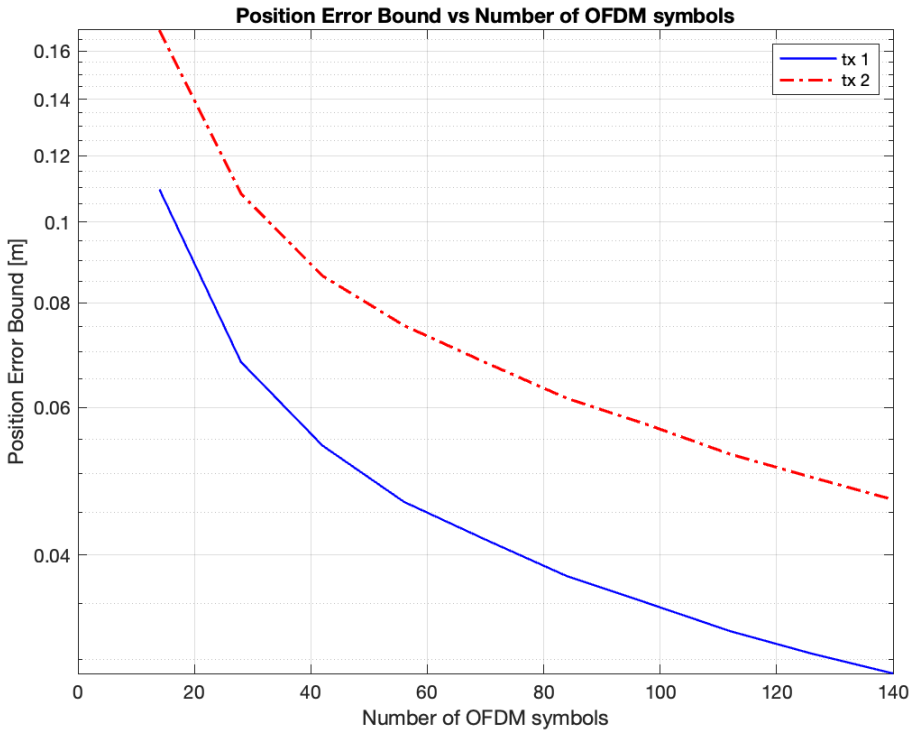


Figure 4.7: PEB in relation to the number of OFDM symbols for the same UE for both scenarios.

Given the simplified nature of the project, these results indicate several key factors that contribute to achieving low PEB values. Specifically, performance improves with a large number of subcarriers, OFDM symbols, increased transmission power, and a short distance between the BS and the corresponding UE. However, minimizing distance alone does not necessarily guarantee improved positioning performance. In complex environments such as Chalmers, the system must also have enough resolution to achieve reasonable positioning bounds. Moreover, rather than focusing solely on proximity, it may be more effective to strike a balance between minimizing the PEB and ensuring adequate coverage when determining the placement of BSs.

4.2 Communication Performance

The impact of different input parameters such as transmission power, distance, number of bits, and subcarriers on the communication KPIs, capacity, and latency was studied for the communication performance metrics.

Figure 4.8 illustrates that capacity exhibits a linear relationship with transmission power, which is expected as an increase in transmission power leads to higher SNR, thereby improving overall signal reception. Consequently, increasing transmission power could enhance the amount of data that can be transmitted. However, increasing transmission power might be costly in terms of resources, potentially rendering such an approach inefficient in terms of resource allocation.

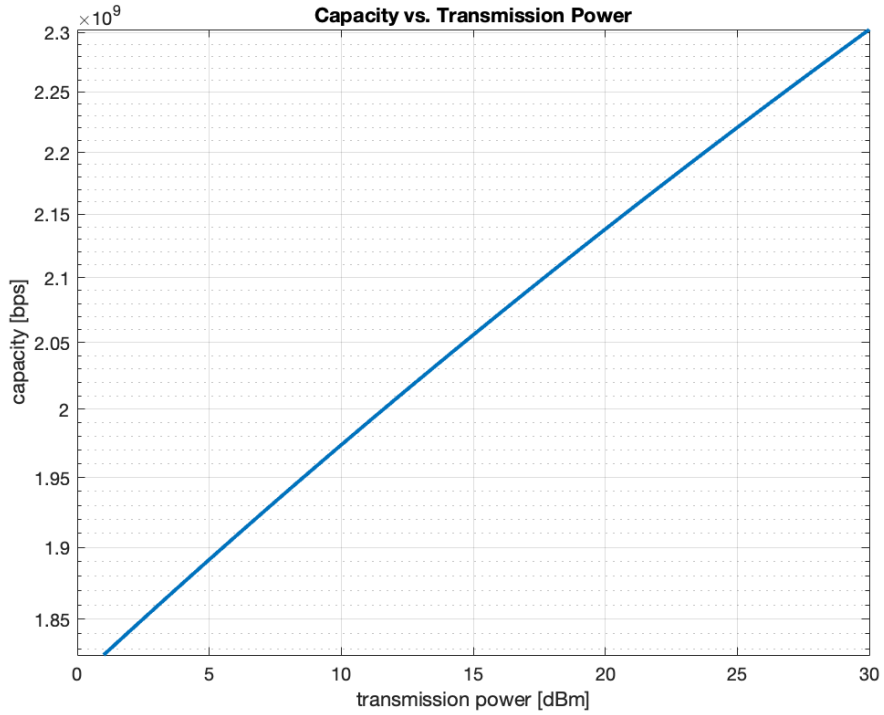


Figure 4.8: Capacity over transmission power.

The evaluation of latency involves a greater number of adjustable input parameters, making it more complex to assess. As shown in Figure 4.9, the latency increases linearly with distance due to propagation delay, which depends solely on the distance between the communicating systems. In this case, the transmission delay remains constant, as it is determined by the rate at which data is pushed into the transmission medium rather than the physical distance between the devices. Thus, it is reasonable to expect that the latency, as defined in Section 3.2, varies linearly with distance. However, an important observation is that the magnitudes of the delays, as illustrated in the figure, are not noticeable for such small differences in distances. For example, a UE located 100 meters apart from a BS would approximately experience the same propagation delay as the one 10 meters apart, i.e., 0.35 μs compared to 0.025 μs. Hence, it can be stated that the effect of propagation delay is not remarkable when comparing short differences in distances.

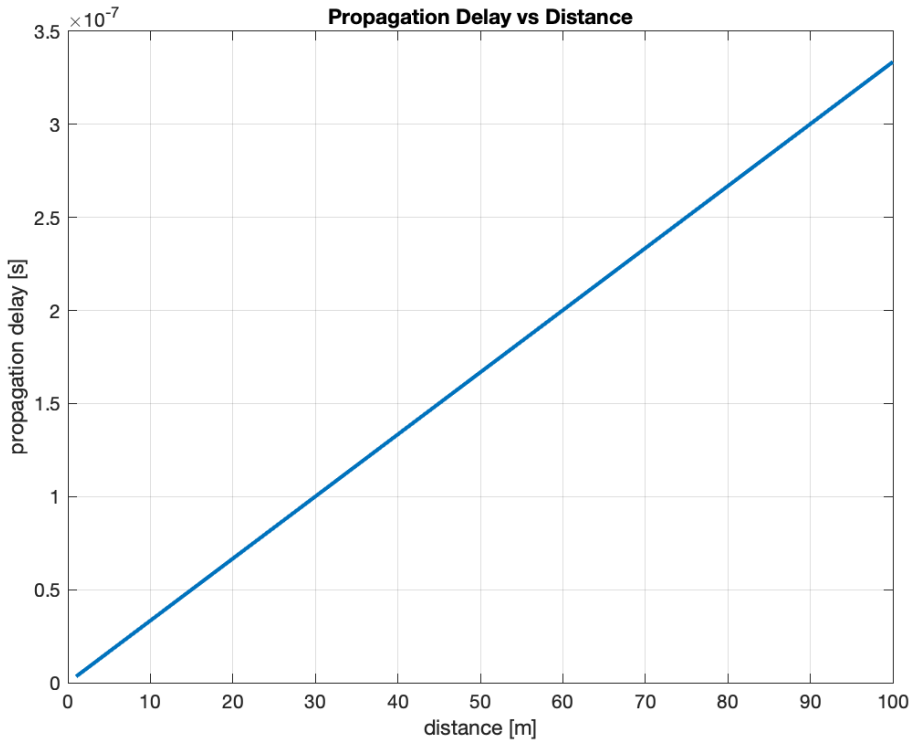
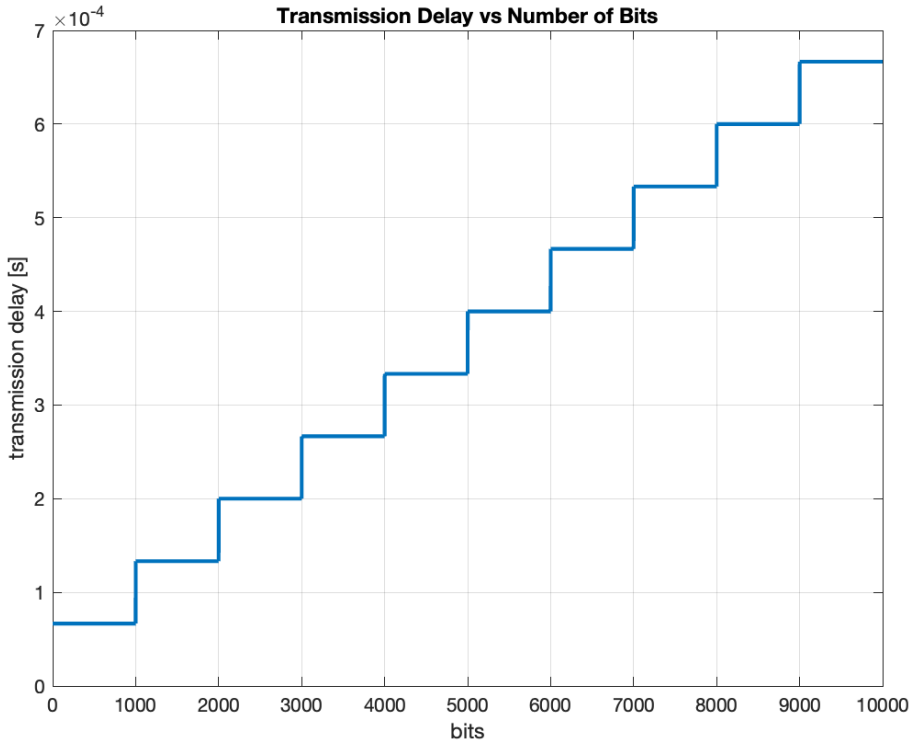


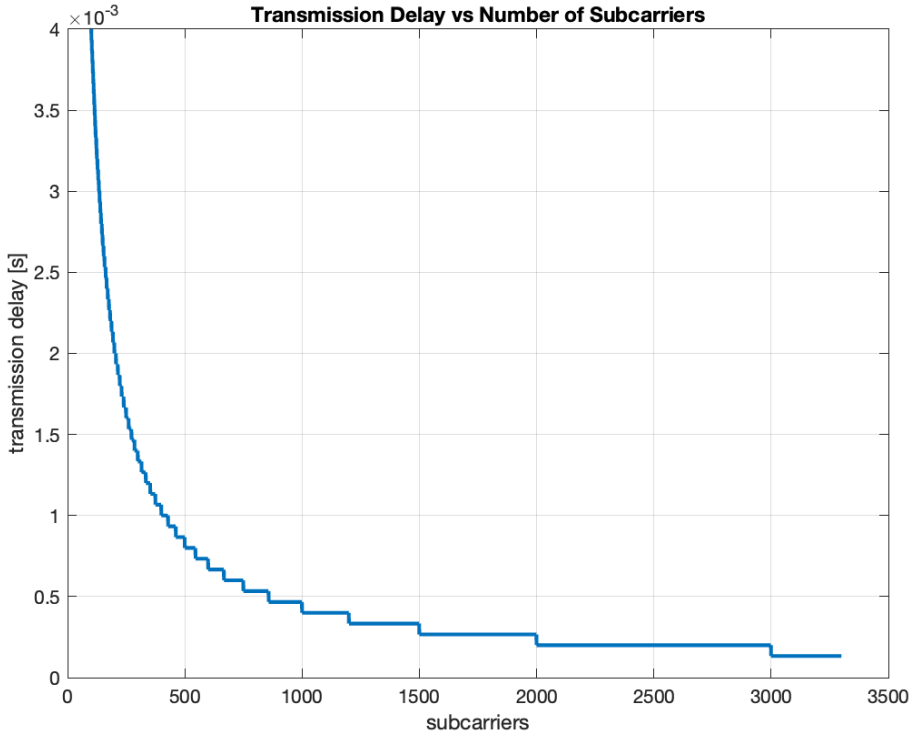
Figure 4.9: Propagation delay variation over distance.

However, when adjusting the parameters related to transmission delay, such as the total number of bits and subcarriers, the relationship no longer remains linear, see Figure 4.10. As illustrated in 4.10a, when varying the total number of bits, while keeping the number of subcarriers, SCS, and modulation rate constant at 3300, 15 kHz, and 2 bits/symbol, respectively, it is apparent that the overall latency increases. This outcome is expected, as the transmission of more data places greater demands on the OFDM system by requiring more time instances to convey all the information. It is also observable that, within a certain interval, the latency remains constant despite increases in the number of bits, which suggests that variations in bit volume do not necessarily lead to changes in latency. Latency is affected only when the number of symbols exceeds multiples of the number of subcarriers for a given time instance l.

In Figure 4.10b, adjusting the subcarriers used, while keeping the total number of bits, SCS, and modulation rate constant at 12000 bits, 15 kHz, and 2 bits/symbol, respectively, it is noticeable that an increase in subcarriers at a single time instance contributes to lower overall latency. This outcome is anticipated, as a higher number of subcarriers allows more data to be transmitted simultaneously, shortening the total transmission duration. However, compared to variations in the total number of bits transmitted, changes in the number of subcarriers, particularly for a lower number of subcarriers, have a more severe impact on the transmission delay, while for higher numbers, not as much. This emphasizes that after a certain point, no matter how much the number of subcarriers is increased, the transmission delay stays almost constant.



(a) Transmission delay variation over number of bits.



(b) Transmission delay variation over number of subcarriers.

Figure 4.10: Latency variation due to changes in transmission delay variables.

Given the results, it is evident that transmission delay is the main contributor to overall latency. This is supported by the relative magnitudes of the delays, where the transmission delay is in the order of milliseconds, whereas the propagation delay is in microseconds. However, one could argue that if the distance between the BS and the UE were sufficiently large, the propagation delay would become the dominant term in the overall latency. For this to be the case, the distance between the communicating devices would have to be approximately 80 km, which is a distance where the signal would most likely fail to reach its target due to severe path loss.

5
Conclusions

In this thesis, the design and implementation of a Digital Radio Twin of the Chalmers University campus are presented, serving as a simulation pipeline for evaluating communication and positioning systems. Realistic radio environments were constructed by integrating 3D modeling in Blender and ray-tracing simulations using the Sionna RT library in Python. This has been done to study the effects of varying system parameters on KPIs, such as latency, capacity, and PEB.

The results highlight that different parameters such as the number of subcarriers, number of OFDM symbols, transmission power, and distance all affect the KPIs in different ways, some more drastically than others. A particularly notable finding was that localization performance was influenced not only by the distance between the communication systems but also by the system's resolution, given the complex radio environment. In addition, it was evident that latency was primarily affected by transmission delay, where it was on the order of milliseconds compared to microseconds for the propagation delay.

The developed pipeline could provide a foundation for future research in the study, development, and optimization of communication networks by offering a tool for simulating complex and realistic scenarios. Its automated structure reduces the need for manual calculations, thereby enhancing accuracy and efficiency in simulation workflows.

Future development could involve enhancing the digital radio environment with increased architectural and material detail to support even more realistic simulation scenarios. Incorporating dynamic UEs would allow for the modeling of real-life user movement, while the use of moving agents could simulate crowded environments, enabling more representative and accurate performance evaluations. Furthermore, a grid field could be placed on the rooftops of buildings within the digital environment, allowing the simulation to iterate over different base station placements to determine optimal positions. Regarding the methodology for computing the positioning and communication KPIs, it could also be beneficial to investigate the NLoS as well, by transforming the corresponding parameters to analyze their effect on the computed PEB. This could provide a more robust way of examining the lower positioning bounds. In addition, it could be of interest to consider other types of delays contributing to overall latency, in order to identify which ones are most dominant and most sensitive to changes in configuration parameters.

6
Usage of AI

AI has been used to help with coding parts of the project. This includes coding in Python, Blender, and MATLAB. ChatGPT has been used to aid in the work process with non-crucial steps and assist in solving encountered errors [28]. Examples of such steps are specific functions for the MATLAB code for concatenating two matrices and creating the code used for generating the grid points in Blender.

AI has also been used to proofread and rephrase parts of the report for easier reading comprehension. ChatGPT was used for proofreading and rephrasing of the text to improve grammar and clarity [28]. Additionally, Scopus AI was used to find scientific publications that were included in the report as sources [29]. No AI was used to generate direct text or new ideas, only to improve what was already given or make suggestions for improvements.

Bibliography

[1] S. Diachenko, *From 1G to 5G: Evolution of Telecommunication Networks*, 2025. [Online]. Available: https://decisiontele.com/news/1g-5g-evolution-telecommunication-networks.html.

[2] B. Becher, *6G: What it is, how it works, when it will launch*, M. Urwin, Ed., 2025. [Online]. Available: https://builtin.com/hardware/6g.

[3] V. Ventures, *Wireless Communication: Definition, Explanation, and Use Cases*, 2024. [Online]. Available: https://www.vationventures.com/glossary/wireless-communication-definition-explanation-and-use-cases.

[4] muRata: Innovator in Electronics, *Basic Knowledge of Wireless Communication: Wireless Mechanism (1)*, 2023. [Online]. Available: https://article.murata.com/en-sg/article/basics-of-wireless-communication-1.

[5] C. Itoh, *A History of Wireless Communication and Yokogawa's Approach*, 2013. [Online]. Available: https://www.yokogawa.com/eu/library/resources/yokogawa-technical-reports/a-history-of-wireless-communication-and-yokogawas-approach/.

[6] S. Bartoletti, H. Wymeersch, T. Mach, O. Brunnegård, D. Giustiniano, and P. Hammarberg, "Positioning and Sensing for Vehicular Safety Applications in 5G and Beyond," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 15–21, 2021. DOI: 10.1109/MCOM.011.2100339.

[7] I. E. B. Purnomowati, G. Asmungi, and A. Y. Wirawan, "Pathloss calculation and analysis using different carrier frequency on wideband code division multiple access technology," *Advanced Science Letters*, vol. 21, no. 10, pp. 3185–3188, 2015, Cited by: 0. DOI: 10.1166/asl.2015.6436. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84960399307&doi=10.1166%2fasl.2015.6436&partnerID=40&md5=61329256ca652905e3162daa562bfab4.

[8] 3GPP, *6G Scenarios and Performance Requirements*, 2024. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/ran-6g-study1.

[9] Z. Cui, P. Zhang, and S. Pollin, "6G Wireless Communications in 7-24 GHz Band: Opportunities, Techniques, and Challenges," Tech. Rep., 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:263830182.

[10] A. Behravan, V. Yajnanarayana, M. F. Keskin, H. Chen, D. Shrestha, and T. E. Abrudan, "Positioning and Sensing in 6G: Gaps, Challenges, and Opportunities," *IEEE Vehicular Technology Magazine*, vol. 18, no. 1, pp. 40–48, 2023. DOI: 10.1109/MVT.2022.3219999.

[11] Blender Foundation, *Features — Blender*, Accessed: 2025-04-30, 2024. [Online]. Available: https://www.blender.org/features/.

[12] OpenStreetMap Contributors, *About OpenStreetMap*, Accessed: 2025-05-11, OpenStreetMap Foundation, 2023. [Online]. Available: https://wiki.openstreetmap.org/wiki/About_OpenStreetMap.

[13] W. Jakob, S. Speierer, N. Roussel, *et al.*, *Mitsuba 3 renderer*, version 3.0.1, 2022. [Online]. Available: https://mitsuba-renderer.org.

[14] J. Hoydis, S. Cammerer, F. Ait Aoudia, *et al.*, *Sionna*, version 1.0.2, 2022. [Online]. Available: https://nvlabs.github.io/sionna/.

[15] R. Enami, S. Gupta, D. Rajan, and J. Camp, "Laik: Location-specific analysis to infer key performance indicators," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4406–4418, 2021, Cited by: 4; All Open Access, Bronze Open Access. DOI: 10.1109/TVT.2021.3067880. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85103253900&doi=10.1109%2fTVT.2021.3067880&partnerID=40&md5=6a8c8429bda8965238a618109586e23b.

[16] S. B. Iqbal, B. Khodapanah, P. Schulz, and G. P. Fettweis, "On the application of reliability theory to cellular network mobility performance analysis," *IEEE Wireless Communications Letters*, vol. 13, no. 11, pp. 3104–3108, 2024, Cited by: 0; All Open Access, Green Open Access. DOI: 10.1109/LWC.2024.3452559. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85202769790&doi=10.1109%2fLWC.2024.3452559&partnerID=40&md5=987173793508aee4c1adadebafc35025.

[17] A. Kizilkaya, "Computation of the exact cramer-rao lower bound for the parameters of a nonsymmetric half-plane 2-d arma model," *Digital Signal Processing: A Review Journal*, vol. 18, no. 5, pp. 835–843, 2008, Cited by: 2. DOI: 10.1016/j.dsp.2008.05.002. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-47049094171&doi=10.1016%2fj.dsp.2008.05.002&partnerID=40&md5=7a2bbbfac1f637bdf6b44395df878741.

[18] Z. Lin, Q. Zou, and R. J. Ober, "The crlb for bilinear systems and its biomedical applications," Cited by: 2, 2005, pp. 1340–1343. DOI: 10.1109/ISCAS.2005.1464843. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-67649088655&doi=10.1109%2fISCAS.2005.1464843&partnerID=40&md5=bf19e2f8166aa0e974203256783d9332.

[19] Q. Zou, Z. Lin, and R. J. Ober, "The cramer-rao lower bound for bilinear systems," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1666–1680, 2006, Cited by: 11. DOI: 10.1109/TSP.2005.863006. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-33646063509&doi=10.1109%2fTSP.2005.863006&partnerID=40&md5=9882c0162f92986ef32e27935dafec81.

[20] C. L. Matson, M. Flanagan, and R. A. Vincent, "The impact of low signal-to-noise ratio values on the achievability of cramér-rao lower bounds with multi-frame blind deconvolution algorithms," Cited by: 1, vol. 7828, 2010. DOI: 10.1117/12.864330. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-78649828246&doi=10.1117%2f12.864330&partnerID=40&md5=befa3af9eb5690eab1309d3a2c854a2e.

[21] D. B. Jourdan, D. Dardari, and M. Z. Win, "Position error bound for uwb localization in dense cluttered environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 613–628, 2008. DOI: 10.1109/TAES.2008.4560210.

[22] Y.-F. Li and C. Jia, "A set of system reliability metrics for mobile telecommunication network," Cited by: 0, 2021, pp. 2933–2938. DOI: 10.3850/978-981-18-2016-8_765-cd. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85135480277&doi=10.3850%2f978-981-18-2016-8_765-cd&partnerID=40&md5=562d9cd59dd03475ecaaf257382df957.

[23] R. Enami, D. Rajan, and J. Camp, "Raik: Regional analysis with geodata and crowdsourcing to infer key performance indicators," Cited by: 16, vol. 2018-April, 2018, pp. 1–6. DOI: 10.1109/WCNC.2018.8377405. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049210925&doi=10.1109%2fWCNC.2018.8377405&partnerID=40&md5=7edbe36e0078257c1ce27cd373a643b0.

[24] H. B. Celebi, A. Pitarokoilis, and M. Skoglund, "Low-latency communication with computational complexity constraints," Cited by: 7; All Open Access, Green Open Access, vol. 2019-August, 2019, pp. 384–388. DOI: 10.1109/ISWCS.2019.8877142. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074659519&doi=10.1109%2fISWCS.2019.8877142&partnerID=40&md5=d2e1657428309004857aa593f15c6a21.

[25] S. Datta, *How to Calculate Packet Time from Latency and Bandwidth*, 2024. [Online]. Available: https://www.baeldung.com/cs/packet-time-latency-bandwidth.

[26] J.-P. Linnartz, P. Tuyls, and B. Skoric, *A communication-theoretical view on secret extraction*. 2007, pp. 57–77, Cited by: 3. DOI: 10.1007/978-1-84628-984-2_4. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84892214680&doi=10.1007%2f978-1-84628-984-2_4&partnerID=40&md5=c52955dca53e490dc4abc3384b71904b.

[27] K. Dugan, M. Harb, and D. Rice, "A reinforcement learning framework for optimizing throughput in docsis networks," Cited by: 0, 2021, pp. 50–55. DOI: 10.1145/3472735.3473389. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85113821511&doi=10.1145%2f3472735.3473389&partnerID=40&md5=d5dd7a43d412db7e002e36aae17891ee.

[28] OpenAI, *ChatGPT*, version ChatGPT-4, ChatGPT-4o, ChatGPT-4o mini, ChatGPT-4.1 mini. [Online]. Available: https://openai.com/chatgpt/overview/.

[29] Elsevier, *Scopus AI: Trusted content. Powered by responsible AI.* [Online]. Available: https://www.elsevier.com/products/scopus/scopus-ai.

A

Appendix

Github repository for the project:
`https://github.com/BaiLiping/Undergraduate_Thesis_Project.git`

A.1 Channel parameter derivatives

The general partial derivatives of the noise-free received signal with resepct to each unknown parameter listed in $\boldsymbol{\eta}_1$ in 3.2.2.

$$\frac{\partial \tilde{y}_{k,l}}{\partial \alpha_{\Re}} = e^{-j2\pi k\Delta_f \tau}\, \mathbf{a}_{\mathrm{BS}}^{\top} \boldsymbol{f}_l s_{k,l}\,. \tag{A.1}$$

$$\frac{\partial \tilde{y}_{k,l}}{\partial \alpha_{\Im}} = j e^{-j2\pi k\Delta_f \tau}\, \mathbf{a}_{\mathrm{BS}}^{\top} \boldsymbol{f}_l s_{k,l}\,. \tag{A.2}$$

$$\frac{\partial \tilde{y}_{k,l}}{\partial \tau} = -j2\pi k\Delta_f \alpha\, e^{-j2\pi k\Delta_f \tau}\, \mathbf{a}_{\mathrm{BS}}^{\top} \boldsymbol{f}_l s_{k,l}\,. \tag{A.3}$$

$$\frac{\partial \tilde{y}_{k,l}}{\partial \theta'_{\mathrm{BS}}} = \alpha\, e^{-j2\pi k\Delta_f \tau}\, \frac{-j2\pi f_c \boldsymbol{p}_n^{\top}}{c}\, \mathbf{a}_{\mathrm{BS}}^{\top} \boldsymbol{u}'_{\theta}\, \boldsymbol{f}_l s_{k,l}\,. \tag{A.4}$$

$$\frac{\partial \tilde{y}_{k,l}}{\partial \varphi'_{\mathrm{BS}}} = \alpha\, e^{-j2\pi k\Delta_f \tau}\, \frac{-j2\pi f_c \boldsymbol{p}_n^{\top}}{c}\, \mathbf{a}_{\mathrm{BS}}^{\top} \boldsymbol{u}'_{\varphi}\, \boldsymbol{f}_l s_{k,l}\,. \tag{A.5}$$

Where the $\boldsymbol{u}'_{\theta}$ and the $\boldsymbol{u}'_{\varphi}$ are defined as

$$\boldsymbol{u}'_{\theta} = \frac{\partial \boldsymbol{u}}{\partial \theta'_{\mathrm{BS}}} = \begin{bmatrix} -\sin(\theta'_{\mathrm{BS}})\sin(\varphi'_{\mathrm{BS}}) \\ \cos(\theta'_{\mathrm{BS}})\sin(\varphi'_{\mathrm{BS}}) \\ 0 \end{bmatrix}. \tag{A.6}$$

$$\boldsymbol{u}'_{\varphi} = \frac{\partial \boldsymbol{u}}{\partial \varphi'_{\mathrm{BS}}} = \begin{bmatrix} \cos(\theta'_{\mathrm{BS}})\cos(\varphi'_{\mathrm{BS}}) \\ \sin(\theta'_{\mathrm{BS}})\cos(\varphi'_{\mathrm{BS}}) \\ -\sin(\varphi'_{\mathrm{BS}}) \end{bmatrix}. \tag{A.7}$$

A.2 Jacobian derivatives 3.11

The general partial derivatives of the channel parameters listed in $\boldsymbol{\eta}_1$ with respect to the transformed parameters in $\boldsymbol{\eta}_2$.

$$\frac{\partial \alpha_{\Re}}{\partial \alpha_{\Re}} = 1\,. \tag{A.8}$$

$$\frac{\partial \alpha_{\Re}}{\partial \alpha_{\Im}} = 0\,. \tag{A.9}$$

$$\frac{\partial \alpha_{\Re}}{\partial x} = 0\,. \tag{A.10}$$

$$\frac{\partial \alpha_{\Re}}{\partial y} = 0\,. \tag{A.11}$$

$$\frac{\partial \alpha_{\Re}}{\partial z} = 0\,. \tag{A.12}$$

$$\frac{\partial \alpha_{\Im}}{\partial \alpha_{\Re}} = 0\,. \tag{A.13}$$

$$\frac{\partial \alpha_{\Im}}{\partial \alpha_{\Im}} = 1\,. \tag{A.14}$$

$$\frac{\partial \alpha_{\Im}}{\partial x} = 0\,. \tag{A.15}$$

$$\frac{\partial \alpha_{\Im}}{\partial y} = 0\,. \tag{A.16}$$

$$\frac{\partial \alpha_{\Im}}{\partial z} = 0\,. \tag{A.17}$$

$$\frac{\partial \tau}{\partial \alpha_{\Re}} = 0\,. \tag{A.18}$$

$$\frac{\partial \tau}{\partial \alpha_{\Im}} = 0\,. \tag{A.19}$$

$$\frac{\partial \tau}{\partial x} = \frac{\left[\Delta \boldsymbol{p}_{\mathrm{UE,BS}}\right]_1}{cd}\,. \tag{A.20}$$

$$\frac{\partial \tau}{\partial y} = \frac{\left[\Delta \boldsymbol{p}_{\mathrm{UE,BS}}\right]_2}{cd}\,. \tag{A.21}$$

$$\frac{\partial \tau}{\partial z} = \frac{\left[\Delta \boldsymbol{p}_{\mathrm{UE,BS}}\right]_3}{cd}\,. \tag{A.22}$$

$$\frac{\partial \theta'_{\mathrm{BS}}}{\partial \alpha_{\Re}} = 0\,. \tag{A.23}$$

$$\frac{\partial \theta'_{\mathrm{BS}}}{\partial \alpha_{\Im}} = 0\,. \tag{A.24}$$

$$\frac{\partial \theta'_{\mathrm{BS}}}{\partial x} = \frac{1}{1+\left(\frac{\boldsymbol{R}_{[1:3,2]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}\right)^2} \frac{\boldsymbol{R}_{[1,2]}^{\top}\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\boldsymbol{R}} - \boldsymbol{R}_{[1:3,2]}^{\top}\boldsymbol{R}_{[1,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{\left(\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}\right)^2}\,. \tag{A.25}$$

$$\frac{\partial \theta'_{\mathrm{BS}}}{\partial y} = \frac{1}{1+\left(\frac{\boldsymbol{R}_{[1:3,2]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}\right)^2} \frac{\boldsymbol{R}_{[2,2]}^{\top}\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}} - \boldsymbol{R}_{[1:3,2]}^{\top}\boldsymbol{R}_{[2,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{\left(\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}\right)^2}\,. \tag{A.26}$$

$$\frac{\partial \theta'_{\mathrm{BS}}}{\partial z} = \frac{1}{1+\left(\frac{\boldsymbol{R}_{[1:3,2]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}\right)^2} \frac{\boldsymbol{R}_{[3,2]}^{\top}\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}} - \boldsymbol{R}_{[1:3,2]}^{\top}\boldsymbol{R}_{[3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{\left(\boldsymbol{R}_{[1:3,1]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}\right)^2}\,. \tag{A.27}$$

$$\frac{\partial \varphi'_{\mathrm{BS}}}{\partial \alpha_{\Re}} = 0\,. \tag{A.28}$$

$$\frac{\partial \varphi'_{\mathrm{BS}}}{\partial \alpha_{\Im}} = 0\,. \tag{A.29}$$

$$\frac{\partial \varphi'_{\mathrm{BS}}}{\partial x} = \frac{-1}{\sqrt{1-\left(\frac{\boldsymbol{R}_{[1:3,3]}^{\top}}{d}\right)^2}} \frac{\boldsymbol{R}_{[1,3]}^{\top} d\Delta\boldsymbol{p}_{\mathrm{UE,BS}} - \frac{\Delta x_{\mathrm{UE,BS}}}{cd}\boldsymbol{R}_{[1:3,3]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{d^2}\,. \tag{A.30}$$

$$\frac{\partial \varphi'_{\mathrm{BS}}}{\partial y} = \frac{-1}{\sqrt{1-\left(\frac{\boldsymbol{R}_{[1:3,3]}^{\top}}{d}\right)^2}} \frac{\boldsymbol{R}_{[2,3]}^{\top} d\Delta\boldsymbol{p}_{\mathrm{UE,BS}} - \frac{\Delta y_{\mathrm{UE,BS}}}{cd}\boldsymbol{R}_{[1:3,3]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{d^2}\,. \tag{A.31}$$

$$\frac{\partial \varphi'_{\mathrm{BS}}}{\partial z} = \frac{-1}{\sqrt{1-\left(\frac{\boldsymbol{R}_{[1:3,3]}^{\top}}{d}\right)^2}} \frac{\boldsymbol{R}_{[3,3]}^{\top} d\Delta\boldsymbol{p}_{\mathrm{UE,BS}} - \frac{\Delta z_{\mathrm{UE,BS}}}{cd}\boldsymbol{R}_{[1:3,3]}^{\top}\Delta\boldsymbol{p}_{\mathrm{UE,BS}}}{d^2}\,. \tag{A.32}$$

CHALMERS
UNIVERSITY OF TECHNOLOGY