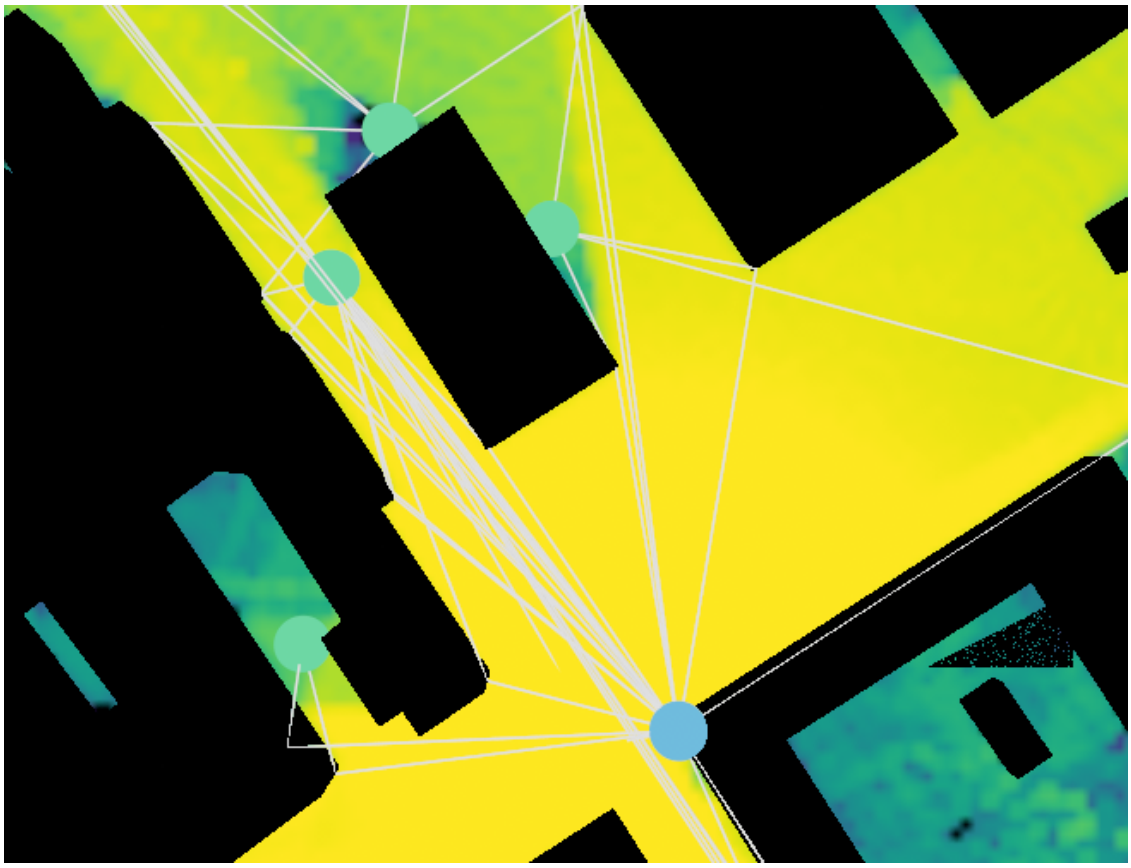

Map Inaccuracies Of Digital Twins For Localization

Evaluating Map Inaccuracies in Digital Twins via Ray Tracing

Master's thesis in Master Program Information and Communication Technology

NG LEAN TONG

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

Master's thesis 2025

Map Inaccuracies Of Digital Twins For Localization

Evaluating Map Inaccuracies in Digital Twins via Ray Tracing

LEAN TONG NG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Communication, Antennas, and Optical Networks
Chalmers University of Technology
Gothenburg, Sweden 2025

Map Inaccuracies Of Digital Twins For Localization
Evaluating Map Inaccuracies in Digital Twins via Ray Tracing
LEAN TONG NG

© LEAN TONG NG, 2025.

Supervisor(Arranged Alphabetically by First Initial):
Gerhard Steinböck, Ericsson
Martin Johansson N, Ericsson

Examiner:
Henk Wymeersch, Department of Electrical Engineering, Chalmers

Master's Thesis 2025
Department of Electrical Engineering
Division of Communication, Antennas, and Optical Networks
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Utilizing digital twin for localization via ray tracing simulation with building facade randomized mimicking inaccurate map.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Map Inaccuracies Of Digital Twins For Sensing and Localization
Evaluating Map Inaccuracies in Digital Twins via Ray Tracing
LEAN TONG NG
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Digital twins are increasingly vital in wireless communication for simulating, analyzing, and optimizing real-world environments, particularly for sensing and localization applications. The fidelity of these digital representations is dependent on the quality of the underlying maps, which, in practical industrial settings, are often procured at significant cost and may exhibit a range of inaccuracies due to survey limitations, temporal changes, and misalignments of the data. This thesis investigates the impact of map inaccuracies, specifically building rotations and translations, on localization accuracy and received signal strength Indicator(RSSI) distributions within a digital twin framework.

A comprehensive methodology was developed that combined ray tracing (using Sionna), mesh manipulation (using PyVista), and Monte Carlo simulation. The process included generating randomized maps according to realistic specifications, simulating electromagnetic propagation on both baseline and perturbed maps, and systematically extracting key channel and localization metrics. Several maximum likelihood-based (ML) positioning algorithms, including Vanilla MLE, Weighted MLE, Gain-Weighted Nonlinear Least Squares and Newton-Raphson ML, were implemented and benchmarked using synthetic multipath data generated via digital twin simulations.

Experimental results demonstrate that map inaccuracies can introduce significant deviations in localization, with position errors increasing as the degree of randomization increases. Although the cumulative distribution functions of the coverage map (CDFs) for path gain remain relatively robust, the evaluation shows a clear degradation in positioning accuracy with lower map fidelity. In particular, all ML-based algorithms significantly outperform baseline approaches, providing marked improvements in robustness and estimation accuracy under realistic conditions. The findings confirm that careful algorithmic selection and robust handling of ray-traced data can partially mitigate the negative effects of map imperfections.

This thesis provides actionable insights for the procurement, specification, and maintenance of digital twin maps in industrial localization deployments, and highlights the necessity of integrating advanced ML-based localization algorithms to maximize reliability and operational value.

Keywords: ray tracing simulation, localization estimation, map inaccuracy, digital twin, Monte Carlo simulation, maximum likelihood estimation, received signal strength Indicator(RSSI), mesh perturbation

Acknowledgements

First and foremost, I wish to extend my sincere gratitude to my supervisors at Ericsson Research, Gerhard Steinböck and Martin Johansson N. Your expert guidance and unwavering support have been instrumental throughout this thesis journey. The regular meetings and insightful discussions were not only a source of inspiration, but also provided invaluable direction whenever I faced uncertainty. I greatly appreciate the considerable time and effort you have dedicated to my development and research.

I thank Vinnova - Sweden's Innovation Agency for supporting my thesis research. This work has been carried out in part with the support of Vinnova, Sweden's Innovation Agency.
I am also grateful to my line manager, Sam Agneessens, for entrusting me with this opportunity and for your continuous readiness to assist whenever I encountered challenges. Your encouragement and availability have greatly contributed to the smooth progress of my work.

Special thanks go to Yen Chiou, Antonio, and Renèe for making my stay in Gothenburg comfortable and enjoyable. I truly appreciate the moments we shared, from providing a welcoming home environment to our time spent together on the playground and in countless playful pursuits.

Finally, and most importantly, I want to express my heartfelt appreciation to my family. Your steadfast support, love, and encouragement have been my foundation and strength, not only during this thesis but throughout my entire life.

Lean Tong Ng, Gothenburg, June 2025

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
BER	Bit Error Rate
DT	Digital Twin
EM	Electromagnetic
GTD	Geometrical Theory of Diffraction
ISAC	Integrated Sensing and Communications
LLS	Linear Least Squares
LOS	Line of Sight
MLE	Maximum Likelihood Estimation
ML	Maximum Likelihood
MIMO	Multiple Input Multiple Output
NLoS	Non-Line-of-Sight
NLS	Nonlinear Least Squares
RSSI	Received Signal Strength Indicator
RT	Ray Tracing
RX	Receiver
SBR	Shooting and Bouncing Ray
TDOA	Time Difference of Arrival
TOA	Time of Arrival
TX	Transmitter
UTD	Uniform Theory of Diffraction

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i, j	Indices for map grid cells along y- and x-axes, respectively
k	Index for rays or paths
N	Total number of samples, rays, or observations
t	Index for iterations (e.g., in optimization)

Sets

$C_{i,j}$	Area (set) of the cell in the coverage map at position (i, j)
$\mathcal{B}$	Set of all bounce points (last-bounce positions of rays)

Parameters

P_t	Transmitted power (W)
P_r	Received power (W)
G_t, G_r	Antenna gains (dimensionless)
λ	Wavelength (m)
$\lvert h(s)\rvert^2$	Squared amplitude of path coefficients
d	Distance between transmitter and receiver antennas (m)
$\lvert C\rvert$	Area of a coverage map cell (m^2)
`cm_cell_size`	Coverage map cell resolution parameter (m)
ϵ	Small positive value for numerical stability
T	Maximum number of iterations (e.g., in Newton–Raphson)

σ_i^2	Variance of measurement i (Gaussian noise variance)

Variables

$g_{i,j}$	Average path gain in coverage map cell (i, j)
$h(s)$	Complex channel/path coefficient at location s
x, y	Spatial position variables (coordinates to be estimated)
$\hat{x}, \hat{y}$	Estimated coordinates of the target/device
$f_i(x, y)$	Theoretical measurement model for anchor i
z_i	Observed measurement for anchor i (e.g., TOA, TDOA, RSS)
$\mathbf{z}$	Vector of observed measurements
$F(x)$	Empirical cumulative distribution function (CDF) at value x
$\mathbf{B}[i]$	Last-bounce 3D coordinates for ray i
$r[i]$	Distance from last bounce to endpoint for ray i (sphere radii)
$\mathbf{x}$	Estimated receiver position (vector), $\mathbf{x} = [x, y, z]^T$
$e[i]$	Residual for measurement or ray i in optimization
$w[i]$	Path gain-based weight for ray i
$w_{\text{norm}}[i]$	Normalized weight for ray i
$\hat{\mathbf{x}}$	Estimated position (solution of localization algorithm)
$\mathbb{I}(\cdot)$	Indicator function (1 if condition is true, else 0)

Special Functions

$\log_{10}$	Logarithm to base 10
$\|\cdot\|$	Euclidean norm of a vector
$\text{diag}(\cdot)$	Diagonal matrix with entries specified by the argument
$\arg\max$, $\arg\min$	Argument of the maximum/minimum value of a function

Contents

List of Figures

List of Tables

1
Introduction

This thesis explores the intersection of digital twin(DT) technology and radio-based localization. A particular emphasis is on how inaccuracies in virtual map representations affect localization performance. As DT become increasingly integral to the design and analysis of real-world systems, ensuring the accuracy and reliability of their spatial models is critical, especially in applications involving electromagnetic (EM) simulations and radio-based localization. By employing ray tracing techniques within simulated environments, this work investigates how controlled map distortions influence localization outcomes, offering new insights into the fidelity requirements of DT in wireless localization contexts.

1.1 Background

The concept of DT has increasingly gained importance across various industries, providing powerful tools for simulating, analyzing, and optimizing physical environments virtually. A DT serves as a virtual counterpart in real time of a physical object or environment, enabling detailed analysis, predictive maintenance, and efficient decision-making processes. The rise of DT can be attributed to advancements in computational resources, sensor technology, and machine learning techniques, which together have enabled more accurate and dynamic representations of complex systems. In particular in the sensing and localization domains, DT offer the potential to significantly streamline data collection processes, improve system accuracy, and facilitate advanced planning and operational strategies.

Accurate localization plays a vital role in various applications, such as navigation systems, autonomous vehicles, urban planning, and telecommunications. Achieving high accuracy requires high-fidelity maps. These maps are the foundational data in DT frameworks. Ray tracing (RT), a deterministic modeling technique for radio wave propagation, is frequently utilized within DT to replicate environments electromagnetically and subsequently predict signal behaviors such as reflections and losses, which are essential for localization and communication systems.

However, despite technological advancements, real-world maps often suffer from inherent inaccuracies due to data acquisition constraints, limited surveying precision, temporal changes, and misalignment. These map inaccuracies, such as misplaced buildings, inaccuracies in the EM representation of materials, or missing environmental details, can significantly affect the reliability and performance of DT appli-

cations, particularly those dependent on RT-based simulations.

1.2 Aim

This thesis aims to investigate and quantify how specific spatial map inaccuracies impact localization accuracy and received signal strength indicator (RSSI) distribution within a DT environment. By leveraging RT simulations, this research intends to systematically examine discrepancies that arise between a high-fidelity baseline map and intentionally modified versions that mimic real-world inaccuracies. The central objective is to develop practical methodologies and metrics that facilitate the assessment and mitigation of these map inaccuracies, providing clear guidelines to maintain the quality of DT in localization applications.

1.3 Objective

The objective of this thesis is to bridge the gap between theoretical DT capabilities and real-world limitations by evaluating how map inaccuracies, arising from practical survey and maintenance constraints, affect the localization performance of wireless systems. The insights gained will inform procurement strategies and map quality requirements for stakeholders in the communication industry seeking to take advantage of DT for advanced network utilization.

1.4 Scope and Limitations

The scope of this master's thesis is to analyze how spatial inaccuracies in DT maps affect localization accuracy and RSSI distributions in wireless network simulations. In particular, the study focuses on map inaccuracies arising from geometric transformations, such as building rotations and translations. The workflow is centered on RT techniques, utilizing the Sionna toolbox (0.19.2) [8] for deterministic multipath propagation modeling, PyVista [21] for flexible 3D mesh visualization and manipulation, and Python for simulation orchestration and data analysis.

Sionna RT is a GPU-accelerated platform originally designed to model realistic radio propagation, including mechanisms such as reflection, scattering, and diffraction. Its core functionality is optimized to determine all possible propagation paths between specific transmitter (Tx) and receiver (Rx) locations within a 3D scene. However, for this thesis, the experimental design requires shooting rays from a source in specified directions and following them until a maximum range is reached, essentially recording each last interaction point as a possible candidate for localization. Sionna's current API does not natively support this use case, particularly the flexible "shoot-and-follow" paradigm needed for custom positioning algorithms.

To address this limitation, a more generic mesh handling library, PyVista, is used.

PyVista enables an efficient computation of intersections between arbitrary rays and 3D meshes. Upon detecting a ray/mesh intersection, PyVista provides the point of intersection and the associated triangle ID, from which the surface normal can be extracted [21]. This allows for the implementation of specular reflection as well as, in principle, custom diffraction or scattering mechanisms (e.g., randomizing the outgoing direction at edges or triangles). However, due to time constraints and the absence of built-in support, advanced scattering and diffraction were not implemented within the PyVista workflow for this project. Only specular reflections from the last intersection point are considered when estimating positions, using a sphere with the remaining path length. This simplification is motivated by computational feasibility and allows for direct comparison with the default Sionna RT workflow.

Other limitations should be noted. First, PyVista does not natively handle the full range of propagation phenomena (such as higher-order diffraction or complex scattering), potentially restricting the completeness of the physical modeling. Second, mesh alteration and manipulation operations in Sionna are significantly slower compared to specialized libraries such as Trimesh. Therefore, Trimesh is used to preprocess meshes efficiently before loading them into PyVista or Sionna. Third, the computational demands of Sionna are high, especially when running RT simulations on large-scale environments (e.g., the Kista map), leading to extended runtimes. Finally, transferring ray path configurations or propagation settings between maps is limited by the design of Sionna, necessitating a combination of toolchains for the full simulation pipeline.

Hardware measurements and propagation modeling beyond the scope of ray tracing, including hardware-based data collection, and the development or integration of new GPU-accelerated toolboxes for custom ray propagation, are outside the project's practical scope due to technical and time constraints.

1.5 Related work

Recent years have seen a surge of research into DT technologies, with broad investigations spanning enabling technologies, common challenges, and emerging trends that highlight the relevance and applicability of DT across diverse domains [1]. Within this expansive landscape, more focused studies have advanced the understanding of EM propagation and localization, especially through the use of RT simulations for benchmarking and error quantification in DT environments [1, 24]. These computational benchmarking efforts are crucial for evaluating localization accuracy and for identifying sources of error, thereby providing the foundation for the methods explored in this thesis.

Previous studies have extensively applied RT methodologies to EM propagation modeling, establishing foundational knowledge of deterministic RT techniques and their practical applications in radio wave propagation prediction [5, 6]. A key recent development is the resurgence of interest in fingerprinting-based localization, a technique that determines the user position by matching observed signal mea-

surements with a database of pre-collected measurements linked to known locations [5, 6]. Fingerprinting-based localization has gained renewed interest through the use of DT generated radio frequency maps [22]. These maps, derived from high-fidelity ray tracing simulations on digital replicas of real-world environments, enable the creation of large-scale fingerprinting databases without the need for labor-intensive field measurements. This digital approach not only mitigates the overhead associated with traditional fingerprinting but also achieves sub-metre accuracy localization in non-line-of-sight (NLoS) conditions localization. By replacing physical data collection with simulation-based fingerprint generation, researchers have demonstrated the feasibility of accurate, scalable, and low-effort positioning systems suitable for future 6G wireless networks [22, 23].

1.6 Structure of the thesis

This thesis is organized into five chapters.

- Chapter 1: Provides the introduction, outlining the research background, aim, scope and limitations, and related work.
- Chapter 2: Covers theoretical concepts related to DT, map inaccuracies, ray tracing fundamentals, and localization principles.
- Chapter 3: Presents the methodology, including the processes of generating randomized maps, conducting ray tracing simulations, formulating the positioning algorithm, and creating coverage maps.
- Chapter 4: Presents the results and an analysis of the impact of map inaccuracies, evaluates the positioning algorithm, and offers detailed conclusions and recommendations.
- Chapter 5: Provides a comprehensive summary of the thesis contributions and outlines potential directions for future research.

2

Theory

This chapter presents the theoretical foundations underpinning the methods and analysis in this thesis. It begins by introducing the core concepts of DT and their significance for wireless communication research. The chapter then details the principles of EM propagation and ray tracing, including key mechanisms such as reflection, scattering, and diffraction, as well as commonly employed techniques such as the image method. The Sionna simulation framework is described in depth, highlighting its flexible scene management, path computation, and coverage analysis capabilities [8]. Practical challenges related to map inaccuracies are examined, followed by a comprehensive review of localization algorithms based on maximum likelihood estimation. Together, these theoretical elements establish the foundation for the simulation, analysis, and positioning experiments presented in the subsequent chapters.

2.1 Digital Twin Concepts

DT (DT) have rapidly evolved into a cornerstone technology for modern engineering, urban management, and wireless communication systems. In essence, DT are dynamic digital replicas of real-world assets, processes, or environments, continuously updated with data from physical systems through integrated sensor networks, computational models, and data analytics [1, 3]. Distinct from conventional simulation tools, DT are characterized by their ability to support real-time, bi-directional interaction: they both reflect the current state of their physical counterparts and can also influence and optimize real-world processes via feedback and control mechanisms. This capability is largely enabled by advances in IoT connectivity, machine learning, scalable cloud/edge computing, and the increasing availability of high-fidelity 3D modeling technologies [1, 2].

The growing influence of DT is tightly linked to the ambitions of Industry 4.0, which envisions deeply interconnected, intelligent, and autonomous systems [1]. Originally conceived as a tool for monitoring and managing the lifecycle of manufactured products, the DT has since expanded to encompass large-scale systems, connecting entire fleets of machines, urban infrastructure, or even national-scale utilities to their virtual analogues [2, 3]. This expansion is made possible through the continuous high-speed synchronization of sensor data and operational states, allowing DT to support predictive maintenance, anomaly detection, and adaptive control in response to rapidly changing environments [1, 3].

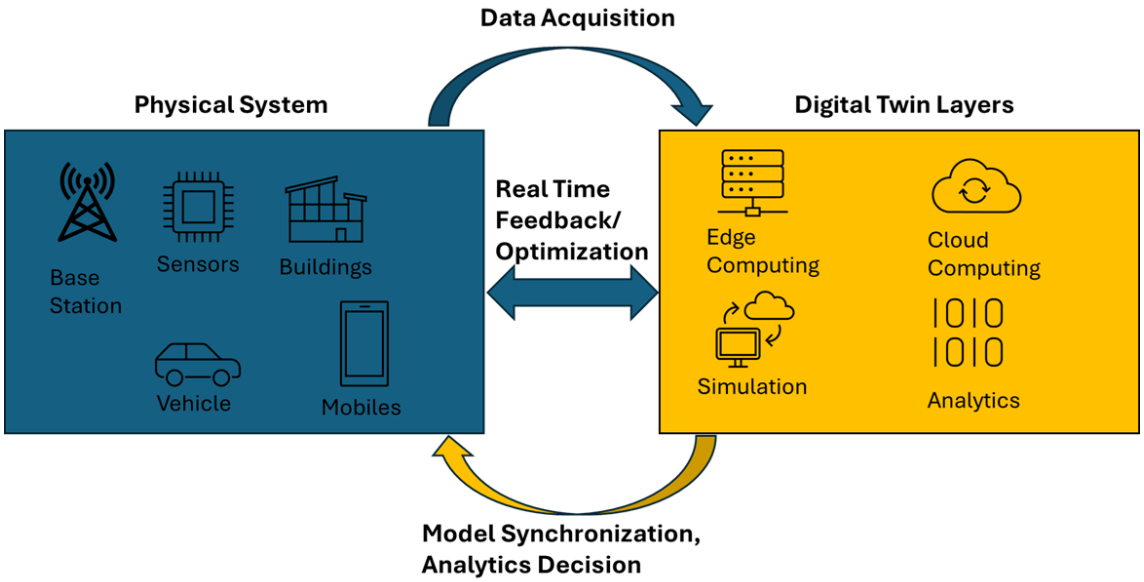

Figure 2.1: Conceptual Overview of Digital Twin Architecture

Within the domain of wireless communications, DT are now recognized as an enabling technology for the next generation of mobile networks, particularly 6G and the broader Internet of Everything (IoE) paradigm [2]. A conceptual overview of the DT architecture for communication systems is illustrated in Figure 2.1. Here, DTs provide the digital scaffolding for proactive network design, dynamic resource allocation, and performance optimization. For example, by creating a virtual copy of a wireless network, including its base stations, channels, user devices, and even the surrounding urban landscape, a DT can facilitate machine learning-driven network optimization, simulate the impact of infrastructure changes, and predict user experience under diverse conditions. DT architectures are generally categorized into three main types (monitoring, simulation, and operational), each serving distinct roles within a cyber-physical system. An operational DT not only encompasses monitoring and simulation capabilities, but also allows direct interaction with, and control of, the physical system itself. This enables system operators to execute actions, fine-tune configurations, and respond proactively to changing conditions, all in a virtualized, risk-free environment [2].

The scope of DT applications continues to broaden. In manufacturing, DTs are leveraged for advanced automation, quality assurance, and energy efficiency. In smart cities, they support initiatives for infrastructure monitoring, traffic optimization, and sustainability assessment [1, 3]. In the communications sector, DT enable virtual radio propagation modeling and channel estimation, allowing engineers to predict and optimize wireless coverage, interference patterns, and network capacity before physical deployment [2, 4]. As a recent example, city-scale DT simulations have been shown to significantly improve channel estimation accuracy in vehicular networks by capturing the influence of real urban environments and building geometries on signal propagation [4].

Despite the promise of DT's, several challenges remain on the path to widespread adoption. Among these challenges are the integration of heterogeneous data sources, the development of standardized and interoperable models, and the establishment of robust security and privacy protections [1, 2, 3]. Furthermore, achieving low-latency and high-fidelity synchronization between the physical and digital domains requires a sophisticated networking infrastructure and scalable computational resources. As DT's become embedded in high-reliability and safety-critical applications, such as autonomous mobility or future 6G communication networks, the importance of scalable, robust, and trustworthy architectures becomes even more critical.

DT provide a powerful bridge between the physical and virtual domains, enabling real-time analysis, predictive simulation, and automated optimization across a broad range of applications. Their relevance in communications is expected to expand further with the emergence of 6G and ubiquitous IoT, where modeling, prediction, and adaptation to complex and dynamic environments are essential to meet ever-stricter requirements for latency, reliability, and network efficiency [2].

2.2 Ray Tracing Fundamentals

RT is a physics-based and deterministic computational method extensively used to model EM wave propagation in complex environments such as urban areas, buildings, and other engineered spaces [5]. Unlike stochastic (statistical) models, which rely on measurement-based averages to represent signal variability, deterministic RT methods calculate the exact trajectories and interactions of individual rays according to the principles of geometrical optics. This provides spatial consistency and detailed predictions of signal strength, delay spread, angle of arrival (AOA), and other multipath characteristics. These predictions are critical for the design and optimization of wireless communication systems and DT's. Site-specific predictions are enabled by modeling reflection, scattering [19], and diffraction [17, 18] using path-finding algorithms such as the image method and Shooting and Bouncing Ray (SBR). Site-specific modeling is crucial for evaluating the impact of map inaccuracies in DT and for the design and deployment of next-generation wireless systems.

2.2.1 Propagation Mechanisms: Reflection, Scattering, and Diffraction

The propagation of radio waves in real environments is governed by three primary mechanisms when the ray is obstructed: reflection, scattering, and diffraction.

Reflection occurs when a ray encounters a surface such as a building facade or the ground plane. The ray changes direction according to the law of reflection, and its strength is determined by the Fresnel equations [5]. Reflections are the dominant propagation mechanism in environments characterized by many large, flat surfaces. The reflective properties depend strongly on the material; for example, concrete, glass, and brick have different permittivity and conductivities, as summarized in

the ITU-R P.2040 recommendation [16]. This standard provides comprehensive tables for the EM properties of common building materials, which are widely used in radio propagation modeling.

Scattering refers to the redirection of energy in multiple directions due to interaction with rough or irregular surfaces (e.g., textured building walls, foliage, or urban clutter). Scattering can be divided into specular (mirror-like) and diffuse (randomized) components. In urban environments, diffuse scattering plays a significant role in multipath propagation and is especially important at high frequencies (see Figure 2.3) [5, 19].

Diffraction occurs when a ray encounters sharp edges or corners, such as building rooftops or street canyons, bending around obstacles into shadow regions that are not directly visible from the Tx. The mathematical treatment of diffraction is more complex than for reflection and relies on the Uniform Theory of Diffraction (UTD) or Geometrical Theory of Diffraction (GTD) for practical calculations (see Figure 2.4) [5, 17, 18].

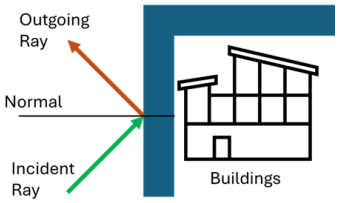

Figure 2.2: Illustration of ray reflection from a building surface.

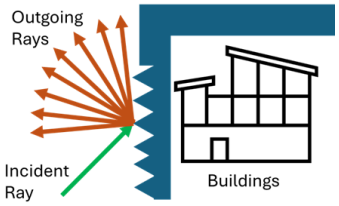

Figure 2.3: Scattering from rough surfaces causing multipath propagation.

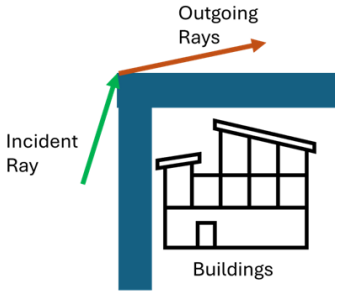

Figure 2.4: Diffraction around building edges allowing signal to reach shadowed regions.

2.2.2 Ray Tracing Algorithms: Image Method and SBR

Accurate modeling of EM wave propagation in complex environments requires specialized algorithms to compute the multitude of possible ray paths between Tx and Rx. Two principal approaches are widely used in ray tracing: the Image Method and the SBR method.

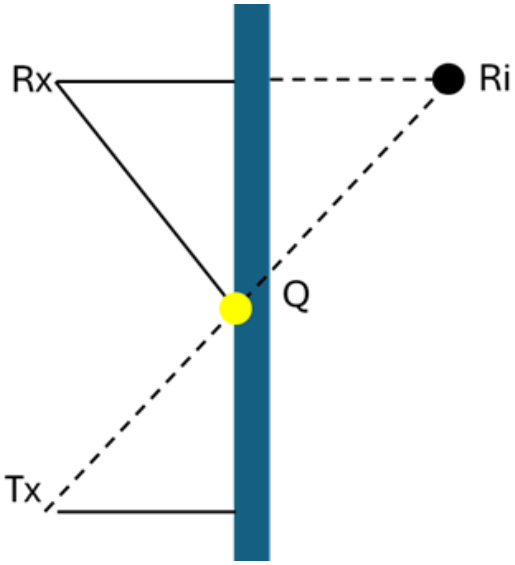

Figure 2.5: The image method for calculating reflection paths: the transmitter (Tx), image source (Ri), reflection point (Q), and receiver (Rx).

Image Method: The image method is a geometric ray tracing technique that leverages the concept of virtual images to efficiently determine reflection paths. When a ray is reflected from a planar surface (such as a wall or the ground), the process involves constructing the "image" of either Tx or Rx across the reflecting surface. For each reflection, the virtual image is found by mirroring the real point with respect to the plane. To find the trajectory of a single reflected ray, one can, for example, construct the image of the Rx (R_i), then draw a straight line from the Tx to this image. The intersection of this line with the reflecting plane gives the exact location where the reflection occurs. The path of the reflected ray is then defined as the sequence $(\mathrm{Tx}, Q, \mathrm{Rx})$, where Q is the reflection point (see Figure 2.5). This construction is mathematically equivalent if one instead images the Tx and connects the Rx to the image point. The image method is not only computationally efficient, since it reduces multi-bounce reflections to a series of geometric constructions, but is also highly accurate for environments with a small number of large, flat reflecting surfaces. It can be extended recursively for multiple sequential reflections by constructing a sequence of image points through successive mirroring. However, the method becomes less practical in environments with highly irregular geometries or numerous obstacles, where the number of possible image paths grows exponentially [5, 6].

Shooting and Bouncing Ray (SBR) Method: The SBR method, first introduced in the context of radar cross-section analysis, has become a cornerstone for

ray-based modeling in radio propagation, especially for urban and indoor environments with complex structures. The SBR algorithm is composed of three main steps: ray launching, ray tracing, and ray reception.

1. In the **ray launching** step, a large number of rays are generated from the Tx, distributed as uniformly as possible in all directions. This uniformity ensures that the energy radiated by an isotropic source is accurately modeled, often using spherical divisions such as icosahedral tiling for near-uniform coverage.
2. During **ray tracing**, each ray is independently propagated through the environment. The algorithm recursively checks for intersections with obstacles: if a ray hits a surface, it is either reflected or diffracted based on the geometry and material; if the ray continues unobstructed, it can reach the Rx directly. Reflection and diffraction events spawn new ray segments that are tracked in subsequent steps. In realistic scenarios, the majority of computational time is spent on detecting ray-surface intersections, which motivates the use of spatial acceleration structures like triangular grids or bounding volumes.
3. In the **reception** step, the algorithm determines whether a given ray or its associated "ray tube" illuminates the receiving point. Ray tubes may be modeled with circular or polygonal cross-sections: circular tubes are simpler but risk double-counting, while polygonal tubes are more accurate but computationally intensive.

The strength of the SBR method is its flexibility and scalability: It can easily handle arbitrary geometries, multiple interactions (reflection, diffraction, scattering) and large-scale environments. However, it may produce less precise ray trajectories compared to the image method, especially in scenarios with a limited number of dominant propagation paths. To address this, hybrid methods have been proposed. SBR is used to identify candidate rays that reach the Rx, and the image method is then used to refine the ray trajectories to improve accuracy with minimal additional computation [5].

2.3 Electromagnetic Path Propagation

The path propagation of EM waves in wireless communications is quantified by the path gain, which expresses the attenuation of signal power between a Tx and Rx due to free-space spreading and additional losses from interactions with the environment [5].
The Friis transmission equation models the ideal free-space case:

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2$$

where:

- P_r: Received power (W)
- P_t: Transmitted power (W)
- G_t, G_r: Antenna gains (dimensionless)
- λ: Wavelength (m)
- d: Distance between antennas (m)

Expressed in decibels (dB), the path gain (loss) is:

$$L_{dB} = 10 \log_{10} \frac{P_t}{P_r} = -10 \log_{10} \left(G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2 \right)$$

When obstacles introduce reflection, scattering, and diffraction, the received power is further reduced. Ray tracing numerically incorporates these losses, calculating the total path gain for each propagation path based on geometric and material parameters of the environment. This detailed modeling is foundational for DT-based wireless simulation and performance prediction.

2.4 Sionna Introduction

Sionna is an open-source, GPU-accelerated Python library designed to facilitate next-generation wireless communication research, with a special focus on physical layer modeling and simulation. Developed by NVIDIA, Sionna integrates a powerful and differentiable ray tracing engine (Sionna RT). This engine allows researchers to simulate, analyze, and optimize radio propagation in realistic 3D environments with acceptable flexibility and computational efficiency [9, 10, 11].

Built atop TensorFlow and leveraging the Mitsuba 3 renderer, Sionna RT not only supports classic ray tracing for EM propagation modeling, but also enables end-to-end learning and gradient-based optimization through differentiable programming. This is particularly valuable for emerging 6G research, reconfigurable intelligent surfaces, integrated sensing and communications (ISAC), and DT environments [11].

2.4.1 Ray Tracing and Scene Management

Sionna allows users to load, manipulate, and visualize detailed 3D scenes using open formats, with assets easily created in Blender or imported from global sources such as OpenStreetMap. The `scene.preview()` functions enable interactive 3D visualization and high-quality rendering of radio propagation scenarios, providing clear insights into RT behavior (see Fig.2.6) [9, 14].

2.4.2 Computing Propagation Paths: `compute_paths`

A core feature of Sionna RT is the `compute_paths` method, which determines all significant propagation paths between Tx and Rx within the modeled scene. By setting parameters such as `max_depth`, users can control the maximum order of reflections. The result is a comprehensive set of ray paths, each characterized by its geometric trajectory, number of interactions, and physical parameters like path loss, delay, angle of departure(AOD) and AOA. These outputs are critical for modeling realistic channel impulse responses and for generating synthetic datasets for localization or channel estimation research (see Fig.2.9) [9, 12].

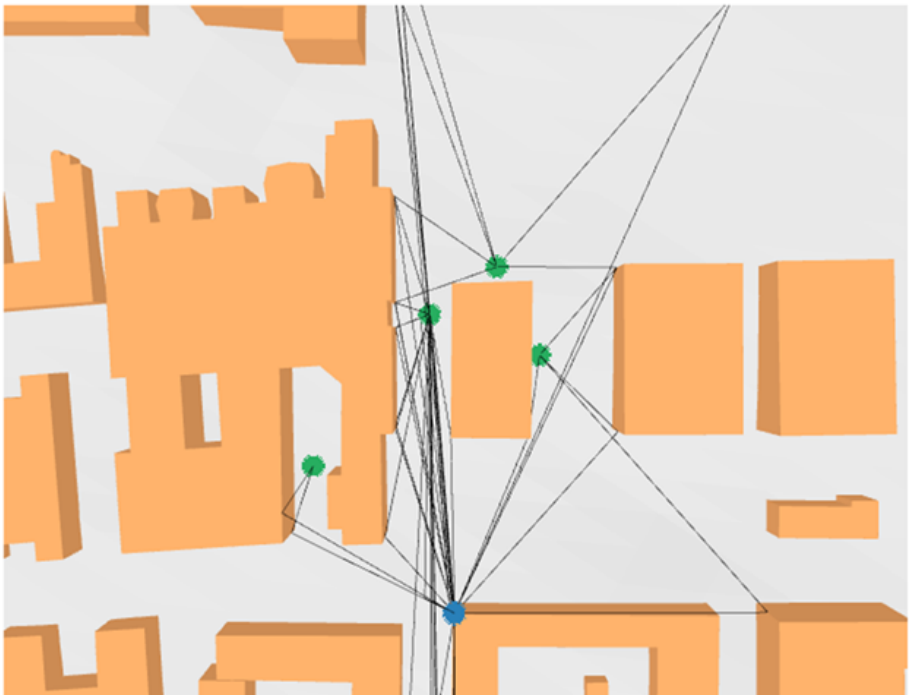

Figure 2.6: Visualization of computed ray tracing paths in a Sionna 3D scene.

2.4.3 Coverage Map and CDF Computation: `coverage_map`

Sionna's `coverage_map` function automates the computation of spatial distributions of path gain (RSSI) or other metrics across a defined area. Users can specify the map resolution, propagation parameters, and which phenomena (reflection, diffraction, scattering) to enable. The resulting coverage maps are essential for visualizing signal strength, identifying coverage holes, and benchmarking network performance. For a given Tx, a coverage map is a rectangular surface with arbitrary orientation, subdivided into rectangular cells of size $|C| = \texttt{cm\_cell\_size}[0] \times \texttt{cm\_cell\_size}[1]$. The user-defined parameter `cm_cell_size` controls the spatial granularity of the map.
The coverage map associates with every cell (i, j) the following quantity:

$$g_{i,j} = \frac{1}{|C|} \int_{C_{i,j}} |h(s)|^2 ds$$

where $|h(s)|^2$ is the squared amplitude of the path coefficients a_i at position $s = (x, y)$, and the integral is over the area $C_{i,j}$ of the cell. Here, $ds = dx \cdot dy$ is the infinitesimal surface element. The index i (respectively j) corresponds to the y-axis (respectively x-axis) of the coverage map in its local coordinate system. Thus, $g_{i,j}$ can be interpreted as the average path gain across the cell $C_{i,j}$. See Fig. 2.8 for a CDF plot example and Fig. 2.7 coverage map example [9, 13].
The computed path gain can be further transformed into the received signal strength (RSSI) by multiplying it with the Tx power, thereby providing a direct mapping to system-level metrics.

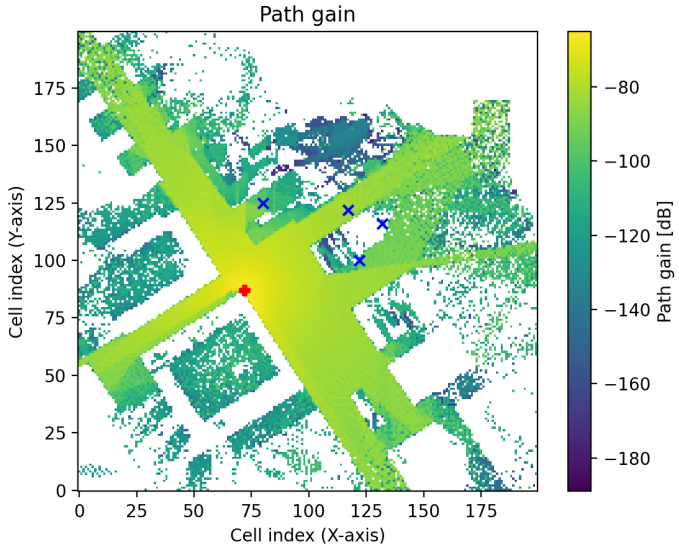

Figure 2.7: Radio Map propagation scenario generated with Path Gain using `Coverage_map`.

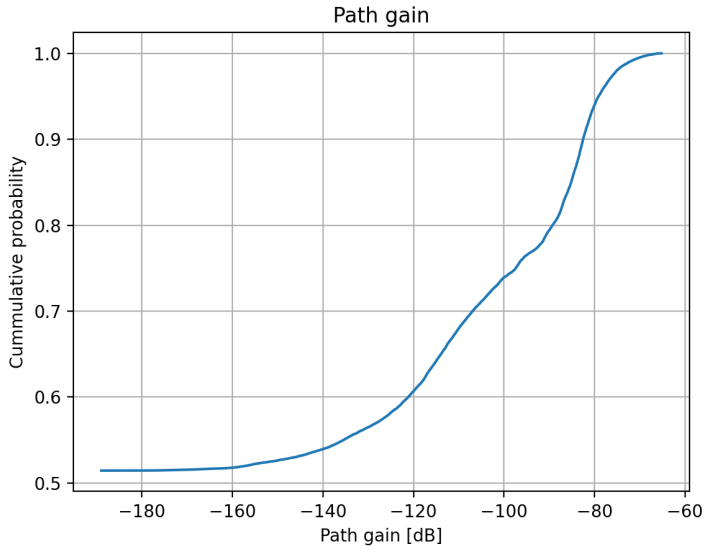

Figure 2.8: Empirical CDF of path gain across a simulated coverage map, illustrating signal reliability statistics.

Mathematical Formulation: Given a set of N path gain samples $\{x_1, x_2, \ldots, x_N\}$ obtained from the coverage map, the empirical cumulative distribution function (CDF) at a value x is defined as:

$$F(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(x_i \leq x)$$

where $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition inside is true, and 0 otherwise.
In the context of Sionna, the `CoverageMap.cdf()` method automates this process by computing the ECDF of the path gain values across the coverage map grid. This function returns the sorted path gain values and their corresponding cumulative probabilities, facilitating straightforward plotting and analysis of the CDF [13].

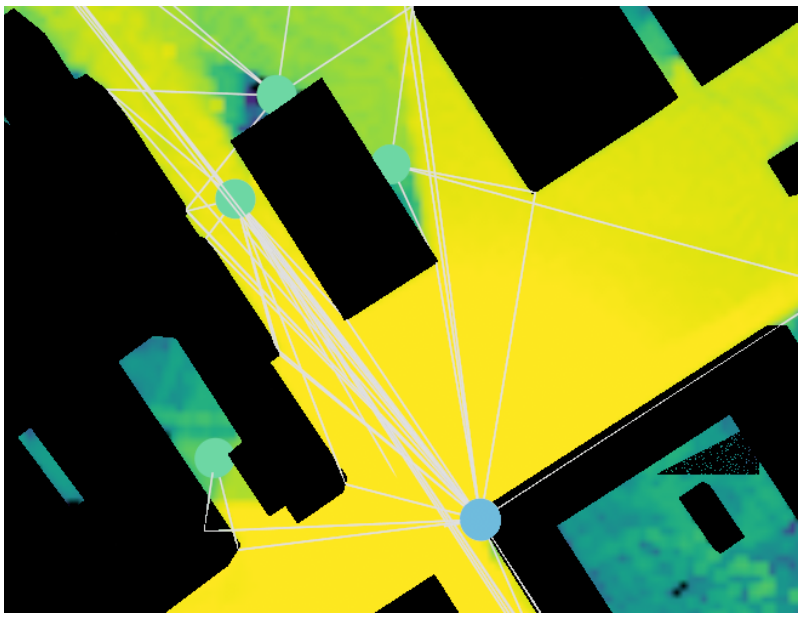

Figure 2.9: Rendered high-resolution image of a propagation scenario generated using `render_to_file`.

2.4.4 Applications and Visualization: `render_to_file`

In this thesis, the `render_to_file` function is employed to produce high-resolution visualizations of the 3D simulation scene from user-defined camera perspectives. These rendered images are generated along with the thesis to qualitatively assess propagation environments, highlight Tx and Rx locations, and visualize ray paths and coverage maps (see Fig. 2.9) [14]. In the visualizations, greenish dots represent Rx and blueish dots represent Tx positions, lines indicate simulated ray paths, and black shapes correspond to building geometries. The background color gradients reflect simulated signal coverage or path gain, although a colorbar is not included in the figure.

2.4.5 Sionna Flow: Compute Paths, Coverage Maps, and Rendering

Sionna's workflow is as follows:

1. Loading or creating a 3D scene (from Blender, OpenStreetMap, or built-in examples) [15].
2. Placing Tx and Rx at user-defined positions with configurable antenna arrays.

3. Using `scene.compute_paths()` to calculate all relevant radio propagation paths, capturing key details like reflection/diffraction order, path length, delay, and path gain.
4. Using `scene.coverage_map()` to evaluate and visualize the distribution of signal power or other metrics over a 2D grid.
5. Exporting results or rendering visualizations to high-resolution files with `scene.render_to_file()`.

This workflow enables rapid iteration on scene design and propagation modeling, making Sionna a valuable tool for both practical engineering and academic exploration [12, 13, 14].

2.5 Map Inaccuracies in Practice

Accurate spatial representation is fundamental to the integrity of DT applications, particularly for wireless sensing and localization tasks. In industry practice, communication companies frequently purchase maps from specialized vendors. The procurement of these maps incurs substantial cost. Pricing is strongly correlated with the resolution, accuracy and level of detail of the map. For example, a typical high-resolution commercial map may feature a 1-metre grid and a relative planimetric accuracy of ± 3 metres, while older or less detailed maps might offer only a 5-metre grid and relative planimetric accuracy of ± 5 metres. The choice of map directly impacts simulation fidelity and, consequently, the reliability of DT-driven localization and network planning.

Despite considerable investment, real-world mapping is inherently subject to inaccuracies stemming from several major sources:

- **Survey resolution:** The spatial resolution of a map, determined by the density of surveyed points or the granularity of remote sensing, limits the minimum distinguishable feature size. Higher-resolution maps capture finer details but are more expensive to obtain and process.
- **Alignment errors:** Misalignments can arise from sensor drift, georeferencing errors, or the merging of heterogeneous datasets. These lead to systematic shifts in building footprints or road networks within the DT.
- **Temporal changes:** Urban environments are dynamic. Construction, demolition, vegetation growth, and infrastructure upgrades introduce discrepancies between the map and the real world if the map is not frequently updated.

The fidelity of EM simulations for wireless communication systems is influenced by the geometric accuracy and material characterization of the underlying maps. While research suggests that local discrepancies, such as misplaced buildings, incorrect orientations, missing features, or material mismatches, can impact radio propagation models and potentially affect coverage prediction and localization, the extent of this impact at a system level is less clear [3, 24]. In particular, while individual locations may exhibit significant deviations, overall statistical properties such as path gain distributions may remain relatively robust. This thesis investigates these effects in detail, examining how map inaccuracies manifest in both localized and aggregate

performance metrics.

The computational cost of high-fidelity EM simulation is further magnified by map complexity and mesh resolution, as shown in recent benchmarking studies [24]. These studies highlight the need to balance model accuracy with practical considerations, such as simulation runtime and hardware requirements, especially in the context of real-time DT applications for 6G and beyond.

To systematically investigate the effects of map inaccuracies, this thesis introduces controlled perturbations to the base map (original map without randomization) in the form of vertex translations and rotations. These modifications serve as representative models for common real-world errors, such as those introduced by limited survey precision or the temporal evolution of the urban landscape. By deliberately simulating these typical inaccuracies within the DT framework, it becomes possible to quantify their impact on localization outcomes. Such controlled experiments are essential for providing actionable guidance on the trade-offs between map fidelity, procurement cost, and application-level performance in wireless network planning and DT-based localization.

2.6 Localization Principles

Wireless localization is a fundamental capability in contemporary communication networks, enabling essential services such as navigation, emergency response, and asset tracking [1, 27]. The process typically involves estimating the position of a mobile device using radio signal measurements taken from multiple anchor nodes (such as base stations or access points). The following subsections provide an overview of the main measurement techniques, statistical estimation methods, fingerprinting strategies, and practical challenges in localization.

2.6.1 Measurement Techniques

A range of physical-layer measurements are utilized in localization systems, each offering distinct advantages depending on the application and environment:

- Time of Arrival (TOA): Determines the distance between a Tx and an Rx by measuring the signal propagation time, which requires precise time synchronization [26, 27].
- Time Difference of Arrival (TDOA): Utilizes differences in arrival times at multiple Rx to constrain the target's position to a hyperbola, reducing synchronization requirements [27].
- Angle of Arrival (AOA): Estimates the direction from which the signal is received using antenna arrays, with the intersection of multiple directions used to infer the Tx's position.
- Received Signal Strength (RSS): Infers distance by measuring signal attenuation, often modeled by the Friis transmission equation or empirically determined path-loss models [1].

2.6.2 Estimation Methods

Because real-world measurements are often noisy and affected by multipath propagation, localization becomes a statistical inference problem. Several key estimation methods are used:

- Linear and Non-linear Least Squares (LLS/NLS): These methods minimize the sum of squared errors between measured and predicted values. LLS is applicable when the problem can be linearized, while NLS directly handles non-linear relationships through iterative optimization [27].
- Maximum Likelihood Estimation (MLE): MLE identifies the position that maximizes the likelihood of observing the measurements, given a specified noise model. For a vector of observed measurements $\mathbf{z}$, the MLE formulation is:

  $$(\hat{x}, \hat{y}) = \arg\max_{x,y} \mathcal{L}(x, y; \mathbf{z}),$$

  where $\mathcal{L}(x, y; \mathbf{z})$ is the likelihood function. When measurement noise is assumed to be independent and Gaussian, this is equivalent to minimizing the weighted sum of squared residuals:

  $$(\hat{x}, \hat{y}) = \arg\min_{x,y} \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(f_i(x, y) - z_i\right)^2,$$

  where $f_i(x, y)$ is the theoretical measurement model for the ith anchor and σ_i^2 is the variance of the ith measurement [25, 26].
- Bayesian and Minimum mean square error(MMSE) Estimation: These techniques incorporate prior distributions or knowledge of the environment and compute minimum mean-square error estimates; they are especially useful in dynamic or uncertain settings [25].

2.6.3 Fingerprinting-Based Localization

In environments where multipath effects and NLOS conditions undermine geometric methods, fingerprinting has become a widely adopted alternative. Fingerprinting constructs a database (radio map) of signal characteristics, such as RSSI, channel impulse responses, or high-dimensional channel vectors, at known locations [22, 23]. During operation, new measurements are matched to those in the database using statistical or machine learning algorithms, including MLE, k-nearest neighbor, or neural networks. Advances in DT technology and ray tracing now enable the generation of large-scale synthetic fingerprint databases, improving coverage and reducing the need for extensive field measurement campaigns [2, 3].

2.6.4 Practical Challenges

Despite progress, several challenges remain for robust localization:

- Multipath and NLOS Effects: Reflections, diffractions, and obstructions introduce errors and biases in TOA, TDOA, and RSS measurements, necessitating advanced channel modeling or robust statistical methods [5, 3].

- Measurement Noise and Outliers: Real-world data often contain outliers and varying quality, which requires localization algorithms to implement robust estimation, adaptive weighting, or outlier rejection strategies [26, 27].
- Environmental Dynamics: Dynamic environments (e.g. moving objects, temporal changes in urban layouts) can degrade the accuracy of the fingerprint database, making frequent updates or algorithm adaptation necessary [3].

These combined techniques and practical considerations establish the foundation for advanced localization systems. Modern approaches often integrate robust statistical estimation (such as MLE), diverse measurement strategies, and DT-based synthetic data generation to achieve accurate and resilient positioning in real-world wireless networks.

3
Methods

This chapter describes the end-to-end methodology developed to analyze the impact of map inaccuracies on wireless localization within a DT framework. The workflow integrates map statistical analysis, randomized map generation, ray tracing simulation, localization algorithm evaluation, and coverage map analysis. The approach combines statistical rigor with realistic modeling, leveraging Python scripting and specialized libraries.

3.1 Image Method Verification

To analyze and verify the behavior of RT performance when building facades are randomized, a dedicated Python simulator was developed. The simulator models both deterministic and randomized reflection scenarios in 2D, providing an efficient environment for controlled Monte Carlo analysis. The objective of this analysis is to systematically quantify how geometric randomization, such as perturbing building surfaces using Gaussian noise, affects the accuracy and reliability of simulated RT reflections and subsequent localization estimates.

3.1.1 Simulation Setup

The simulator implements the image method for two parallel planes, representing idealized facades of buildings. In each experiment, a ray is emitted from a fixed source position towards a facade at a fixed departure angle and total path length. The reflection is then computed for both a reference (unaltered) plane and a randomly perturbed plane, where the endpoints of the facade segment are shifted by samples drawn from a Gaussian distribution. This process mimics realistic facade uncertainty as might occur in vendor-supplied maps.

3.1.2 Monte Carlo Workflow and Statistical Analysis

The analysis is performed over a large number of Monte Carlo trials (e.g., $N = 200$), with each iteration generating a new randomized facade and corresponding reflection path. For each trial, the simulator records:

- The intersection and endpoint of the reflected ray on both reference and randomized planes.
- The deviation (Euclidean distance) between the reference and randomized reflection endpoints.

- The normal vector angles and other geometric attributes of the perturbed surfaces.

This process is repeated for multiple geometric configurations, including different planes and incident angles, enabling a robust statistical characterization of reflection error distributions. Figures 3.1, 3.2, and 3.3 illustrate example results from these simulations.

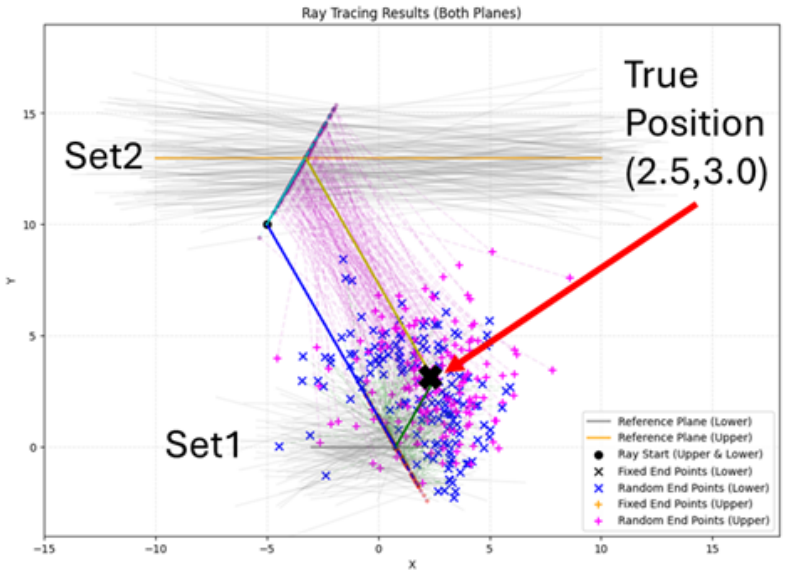

Figure 3.1: Visualization of ray tracing on reference and randomized planes, showing reflection endpoints and path deviations

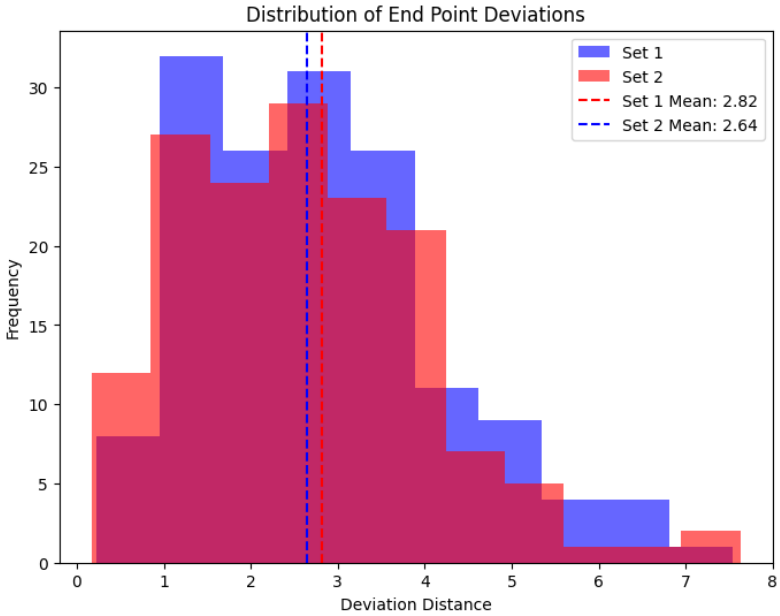

Figure 3.2: Histogram of endpoint deviations are shown individually for both planes.

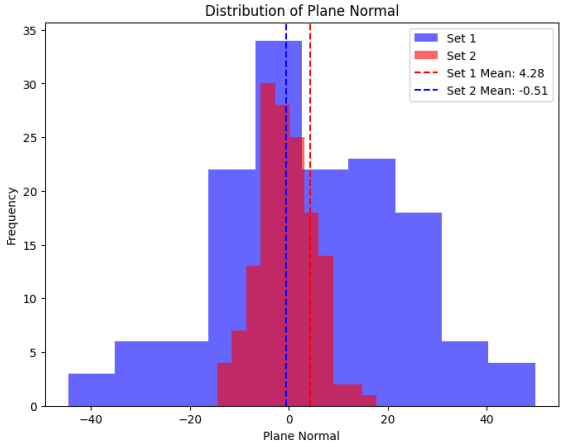

Figure 3.3: Histogram plot for upper and lower plane normal

3.1.3 Effect of Reflection Path Length and Facade Length

The simulation results demonstrate that reflection paths with a longer remaining distance after the last interaction are more sensitive to facade randomization, resulting in a higher spread of endpoint deviations (see Fig. 3.2 and Fig. 3.4). This effect is particularly pronounced for rays that traverse longer distances after reflecting, as minor geometric perturbations in the facade can lead to large shifts in the final intersection point.

Additionally, simulations reveal that shorter facade lengths contribute to a greater spread in the distribution of plane normals (see Fig. 3.3). This increased variability in surface orientation amplifies the deviation in ray endpoints, further degrading localization accuracy. Thus, both longer remaining reflection paths and shorter, more randomized facade segments can significantly impact the spatial consistency of the estimated positions.

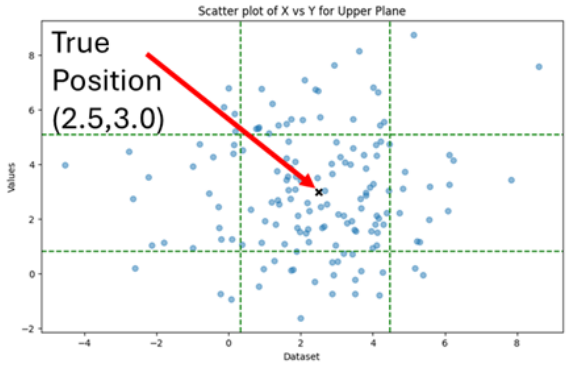

Figure 3.4: Scatter plot of X-Y with Respect to Rx Position

3.1.4 Purpose and Practical Implications

The primary purpose of this simulation is to deepen the understanding of how DT localization is affected by geometric map errors and to validate the robustness of

subsequent positioning algorithms. By visualizing the spatial distribution of reflection points, angle deviations, and their statistical properties, the simulator provides practical insight into the propagation of map-induced uncertainties through the RT pipeline. These insights guide the design of downstream localization methods and inform parameter choices for map randomization in larger simulation workflows.
It should be noted that this investigation only considers specular reflections as the primary propagation mechanism. Other types of interactions, such as diffraction and scattering, are not included in the current study but may warrant similar investigations in future work to fully capture their impact on localization accuracy.

3.2 Overall Workflow

The methodology is structured into distinct (see Fig. 3.5), interlinked stages:

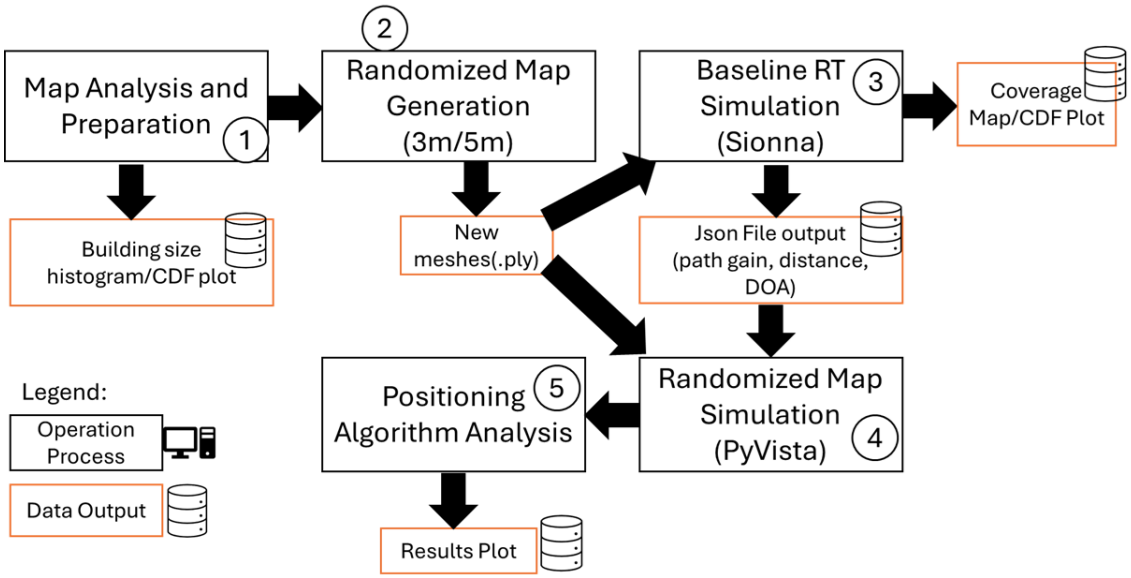

Figure 3.5: Overall Workflow of Simulation

1. **Map Analysis and Preparation:** Prior to simulation, each candidate map is statistically analyzed. Building footprints and dimensions are extracted and the distributions of building sizes are visualized using histograms and CDFs. This analysis directly informs the choice of rotation and translation bounds for subsequent map perturbations, ensuring that simulated inaccuracies realistically reflect vendor specifications.
2. **Randomized Map Generation:** To systematically study the effects of inaccuracies, randomized maps are generated by applying rotation and translation perturbations to buildings based on their centroid. Perturbations follow a truncated Gaussian distribution with a standard deviation chosen such that 95% of values fall within the specified map accuracy tolerance (e.g., $\pm 3\,\mathrm{m}$ or $\pm 5\,\mathrm{m}$). This controlled randomization aims to mimic real-world vendor map specifications.

3. **Baseline RT Simulation(Sionna):** The unperturbed "baseline map" is used to perform a reference RT simulation. This step extracts ground-truth metrics such as path distances, path types (reflection, diffraction), path gain, and AOD for each Tx-Rx pair. In a practical scenario, these quantities are analogous to the information that could be estimated at the base station in the real world, such as path distances and arrival angles, using sounding reference signals transmitted by the mobile terminal.

   Coverage Map and CDF Construction: Path gain (RSSI) values from each RT run are spatially aggregated to form coverage maps. For each scenario, the empirical CDF of the path gain is calculated, allowing visualization and benchmarking of coverage reliability. Changes in the CDF are analyzed under different perturbation scenarios to assess the impact of map errors on the predicted network performance.

4. **RT on Randomized Maps(PyVista):** For each randomized map realization, RT simulations are rerun using PyVista and the resulting path metrics are recorded. This collects more data for statistical study, quantifying the variability in localization and coverage due to map inaccuracies. Positioning errors, such as the absolute deviation, are statistically analyzed across many simulation runs on randomized maps. [21]

5. **Localization Algorithm Evaluation:** Multiple maximum likelihood-based localization algorithms are applied to the data ray-traced on the random maps. The robustness of each algorithm is assessed by computing position errors over many randomized maps. Outlier removal techniques are used, typically discarding rays with extreme path lengths or low path gain. Reliable absolute positioning estimates are obtained by computing the mean error after the outlier exclusion.

3.3 Software Stack

The core implementation leverages the following software components:

- **Sionna**: For GPU-accelerated ray tracing and propagation modeling, providing detailed multipath metrics for each simulation scenario.
- **PyVista**: Perform RT based on Sionna output and interactive visualization of both maps and simulation results.
- **Trimesh**: For efficient manipulation of 3D meshes, implementation of vertex perturbations.
- **Python**: As the orchestration and scripting language, enabling seamless integration of data analysis, simulation workflow, and visualization routines.

This stack allows rapid prototyping, batch simulation, and reproducible research while supporting large-scale Monte Carlo experiments necessary for statistical significance.

3.4 Implementation Steps

3.4.1 3D Map Information/Generation

The simulation environment utilized in this thesis is based on a detailed 3D map of the Kista district, covering an area of approximately 1636×1453,m^2. The 3D map data were generated using the Blosm plugin[1], an open-source tool for Blender that extracts building footprints and terrain information directly from OpenStreetMap (OSM) for a user-specified location. For this study, the Kista area in Stockholm was selected as the region of interest. Terrain elevation data were included, while vegetation was deliberately excluded to simplify the simulation model. [15]

To ensure modeling consistency and computational efficiency, all buildings in the DT are assigned a uniform material property corresponding to ITU concrete [7]. The accuracy of this map is sufficient for simulation-based analysis and serves as a fixed reference against which all map perturbations are applied. It is important to note that, as the aim of this study is to evaluate the relative impact of spatial inaccuracies, the absence of direct measurement data for validation does not affect the comparative conclusions. The planimetric accuracy levels of 3 m and 5 m used in the randomized maps are representative of the typical tolerances observed in commercially available building maps.

The Blosm import process generates individual building meshes, allowing for targeted manipulation and perturbation of each structure. The scenario evaluated throughout this thesis corresponds to a large plaza deployment, with a single Tx and twenty Rx distributed throughout the map area. The exact Tx and Rx locations, as well as the coverage region, are indicated in Figure 3.6. The spatial configuration of Txs and Rxs is designed to represent a realistic and challenging localization environment and forms the foundation for all subsequent simulation studies.

This 3D DT enables systematic analysis of the impact of spatial map perturbations, such as rotations, translations, and combined transformations, on radio propagation and localization accuracy in a representative urban environment.

[1] Blosm: `https://github.com/vvoovv/blosm`

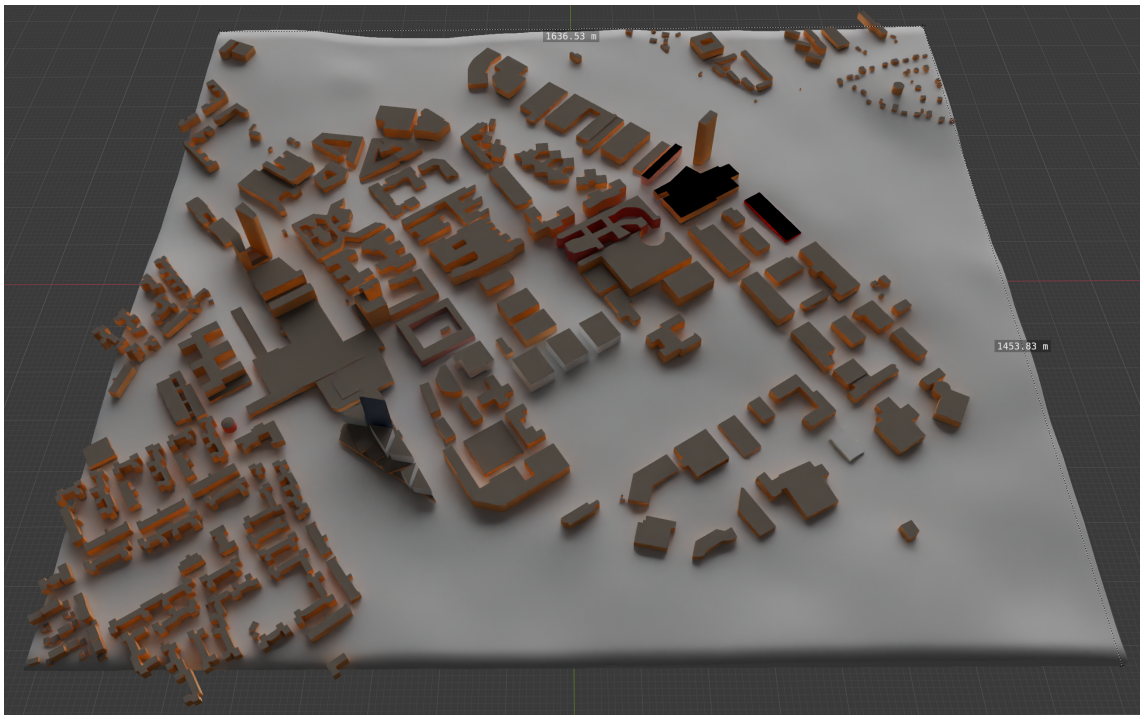

Figure 3.6: Kista 3D model generated with Blosm plugin in Blender based on Openstreet maps data.

3.4.2 Map Analysis and Randomized Map Generation

Prior to radio propagation simulations, all building meshes in the baseline map are analyzed to extract geometric statistics. Using the Trimesh library, the axis-aligned bounding box dimensions (X and Y) of each building are computed. The resulting distributions are visualized via histograms and CDFs in Fig. 3.7, providing a statistical foundation for map perturbation parameters. From Fig. 3.7, we observe mean of X dimensions is 51.46 and Y dimensions is 52.36 which show the distribution of building size in Kista.

For map randomization, two types of geometric inaccuracies are considered: rotation and translation, as well as their combination, which is referred to as a transformation. Rotation angles are determined based on the minimum of the X and Y dimensions of a building's bounding box. This approach ensures large buildings experience smaller angular deviations, it also minimize building colliding and overlapping on each other (see Fig. 3.8, left and center). Each building is rotated about its centroid and around the vertical (Z) axis, reflecting realistic errors observed in vendor-supplied maps.

Translations are generated independently in the X and Y directions, with offsets sampled from a truncated Gaussian distribution such that 95% of all translations remain within either ±3,m or ±5,m, matching typical commercial vendor specifications. The translation is applied about each building centroid (see Fig. 3.8, right).

Transformations combine both rotation and translation, compounding the effect of each source of map error and capturing more complex real-world scenarios. The statistical analysis in Fig. 3.7 shows that most buildings in the area are relatively

large, according to the applied perturbation method, means that their positional and angular changes under the chosen error bounds remain moderate.
By systematically varying both rotation and translation parameters in Monte Carlo simulations, this framework enables a robust evaluation of how realistic map inaccuracies impact radio propagation and localization performance. The map perturbation methodology and its effect on building geometries are summarized visually in Fig. 3.8.

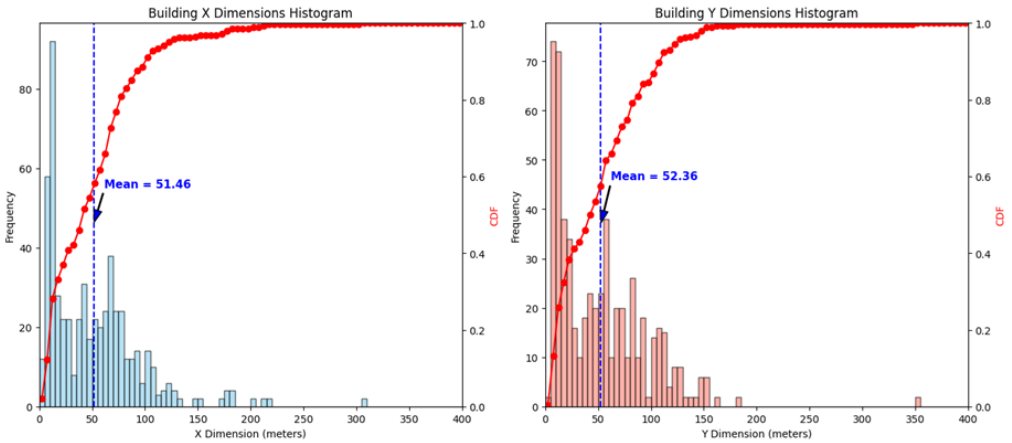

Figure 3.7: Histogram and CDF plot for X and Y Dimensions of Building Mesh Bounding Boxes.

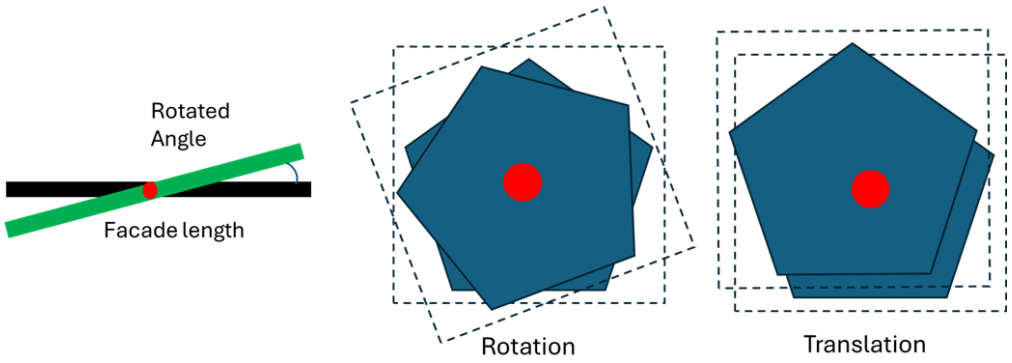

Figure 3.8: Left: Rotated Angle ;**Center:** Rotation ;**Right:** Translation.

3.4.3 Baseline RT Simulation

The baseline RT simulation establishes reference propagation metrics on the unperturbed, original map geometry. From a DT perspective, this is considered as the "real" world observations. Using the Sionna library, Tx and Rx are positioned according to experimental design, and the 3D environment is configured to include all relevant buildings and structures. For each Tx-Rx pair, the RT engine computes all feasible paths, accounting for line-of-sight (LOS), reflections, and diffraction up to a predefined maximum order of interactions.
The simulation outputs the following key metrics for every path:

- Path distance (derived from time-of-flight delays)
- Path type (LOS, single/multi-order reflections, diffraction)
- AOD/AOA
- Path gain (in both linear and dB scale)
- 3D bounce vertices for each multipath component

These metrics are saved as a JSON file for each Tx-Rx link. This serves as the "ground truth" for evaluating the influence of map perturbations in subsequent Monte Carlo experiments. (See Fig. 3.9)

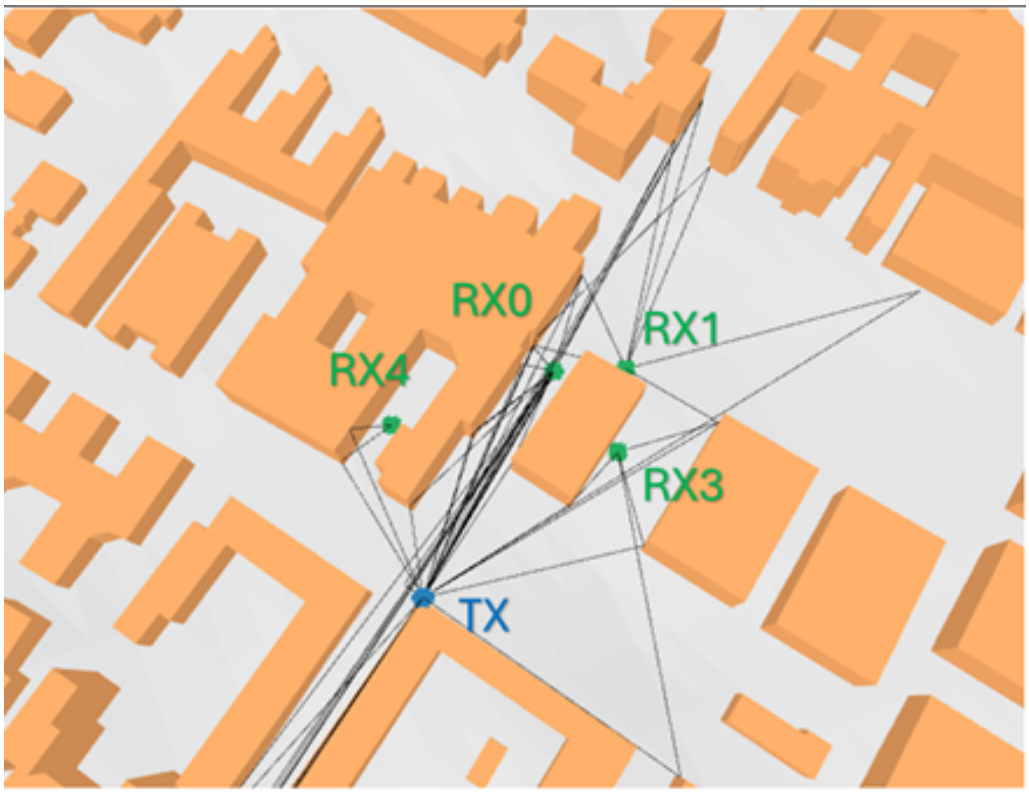

Figure 3.9: Sionna Ray Tracing on Baseline Map

3.4.4 Coverage Map Simulation and CDF Construction

For each simulation scenario, the Sionna RT engine generates path gain (RSSI) values over a defined spatial grid, resulting in a 2D coverage map that quantifies signal strength throughout the environment. Tx and Rx are placed according to the experimental setup, and the coverage map is constructed by sampling path gain at a high density of grid points. Key simulation parameters, such as the center, size and cell resolution of the coverage map, are set to ensure a consistent coverage evaluation for the base and randomized map configurations.

To assess the statistical impact of map inaccuracies, the empirical CDF of path gain is computed for each simulation. All path gain values sampled from the coverage grid are sorted, and the CDF is plotted to visualize the fraction of the environment above a given signal threshold. This method enables direct comparison of coverage reliability under different types of map perturbations.

Figure 3.10 presents an overlay of CDF plots for multiple randomized map variants—including baseline, rotated, translated, and transformed scenarios at both 3,m and 5,m perturbation levels. The figure aggregates results across all realizations, providing a comprehensive view of how geometric map errors (of varying type and magnitude) influence the overall coverage distribution. The close alignment of the CDF curves in this example suggests that, for the tested configurations, map perturbations introduce only modest variability in path gain statistics at the system level.

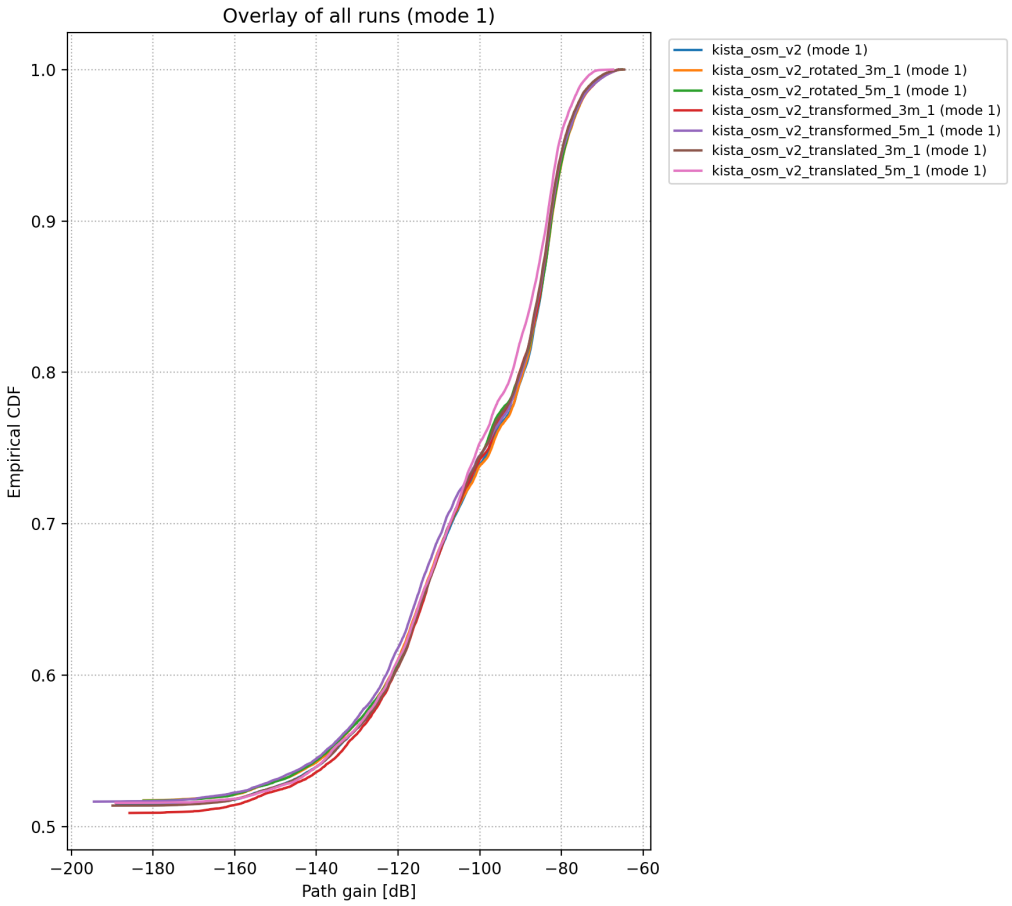

Figure 3.10: Sionna CDFs Plot Overlay

3.4.5 RT on Randomized Maps

To evaluate the impact of spatial map inaccuracies on wireless localization, a Monte Carlo approach is used, in which each randomized map realization undergoes an independent RT simulation. The simulation setup is driven by path information, such as path length and departure angles[2], originally extracted from the baseline (reference) map using Sionna. This baseline data ensures that for every Tx-Rx pair, the corresponding multipath structures in the randomized scenario closely mirror those present in the true environment, enabling controlled comparison.

[2]It is assumed i) that this data can be extracted in the real network by the base station from the reference signals transmitted by the mobile terminal and ii) that the reciprocity holds.

For each randomized map, the scene geometry is rebuilt in PyVista by importing the modified building meshes. Then, for each precomputed path from the baseline JSON, a ray is traced from the Tx with the same initial direction and length. At each interaction with the mesh, the code computes specular reflection directions based on the triangle normal. The ray continues until its path length is used up. This procedure reconstructs the paths as they would propagate in the altered geometry.

For every traced path, two key points are recorded: the last-bounce point (the final surface interaction before reaching the Rx) and the endpoint (the simulated ray's actual termination). The Euclidean distance between this endpoint and the true Rx location is calculated as the positioning error. The last-bounce point is later used as a direct input to the localization algorithm evaluation.

To document the process, a CSV file is generated for each simulation run, containing all key path features: Tx and Rx indices, path type, path length, last bounce point, endpoint location, and the resulting error. This tabulated dataset enables statistical characterization of how map perturbations propagate into localization performance.

A visual representation of the PyVista simulation is shown in Fig. 3.11. In this figure, **green spheres** indicate Tx locations, **blue spheres** denote Rxs, **black lines** show the path departure, **yellow spheres and lines** mark the last-bounce location and the path segment from last bounce to endpoint, and **red spheres** indicate the final simulated path endpoints. This visualization highlights how even modest spatial perturbations (e.g., 3 m or 5 m) can shift the structure of multipath propagation and affect the distribution of ray endpoints, thereby influencing localization results.

This detailed ray-level simulation using baseline path parameters and PyVista geometry is essential for robust and fair benchmarking of localization algorithms under realistic, map-induced uncertainty.

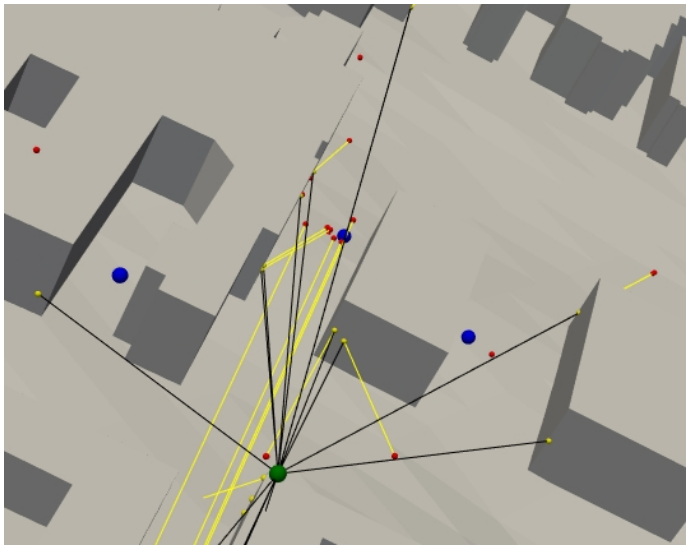

Figure 3.11: Visualization of ray tracing simulation on a randomized map using PyVista. Green spheres: Txs; blue spheres: Rxs; black lines: path depart from Tx; yellow spheres/lines: last-bounce locations and segments; red spheres: path endpoints.

3.4.6 Localization Algorithm Implementation

The evaluation of localization performance in this study is based on four maximum likelihood-based algorithms, each using multipath ray tracing data collected from the randomized map simulations. For every Tx-Rx pair, the inputs consist of N rays per Rx: each ray is described by its last-bounce position $\mathbf{B}[i]$, the sphere radius $r[i]$ (distance from the last bounce to the estimated Rx) and associated metadata such as path length and path gain.
The localization algorithms are implemented in Python using both closed-form and iterative optimization routines, operating on the output CSV files from the RT simulations. The pipeline processes each Rx independently as follows:

- **Data preparation:** For each Rx, the rays are filtered to remove those without valid bounces or with LOS-only characteristics. The last-bounce coordinates and sphere radii are extracted.
- **Algorithm application:** Five estimation approaches are applied, detail of each algorithm is explain in the following section.
- **Algorithm Summary:** A comprehensive explanation, method, parameters and remarks are listed in table 3.1.

Baseline Algorithm

The baseline approach provides a reference for localization accuracy by computing the mean Euclidean error between the estimated Rx position and the true Rx location. All available rays are used, and treated equally, with the final error value reflecting the average positional deviation across all rays without any optimization or selection.
Inputs:

- $\mathbf{x}_{\text{est}}$: endpoints computed from PyVista RT
- $\mathbf{x}_{\text{true}}$: Ground-truth Rx position

Algorithm:

1. For each ray $i = 1, \ldots, N$:
   - Compute the position estimate for the Rx (can be the endpoint or the geometric mean of bounces).
   - Calculate the error $e[i] = |\mathbf{x_{ext}}, \mathbf{i} - \mathbf{x_{true}}|$ for each ray.
2. Compute the mean error: Baseline Error $= \frac{1}{N} \sum_{i=1}^{N} e[i]$
3. Output: Baseline mean localization error.

Key properties: The baseline method does not apply any weighting, selection, or optimization—every ray is considered, and the error is simply averaged. This approach serves as a reference point for evaluating the improvement provided by more advanced algorithms. It is conceptually equivalent to reporting the mean distance error using all rays, providing a simple but informative lower benchmark.

Vanilla MLE

This approach treats all valid rays equally, seeking the Rx position that minimizes the sum of squared differences between measured and theoretical distances from

each last-bounce point. This is achieved by nonlinear least-squares optimization, yielding the position estimate $\hat{\mathbf{x}}$ that best fits all available ray constraints.

Inputs:

- $\mathbf{B}[i]$: Last-bounce 3D coordinates for ray i, $i = 1, \ldots, N$
- $r[i]$: Estimated radius for ray i (distance from last-bounce to endpoint)

Algorithm:

1. Initialize $\mathbf{x}$ as the centroid (mean) of $\mathbf{B}[i]\_i = 1^N$
2. Repeat until convergence:
   - For each $i = 1, \ldots, N$: compute the residual $e[i] = |\mathbf{B}[i] - \mathbf{x}| - r[i]$
   - Update $\mathbf{x}$ using a nonlinear least-squares solver (as in `scipy.optimize.least_squares`)
3. Output: $\hat{\mathbf{x}}$ (final position estimate)

Key properties: The vanilla MLE method is straightforward, using all rays and minimizing the squared sum of their distance errors. The optimizer minimizes the residual function, with x initialized at the mean of all bounce points. The resulting $\hat{\mathbf{x}}$ is the estimated Rx location.

Weighted MLE (Top-K Rays)

This variant of the MLE approach improves robustness to outliers by selecting and using only the most reliable multipath components for localization. In this implementation, rays are prioritized solely based on their path gain: the five rays with the highest gain (strongest signals) are selected, and the maximum likelihood estimation is performed using only this subset. By focusing on high-gain rays, the algorithm reduces the influence of noisy or error-prone measurements, which are more likely among weaker paths.

Inputs:

- $\mathbf{B}[i]$: Last-bounce 3D coordinates for ray i, $i = 1, \ldots, N$
- $r[i]$: Bounce-to-endpoint distance for ray i
- path_gain_db$[i]$: Path gain in dB for ray i

Algorithm:

1. For each ray i:
   - Normalize the path gains: $\text{gain}[i] = \frac{\text{path\_gain\_db}[i] - \min(\text{path\_gain\_db})}{\max(\text{path\_gain\_db}) - \min(\text{path\_gain\_db})}$
   - Compute score: $\text{score}[i] = -\text{gain}[i]$ (rays with higher gain have lower scores)
2. Select $K = \min(5, N)$ rays with the lowest score$[i]$ (the five rays with the highest gain).
3. Extract the selected rays' bounce points $\mathbf{B}[j]$ and radii $r[j]$.
4. Initialize $\mathbf{x}$ as the centroid of the selected $\mathbf{B}[j]$.
5. Repeat until convergence (only on the selected rays):
   - For each j in the selected subset: $e[j] = |\mathbf{B}[j] - \mathbf{x}| - r[j]$
   - Update $\mathbf{x}$ via nonlinear least-squares to minimize $\sum_j e[j]^2$
6. Output: $\hat{\mathbf{x}}_w$ (final estimated Rx position using only high-gain rays).

Key properties: By ranking and selecting rays using only the strongest path gains, this approach maximizes the likelihood that the used measurements are accurate and

physically meaningful, while minimizing the influence of multipath outliers. The method is computationally efficient and well-suited for scenarios where path gain is a reliable indicator of ray quality.

Gain-Weighted NLS

A gain-weighted nonlinear least-squares (NLS) algorithm is applied to improve localization robustness by prioritizing rays with strong path gains. Specifically, only rays with a path gain within 30dB of the maximum observed value are considered; weaker rays are ignored. Each residual is further weighted by a normalized function of its path gain, placing greater emphasis on strong, reliable rays and reducing the influence of noise-prone or spurious paths. This selective and weighted approach helps the algorithm avoid outliers that would otherwise degrade the localization accuracy.

Inputs:

- $\mathbf{B}[i]$: Last-bounce 3D coordinates for ray i, $i = 1, \ldots, N$
- $r[i]$: Estimated bounce-to-endpoint distance for ray i
- path_gain_db$[i]$: Path gain in dB for ray i

Algorithm:

1. For each ray i, define the gain threshold as $\max(\text{path\_gain\_db}) - 30$. Only rays with path_gain_db$[i] \geq$ threshold are assigned nonzero weight; all others are ignored.
2. Assign raw weights, $w[i]$:
$$w[i] = \begin{cases} \text{path\_gain\_db}[i] - \min(\text{path\_gain\_db}) & \text{;if path\_gain\_db}[i] \geq \text{threshold} \\ 0 & \text{;otherwise} \end{cases}$$
3. Clip all weights: $w[i] = \max(w[i], \epsilon)$ where $\epsilon = 10^{-2}$ (to avoid zero weights).
4. Normalize all weights: $w_{\text{norm}}[i] = w[i] / \max(w)$, unless all $w[i] = 0$.
5. Initialize $\mathbf{x}$ as the centroid of $\mathbf{B}[i]$.
6. Minimize the weighted residual sum using nonlinear least-squares:
   - For each i: $e[i] = (|\mathbf{B}[i] - \mathbf{x}| - r[i]) \cdot w_{\text{norm}}[i]$
   - Update $\mathbf{x}$ to minimize $\sum_i e[i]^2$ using a least-squares optimizer.
7. Output: $\hat{\mathbf{x}}_{\text{nls}}$ (final estimated RX position).

Key properties: By limiting the optimization to only the strongest rays (those within 30,dB of the maximum path gain), the algorithm effectively suppresses outliers due to noise, severe multipath, or weak/diffracted paths. The gain-based weighting further prioritizes the most reliable multipath components, enhancing the accuracy and stability of the localization solution.

Newton–Raphson ML

The Newton–Raphson ML algorithm estimates the Rx position by iteratively minimizing the sum of squared distance residuals between the estimated position and all bounce points, using analytic expressions for the gradient and Hessian of the cost function. This enables rapid, second-order convergence when initialized near the true position, though the method may be sensitive to local minima and numerical instability when data are ill-conditioned.

Inputs:

- $\mathbf{B}[i]$: Last-bounce 3D coordinates for ray i, $i = 1, \ldots, N$
- $r[i]$: Estimated bounce-to-endpoint distance for ray i
- $\mathbf{x}\_0$: Initial guess for Rx position (typically the mean of $\mathbf{B}[i]$)
- T: Maximum number of iterations (default: $T = 10$)

Algorithm:

1. Initialize $\mathbf{x} \leftarrow \mathbf{x}\_0$
2. For $t = 1, \ldots, T$ (iterate up to the maximum allowed):
   - For each $i = 1, \ldots, N$:
     - Compute $\mathbf{d}\_i = \mathbf{x} - \mathbf{B}[i]$
     - Compute $d\_i = |\mathbf{d} * i|$
     - Let $d * i, \text{safe} = \max(d\_i, \epsilon)$ (to avoid divide-by-zero, with $\epsilon = 10^{-6}$)
   - Compute the gradient vector: $g = 2 \sum_{i=1}^{N} \frac{d_i - r[i]}{d_{i,\text{safe}}} \mathbf{d}_i$
   - Compute the Hessian matrix: $H = 2 \sum_{i=1}^{N} \left[\frac{\mathbf{p}_i \mathbf{p}_i^T}{d_i^2} + (d_i - r[i]) \left(\frac{I}{d_i} - \frac{\mathbf{p}_i \mathbf{p}_i^T}{d_i^3} \right) \right]$ where $\mathbf{p}\_i = \mathbf{d} * i$ (as a column vector), $d\_i = d * i, \text{safe}$, and I is the 3×3 identity matrix.
   - Solve the linear system $H \cdot \Delta = g$ for the Newton step Δ
   - Update the estimate: $\mathbf{x} \leftarrow \mathbf{x} - \Delta$
3. Return final position estimate $\mathbf{x}$

Key properties: The Newton–Raphson approach efficiently leverages analytic derivatives for fast convergence, but requires robust initialization and regularization for stability—especially when bounce points are collinear or nearly coincident.

Genie-Aided MLE (Lower Bound)

The Genie-Aided MLE algorithm adaptively selects the best available rays for position estimation using only observable information without access to the true Rx position. For each Rx and map, Genie prioritizes rays according to a weighted score based on low endpoint error, short propagation length, small bounce radius, and high path gain, always giving preference to type 1 rays(specular reflection) if available. For each candidate subset of $n = 3$ to 10 rays, all four previous MLE algorithms are executed, and the result with the lowest estimated error is selected as the Genie output. If none of these Genie solutions outperform the baseline algorithms (Vanilla MLE, Weighted MLE, Gain-Weighted NLS, Newton–Raphson), Genie falls back to the best-performing baseline result, thus guaranteeing it never performs worse than any baseline.

Inputs:

- All per-ray metadata as above: last-bounce positions, bounce radii, propagation lengths, path gain, and path type.

Algorithm:

1. For each candidate subset size $n = 3, 4, \ldots, 10$:
   - Sort all rays by a composite priority score:
     - Lower endpoint error, length, and radius, and higher path gain are favored (all features normalized).
     - Type 1 rays (strongest) are always selected first if available.

- Select n top rays as the candidate set (filling with lower-priority rays if not enough type 1 rays).
- Compute Genie position estimates by running all four baseline MLE methods on this candidate set.
- Track the candidate and method yielding the lowest position error with respect to the true Rx.

2. Compare Genie's best error to the lowest error among all four baseline methods (computed on all rays for each Rx).
   - If Genie's error is lower, output Genie's position estimate.
   - Otherwise, output the baseline position estimate with the lowest error.
3. Output: Genie solution (guaranteed lower bound for this set of baseline methods).

Key properties:

- Genie-Aided MLE uses only observable ray features, without "cheating" by using the true Rx as a virtual ray.
- By construction, its reported error is always less than or equal to any of the four baseline algorithms for every Rx and map.
- Provides a practical lower bound for evaluating further improvements to ray selection or robust MLE localization.

For each algorithm, the errors are computed as the Euclidean distance between the ground-truth and the estimated Rx position. The results are compiled into a CSV file for further analysis and visualization. The comparative results of these algorithms, including error distributions and robustness to spatial inaccuracies, are further detailed and interpreted in Chapter 4.

Method	**Formula/Objective**	**Data Used**	**Optimization Method**	**Key Difference / Note**
Baseline	mean $= \frac{1}{N}\sum_{i=1}^{N} e[i]$	All rays	Mean	All rays used, equally weighted
Vanilla MLE	$\min_{\mathbf{x}} \sum_{i=1}^{N} (\|\mathbf{B}_i - \mathbf{x}\| - r_i)^2$	All rays	Least Squares	All rays used, equally weighted
Weighted MLE (Top-K Rays)	$\min_{\mathbf{x}} \sum_{i \in S_5} (\|\mathbf{B}_i - \mathbf{x}\| - r_i)^2$, S_5: 5 rays with lowest score (high gain, close to median length)	Top 5 rays by composite score	Least Squares	Only strongest/trustworthy rays
Gain-W NLS	$\min_{\mathbf{x}} \sum_{i \in S_{30}} \left(w_i(\|\mathbf{B}_i - \mathbf{x}\| - r_i)\right)^2$, $w_i = \frac{\max(\text{gain}_i - \min(\text{gain}), 0.01)}{\max(\text{gain})}$	All rays within 30 dB of max gain	Weighted Least Squares	Discards weak rays, weights by gain
NR-MLE	Same as Vanilla MLE, but solved via Newton–Raphson updates	All rays	Newton–Raphson (analytic grad/Hessian)	Second-order optimizer, can converge faster
Genie-Aided MLE (Lower Bound)	Adaptive: (i) For $n = 3$ to 10, select n rays with lowest composite priority score (normalized error$_m$, length, radius, high path gain, type priority) (ii) Always prioritize type 1 rays (iii) Run all 4 MLE variants on each candidate set (iv) Report the minimum error; if not lower than all baselines, fallback to best baseline result	Top 3–10 prioritized rays (type 1 if available)	All above	Practical lower bound: always $\leq$ best baseline; no oracle/ground-truth RX used

Table 3.1: Summary of all localization algorithms implemented and compared.

4

Results

This chapter presents the main simulation results, focusing on how map inaccuracies affect radio propagation and localization accuracy in the Kista DT environment. Analyses of coverage maps, CDFs of path gain, and localization error are provided for both baseline and perturbed scenarios. The presented results highlight the influence of map errors, ray types(specular reflection, diffraction), and algorithm selection on system performance.

4.1 System Model

The simulation environment for this thesis is based on the Kista map described in Section 3.4.1 and shown in Fig. 3.6. The Sionna ray tracing engine is used for the propagation study. The deployment scenarios are defined in a large plaza layout. Each simulation uses a single Tx and a predefined set of Rx locations. The simulation leverages an XML scene description, with all runs performed in batch across multiple variants of the scene with varying map perturbations. The Tx, Rx and simulation parameters are summarized in Table 4.1.

The key simulation settings are as follows:

- **Ray tracing platform:** Sionna RT (GPU-accelerated).
- **Tx configuration:** Single isotropic antenna at a fixed location per scenario.
- **Rx configuration:** 20 RXs placed in a plaza .
- **Ray tracing parameters:** Up to 5 bounces, reflections and diffractions enabled, scattering disabled.
- **Coverage analysis:** Each run produces a 2D coverage map (path gain) and empirical CDF, saved for post-processing.
- **Map variants:** Simulations are conducted on maps with 3-metre and 5-metre spatial accuracy, including 3 variants for each on rotation, translation, and combined transformation (rotation + translation).
- **Random seed:** All runs are reproducible via a fixed random seed.

Table 4.1: System Model Simulation Settings

Parameter	**Setting**
Number of TXs	1
Number of Rxs	20
TX Array Type	Isotropic, 1×1
RX Array Type	Isotropic, 1×1
Ray Tracing Depth	Up to 5 bounces
LOS	Enabled
Reflection/Diffraction	Enabled
Scattering	*Disabled*
Random Seed	42 (fixed)

4.2 Propagation Results for Random Map Variants

The impact of map perturbations on radio coverage was evaluated through Sionna-based ray tracing simulations, generating cumulative distribution function (CDF) plots and 3m/5m randomized map variant (see Fig 4.2 and 4.1).

Figure 4.2 overlays the CDFs of path gain for the original (baseline) map and for each perturbed variant, grouped by perturbation type (3m and 5m planimetric accuracy). Interpretation of these CDFs is closely tied to the physical meaning of coverage: for a given path gain threshold, the CDF value indicates the fraction of locations with equal or lower signal strength.

A key observation is that, in the baseline map (black line), a substantial portion of the environment is effectively uncovered at path gain, -180dB. For instance, a minimum required path gain of -180dB for system operation, only about 32% of the coverage map achieves sufficient signal strength, as indicated by the sharp CDF increase at this threshold; conversely, 68% of the environment, largely corresponding to grid points inside buildings or deep NLOS regions, receives no coverage at all. This spatial distribution of coverage is visually shown in Figure 4.1, where extensive white areas indicate uncovered or unreachable regions in both the 3m and 5m scenarios.

The introduction of spatial perturbations to the map geometry leads to variation in coverage. For the 3m perturbed maps, coverage fractions remain close to the baseline: the majority of random realizations cluster near 31–34% coverage, with minor deviations depending on the specific perturbation. The 5m case exhibits a wider spread in coverage, with some realizations achieving slightly better or worse coverage compared to the baseline; this larger variability is expected given the larger perturbation magnitude, though the limited number of random realizations (nine per case) constrains the statistical interpretation.

Examining the tails of the CDFs offers additional insight into the variability introduced by map errors. For the baseline, the 5^{th} percentile of covered locations (i.e., 5% out of the 32% that are covered) occurs at a CDF value near 0.70 (i.e., $68\% + 1.6\%$). At this percentile, the range of path gain values in the 3m case spans roughly 18,dB (from -153,dB to -135,dB), while the 5m case covers a broader 25,dB range (from -160,dB to -135,dB). This indicates that, for locations near the margin of coverage, map inaccuracies can induce considerable signal fluctuations that could be significant in bit error rate or link budget analyses.

In contrast, the upper tail of the CDF (from the 85^{th} to 100^{th} percentile) remains relatively stable across all map variants. This segment corresponds primarily to locations with LOS to the Tx, which are largely unaffected by map perturbations. Notably, these high-CDF regions account for approximately 50% of the locations with valid coverage, suggesting that in this plaza scenario, map errors have limited influence on the most reliably covered areas.

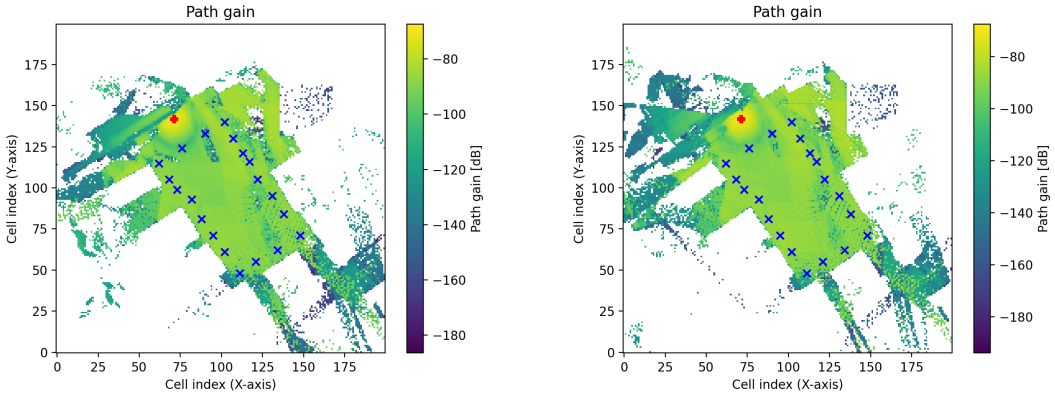

Figure 4.1: Sionna Coverage simulation. Left:3metre map: Right:5metre map

Finally, it is important to emphasize that the large fraction of white (uncovered) areas visible in Figure 4.1—approximately 65% of the map—results from grid points situated within buildings or in deep shadowed regions where the ray tracer fails to identify valid signal paths. These findings underline the fact that, while map perturbations introduce nontrivial uncertainty into marginal coverage regions, the overall impact on the robustly covered areas is modest. The observed range of coverage variation due to map inaccuracies should be taken into account in system design and in applications sensitive to edge-of-coverage effects.

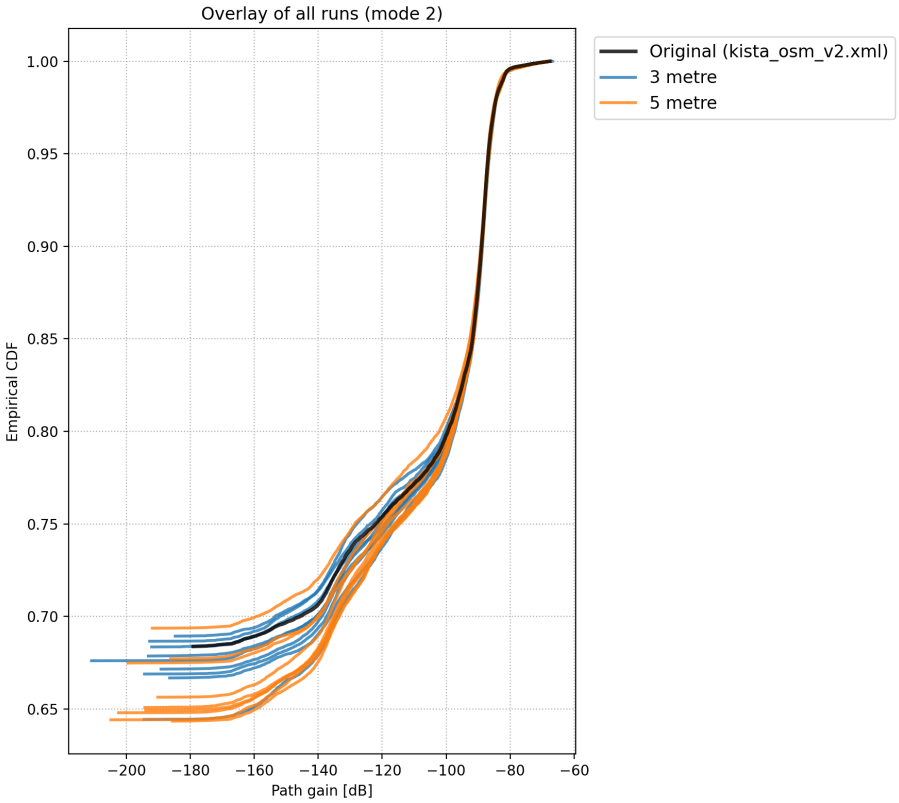

Figure 4.2: Sionna simulation result of CDF overlay group by type

4.3 Positioning Results

4.3.1 DT Results for Random Map Variants

As described in Section 3.2, Sionna is employed to perform RT simulations on the baseline map. Figure 4.3 illustrates the scenario: the left panel depicts the configuration with 20 RX, while the right panel shows a clearer view for just one of the 20 RX's. Further examination of Fig. 4.3 highlights that individual Rx locations receive up to 43 distinct ray paths, which include line-of-sight (LOS), specular reflections, and diffracted components. The Sionna RT simulation generates detailed output in the form of JSON files, including metrics such as ray path distances, path gains (in dB), and AOD. Subsequently, PyVista leverages the baseline map information and these reference metrics to conduct RT simulations across all randomized map realizations, enabling direct comparison of localization performance under varying map inaccuracies.

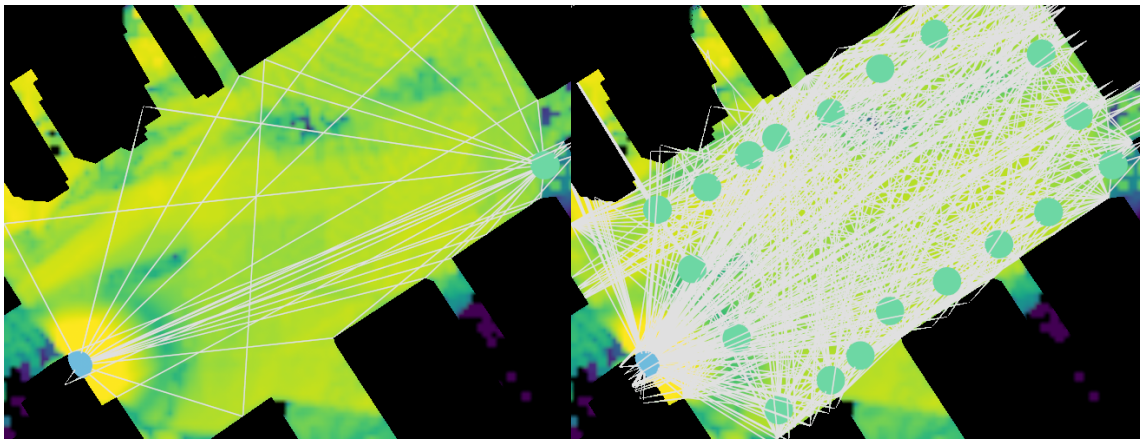

Figure 4.3: Left: 1 Rx Ray Tracing for clarity; Right: 20 Rx Ray Tracing.

The RT simulations performed with PyVista are restricted to purely specular reflections, as diffraction or scattering mechanisms are not implemented due to time constraint. Figure 4.4 presents the consolidated absolute position deviations for two ray categories: type 1 (specular reflections) and type 2 (diffracted rays, as provided by Sionna). As shown in Figure 4.5, approximately 80% of the type 1 rays exhibit position errors below 20 metres, indicating that these rays can be utilized in the positioning algorithm with relatively high confidence. In contrast, Figure 4.6 shows that only about 10% of type 2 rays fall below the same error threshold, rendering them much less reliable for accurate localization. This significant disparity in ray reliability highlights a key challenge: Diffraction and scattering not implemented in PyVista greatly complicates the ray selection process and limits the overall effectiveness of the positioning algorithm.

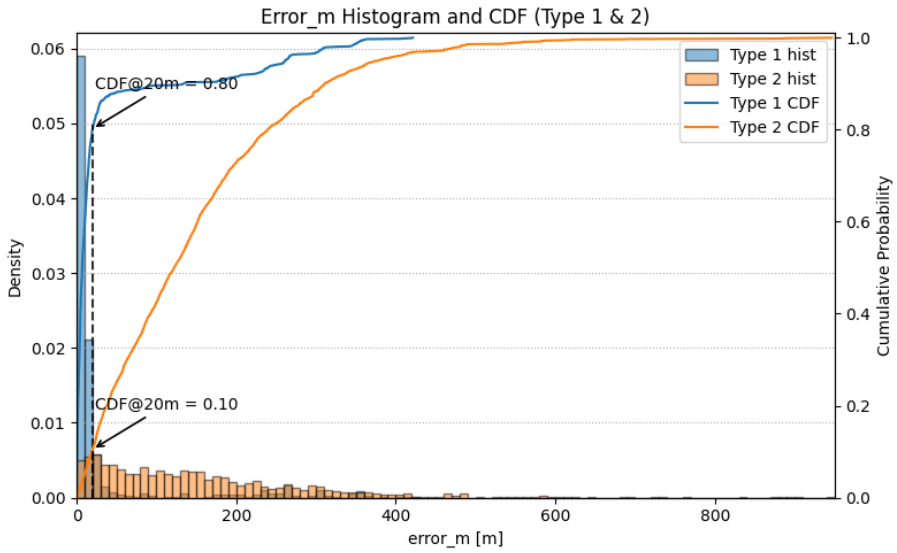

Figure 4.4: PyVista RT Result by Rays Index Overall Errors

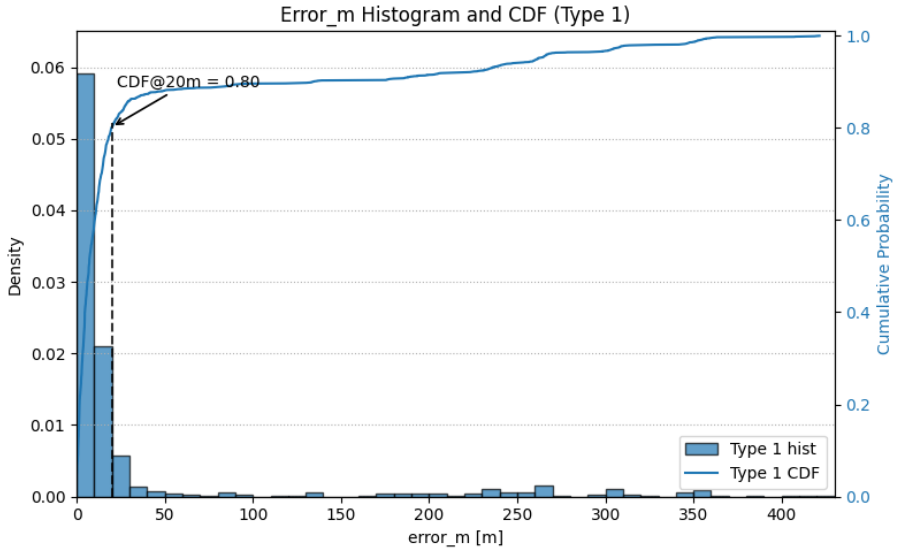

Figure 4.5: PyVista RT Result by Rays Index Type 1 Errors

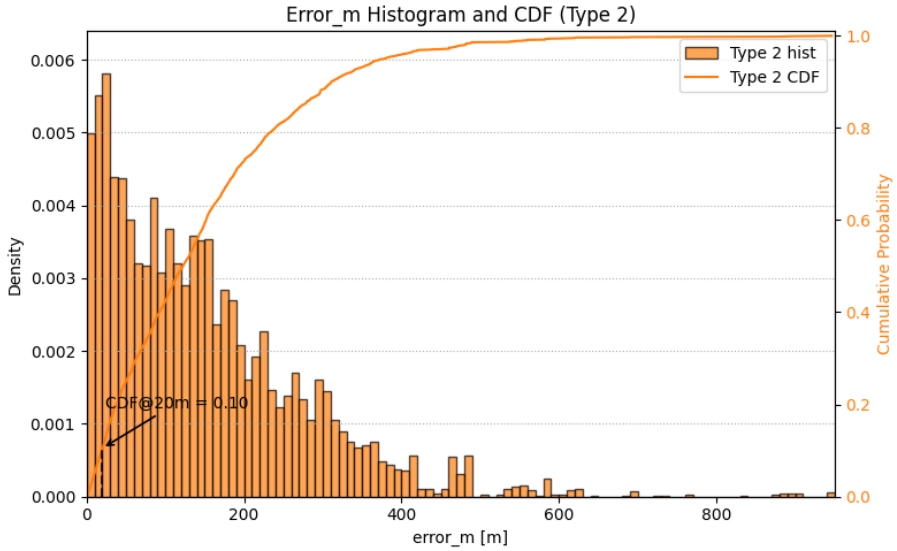

Figure 4.6: PyVista RT Result by Rays Index Type 2 Errors

Based on the simulation results, the Genie-aided-MLE algorithm described in Section 3.4.6 achieved a localization deviation of only 0.93 metres on the original map, in contrast to the baseline case where no estimation algorithm was applied which yielded a mean deviation of 117.49 metres. The Genie-aided-MLE serves as a lower bound benchmark for algorithmic performance, representing an ideal scenario with optimal ray selection.

When comparing the localization results for the 3-metre and 5-metre perturbed map scenarios, the 3-metre maps typically demonstrated superior performance from the various positioning algorithms results see Figure 4.7. Specifically, the weighted-MLE algorithm achieved a mean localization error of 18.15 metres for the 3-metre

maps, compared to 25.45 metres for the 5-metre maps(see Fig. 4.7). This trend was similarly observed for the gain-weighted-NLS algorithm. In contrast, both the Vanilla-MLE and Newton–Raphson ML algorithms showed performance patterns closely aligned with the baseline case and did not produce substantial improvements. This underperformance is primarily due to the inability of these algorithms to robustly exclude rays that incur large errors. The rays with large errors are often the result of wrongly treated diffractions or significant deviations in the ray paths caused by building randomization. This severely compromises the accuracy of the location.

Each algorithm presents its own set of advantages and limitations, largely stemming from the challenge of effective ray selection when only path gain and path length are available as discriminative features. Algorithm perform vary from Rx to Rx due to the selection of rays, individual Rx will observe each position algorithm performance to vary (see Fig. 4.8). These metrics alone do not fully capture the intricacies of error propagation in RT based localization. Additionally, the use of PyVista's ray tracing for multipath simulation occasionally resulted in rays traversing solid objects, an issue arising from floating-point precision issues in mesh intersection detection and from initializing subsequent ray segments too close to surface boundaries following reflections.

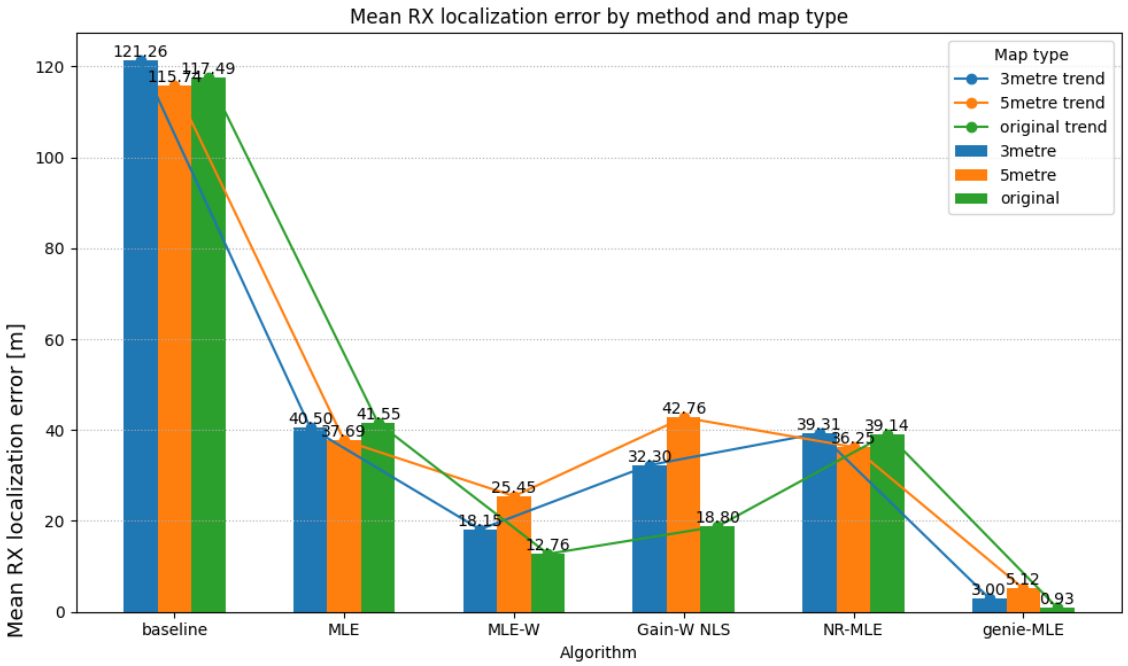

Figure 4.7: Overall Result for each Algorithms Implemented

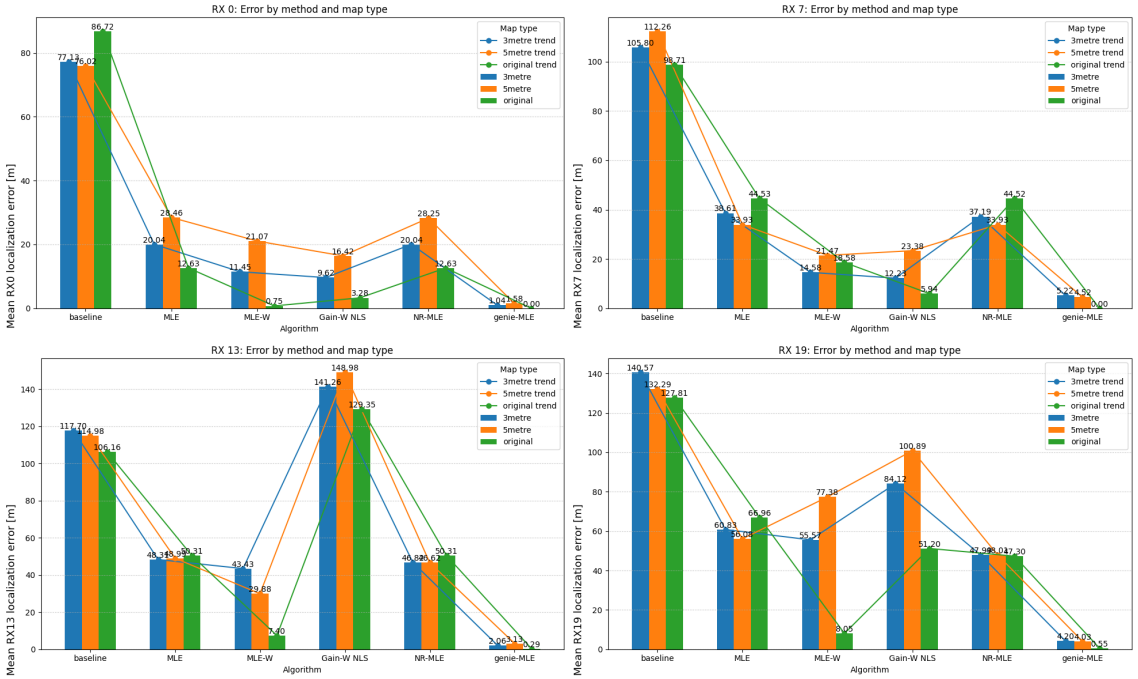

Figure 4.8: 4 RX Results with All Algorithms Implemented

4.4 DT Results for Specular Reflection Type 1 Only

To further isolate the performance impact of high-confidence ray paths, this section presents DT localization results considering only type 1 specular reflections using PyVista RT simulations. As discussed previously, approximately 80% of type 1 rays yield absolute position deviations below 20 metres (see Fig. 4.5), establishing them as the most reliable subset for localization algorithms in the absence of diffraction or scattering information.

The Genie-aided-MLE algorithm, which serves as an idealized benchmark through perfect ray selection, achieved mean localization errors of 1.11 metres for the 3-metre maps and 2.05 metres for the 5-metre maps (see Fig. 4.9), respectively. These results represent the lower bound for achievable localization accuracy when limited exclusively to type 1 rays.

Among practical algorithms, the gain-weighted NLS also demonstrated robust performance, with mean localization errors of 2.53 metres for the 3-metre scenario and 4.47 metres for the 5-metre scenario. The increased deviation observed in the 5-metre maps reflects the broader spatial perturbations inherent to their construction, leading to reduced confidence in the resulting position estimates. Nevertheless, both Genie-aided and gain-weighted algorithms substantially outperformed less discriminative methods, consistent with the higher reliability associated with specular reflection paths.

These results reinforce the importance of confident ray selection in practical DT localization workflows. By restricting analysis to type 1 rays(specular reflection), it is possible to achieve significant improvements in localization accuracy demonstrating that, even under moderate spatial map inaccuracies, robust localization remains feasible when leveraging only high-confidence multipath information, see Fig. 4.10.

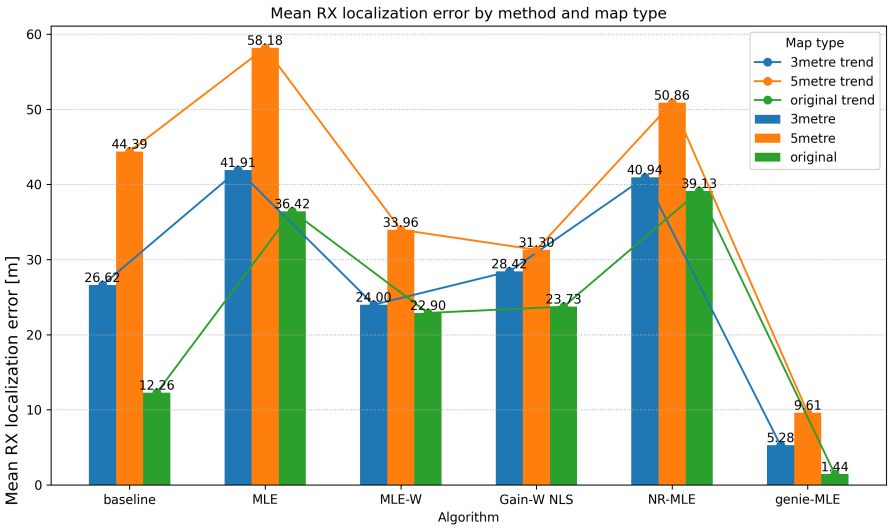

Figure 4.9: Type1 Only Overall Result for each Algorithms Implemented

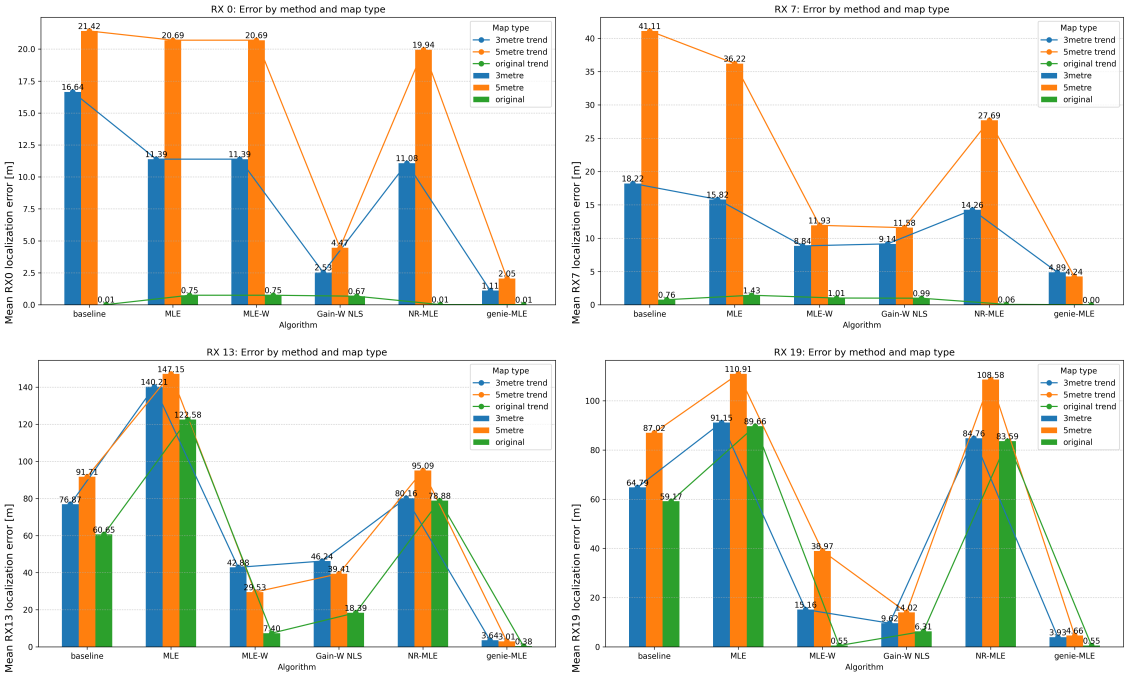

Figure 4.10: 4 RX Type 1 Only Results with All Algorithms Implemented

4.4.1 Ray Selection and Impact on Positioning Algorithms

Accurate ray selection is one of the central challenges in RT-based localization algorithms, as the correlation between observable features and localization error can be complex and scenario dependent. The positioning algorithm must rely primarily on physical metrics such as path length, bounce radius, and path gain to discriminate between reliable and error-prone rays. However, these metrics alone do not always offer clear separation between high- and low-quality paths, especially in dense multipath environments or when map inaccuracies are present.

To illustrate the complexity of this problem, correlation matrices were computed for key ray attributes (absolute position error, path length, bounce radius, and path gain) using the data from both type 1 (specular reflection) and type 2 (diffracted) rays. Figure 4.11 shows the correlation matrix for all ray types, while Figure 4.12 focuses exclusively on type 1 rays.

A few important observations emerge from these plots:

- For all ray types, the correlation between error and path gain is negative, suggesting that stronger rays are generally associated with lower localization error. However, the relationship is far from perfect, particularly when type 2 (diffracted) rays are included.
- When only type 1 rays are considered, the correlation between path gain and both error and path length becomes more pronounced (see Fig. 4.12). Despite this, a non-negligible spread remains—indicating that even high-gain reflections can occasionally produce large deviations.
- Low path gain or long path length rays tend to correspond to larger errors, but the presence of map inaccuracies and complex geometry can produce significant outliers. This further complicates robust ray selection, as removing too many rays risks discarding valuable information, while including too many can degrade accuracy.

These results demonstrate why ray selection remains a bottleneck for high-precision positioning. The correlation structure (see Figs.4.11 and4.12) suggests that naïve selection based on any single metric—such as path gain or path length may not be sufficient for robust performance. Instead, future work may require more sophisticated selection strategies, potentially incorporating multi-feature analysis or machine learning approaches to distinguish reliable rays.

Overall, the difficulty in selecting appropriate rays for positioning algorithms is a fundamental limitation that must be addressed to further improve localization performance particularly as system complexity and environmental uncertainty increase.

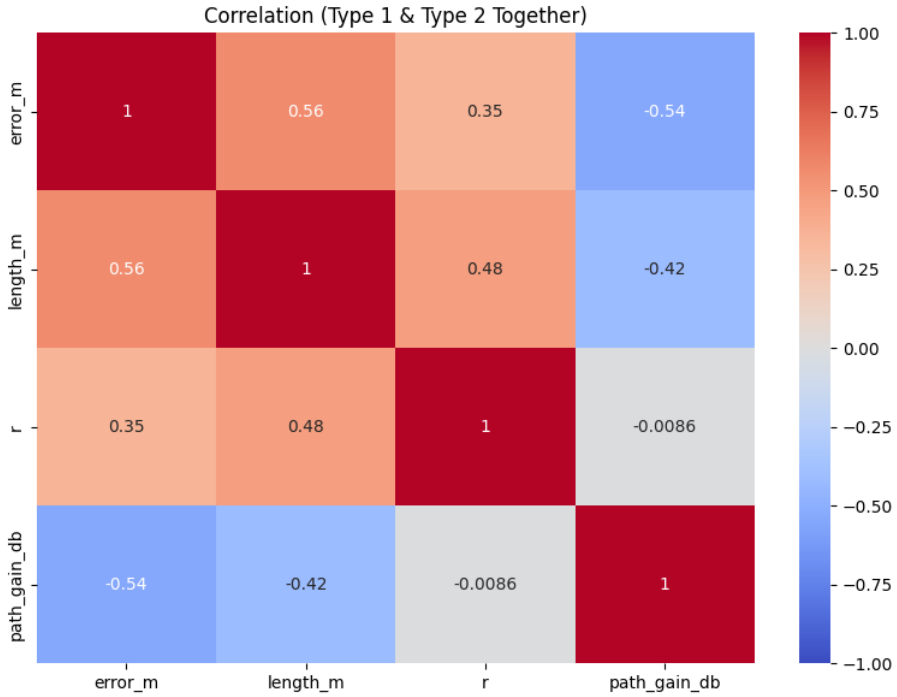

Figure 4.11: Correlation matrix for all ray types (specular reflection and diffraction).

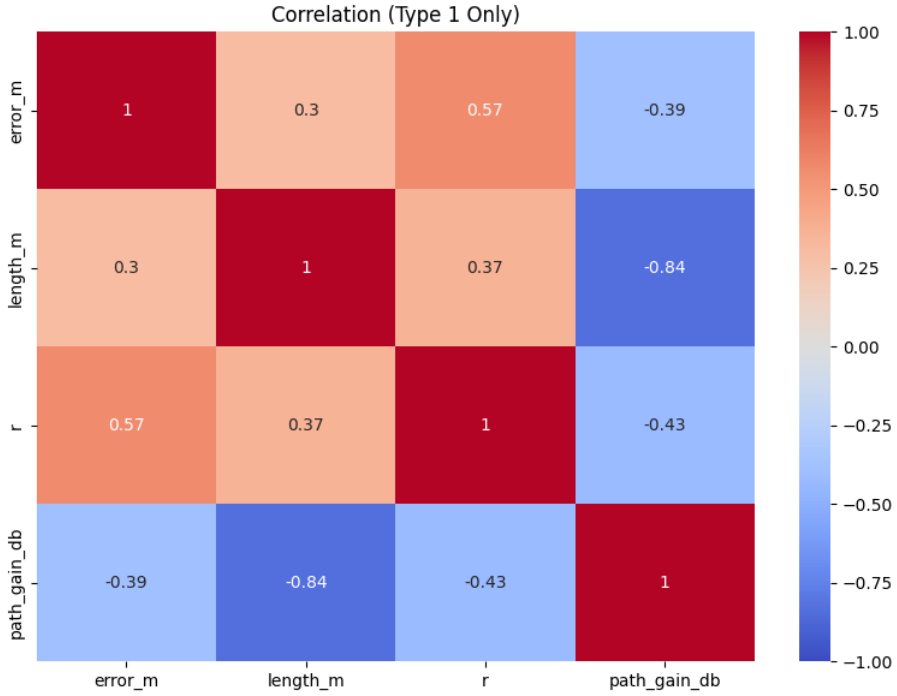

Figure 4.12: Correlation matrix for type 1 (specular reflection) rays only.

4.5 Discussion and Key Findings

Type	Baseline	MLE-W	Gain-W-NLS	Genie-Aided-MLE
3 Metre	121.26	18.15	32.30	3.00
5 Metre	115.74	25.45	42.76	5.12

Table 4.2: All Rays Comparison of Mean Errors Result 3metres vs 5metres

Type	Baseline	MLE-W	Gain-W-NLS	Genie-Aided-MLE
3 Metre	26.62	24.00	28.42	5.28
5 Metre	44.39	33.96	31.30	9.61

Table 4.3: Type1 Comparison of Mean Errors Result 3metres vs 5metres

Table 4.2 and 4.3 summarizes the mean localization errors achieved under various algorithms for both the complete set of rays (including all path types) and for type 1 rays (specular reflections only), across both 3-metre and 5-metre map accuracy scenarios. The Genie-Aided-MLE serves as an idealized benchmark, providing a theoretical lower bound for achievable error by leveraging perfect ray selection.

Several trends are evident from the results. When all rays are considered, the baseline error is exceedingly high, highlighting the detrimental impact of unfiltered multipath and diffracted rays. Weighted-MLE and Gain-Weighted-NLS algorithms provide substantial improvements, particularly in the 3-metre scenario, yet their effectiveness is still limited compared to the Genie-aided case. When restricting the analysis to only type 1 rays, mean errors are dramatically reduced across all algorithms, with Genie-Aided-MLE achieving localization errors as low as 1.11 metres for the 3-metre maps and 2.05 metres for the 5-metre maps. Gain-Weighted-NLS also performs well under this restriction, but a performance gap remains relative to the ideal.

The superior results for type 1 rays underscore the importance of reliable ray selection. However, as the current implementation with PyVista only supports specular reflections, and does not incorporate diffraction or scattering, the ability to robustly identify high-confidence rays is fundamentally constrained. This limitation is further compounded by the exclusive reliance on path distance and path gain as discriminative features—information that is insufficient to fully characterize the underlying propagation environment, especially when using a single base station.

It should be noted that the use of only one base station is a deliberate design decision, motivated by the additional synchronization and channel state information requirements associated with multiple anchor nodes requirements which fall outside the scope of this thesis. As such, the ray selection challenge remains an open problem. The absence of a robust diffraction algorithm in PyVista, primarily due to the complexity and development time required, precluded further investigation into

hybrid or more advanced multipath selection strategies.

In conclusion, while significant improvements in localization accuracy are possible through careful ray selection, the constraints imposed by the available information and simulation tools limit the extent to which optimal results can be routinely achieved in practice.

5
Conclusion

This thesis demonstrates that the accuracy of DT maps plays a critical role in wireless localization performance, with higher-fidelity maps and robust ray selection algorithms substantially reducing positioning errors. The results highlight both the promise of DT-based localization and the need for continued advances in map quality and propagation modeling to fully realize its potential.

5.1 Summary

This thesis has systematically investigated the impact of spatial inaccuracies in DT maps. Specifically, the impact of building rotations and translations on wireless localization accuracy was investigated within the ray tracing (RT) simulation framework. Comparison between 3- and 5-metre map perturbations revealed that higher-fidelity maps (3-metre accuracy) consistently yield superior localization performance across all evaluated algorithms. Among these, the Genie-aided-MLE approach, representing a lower bound benchmark, achieved the best results, particularly when restricted to high-confidence specular reflection (type 1) rays.

The findings proof that DT, when constructed with sufficiently accurate spatial data, are effective at supporting localization in wireless environments. In particular, the weighted-MLE and gain-weighted-NLS algorithms demonstrated marked improvements in mean error over the baseline, with 3-metre maps providing a mean localization error as low as 11.39 metres (type 1 rays), compared to 20.69 metres for the 5-metre case. These results underscore the critical importance of both map fidelity and algorithmic ray selection.

The analysis of coverage maps and CDFs demonstrated that most receiver locations are in LOS with consistently high path gain, and the overall impact of map perturbations on signal coverage is relatively minor. However, the CDFs show only small differences between map variants, the localization accuracy is much more sensitive to spatial errors in the map. This suggests that, for LOS-rich scenarios, positioning accuracy is a more sensitive and informative metric for assessing the impact of map inaccuracies than aggregate coverage or path gain statistics alone.

Overall, the results indicate that map inaccuracies directly influence the achievable accuracy of wireless positioning. Applications demanding sub-metre or low-metre localization precision, such as indoor navigation, emergency response, or advanced

asset tracking, require careful control over map quality, emphasizing the necessity of rigorous DT construction and maintenance practices.

5.2 Improvements and Future Work

Several avenues exist for further advancing the fidelity and applicability of DT-based localization:

- **Coverage and BER Analysis:** Future work should integrate bit error rate (BER) simulations based on the CDF coverage results, with moving receivers to evaluate Doppler effects and dynamic coverage variations.
- **Enhanced Propagation Modeling:** Implementing diffraction and scattering mechanisms in the PyVista pipeline would enable more realistic multipath modeling and support improved ray selection for localization, especially in non-line-of-sight scenarios.
- **Machine Learning Approaches:** The simulation outputs generated in this thesis could serve as a labeled dataset for supervised machine learning models, potentially enabling robust, data-driven classification and selection of high-confidence rays in practical deployments.
- **Multi-Anchor, Synchronization, and Temporal Filtering:** While this study focused on a single-base-station scenario to avoid the added complexity of synchronization and channel state estimation, extending the framework to support multiple anchors could reduce localization ambiguity and improve accuracy, especially in the context of future distributed massive-MIMO systems for 6G. Additionally, incorporating time-evolution techniques, such as Kalman filtering, would allow the system to leverage temporal correlations in both position and path observations, further mitigating errors due to incorrect ray selection and enhancing overall localization robustness.

In summary, DT is shown to be a promising platform for wireless localization, their effectiveness is fundamentally dependent on the accuracy of their spatial representations and the sophistication of the underlying propagation and localization algorithms. Continued research and development in these directions will be crucial to realize the full potential of DT-based positioning systems in next-generation wireless networks.

Bibliography

[1] S. Mihai et al., "Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2255–2291, Fourthquarter 2022. doi: 10.1109/COMST.2022.3208773.

[2] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, Jan. 2022. doi: 10.1109/MCOM.001.21143.

[3] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020. doi: 10.1109/ACCESS.2020.2998358.

[4] C. Ding and I. W.-H. Ho, "Digital-Twin-Enabled City-Model-Aware Deep Learning for Dynamic Channel Estimation in Urban Vehicular Environments," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1604–1612, Sept. 2022. doi: 10.1109/TGCN.2022.3173414.

[5] Z. Yun and M. F. Iskander, "Ray Tracing for Radio Propagation Modeling: Principles and Applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015. doi: 10.1109/ACCESS.2015.2453991.

[6] Z. Yun, Z. Zhang, and M. F. Iskander, "A Ray-Tracing Method Based on the Triangular Grid Approach and Application to Propagation Prediction in Urban Environments," *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 5, pp. 750–758, May 2002. doi: 10.1109/TAP.2002.1011243.

[7] ITU, "Recommendation ITU-R P.526-15: Propagation by Diffraction," International Telecommunication Union, Oct. 2019.

[8] Sionna, "main_page" `https://jhoydis.github.io/sionna-0.19.2-doc/api/rt.html`, [Online; accessed 2025-05-30].

[9] J. Hoydis, S. Cammerer, F. Ait Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, "Sionna: An Open-Source Library for Next-Generation Physical Layer Research," *arXiv preprint arXiv:2203.11854*, 2023. [Online]. Available: https://arxiv.org/abs/2203.11854.

[10] J. Hoydis et al., "Learning Radio Environments by Differentiable Ray Tracing," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 1527–1539, 2024. doi: 10.1109/TMLCN.2024.3474639.

[11] J. Hoydis et al., "Sionna RT: Differentiable Ray Tracing for Radio Propagation Modeling," in *2023 IEEE Globecom Workshops (GC Wkshps)*, Kuala Lumpur, Malaysia, 2023, pp. 317–321. doi: 10.1109/GCWkshps58843.2023.10465179.

[12] Sionna, "compute_paths" https://jhoydis.github.io/sionna-0.19.2-doc/api/rt.html#compute-paths, [Online; accessed 2025-05-15].

[13] Sionna, "coverage_map" https://jhoydis.github.io/sionna-0.19.2-doc/api/rt.html#coverage-map, [Online; accessed 2025-05-15].

[14] Sionna, "render_to_file" https://jhoydis.github.io/sionna-0.19.2-doc/api/rt.html#render-to-file, [Online; accessed 2025-05-15].

[15] Github, "sionna_osm_scene" https://github.com/manoj-kumar-joshi/sionna_osm_scene, [Online; accessed 2025-05-15].

[16] ITU Radiocommunication Sector, "Effects of Building Materials and Structures on Radiowave Propagation Above about 100 MHz," Recommendation ITU-R P.2040-2, 2024.

[17] D. A. McNamara, C. W. I. Pistorius, and J. A. G. Malherbe, *Introduction to the Uniform Geometrical Theory of Diffraction*, Artech House, Boston, 1990.

[18] R. G. Kouyoumjian and P. H. Pathak, "A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface," *Proceedings of the IEEE*, vol. 62, no. 11, pp. 1448–1461, 2005.

[19] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, "Measurement and Modelling of Scattering from Buildings," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 1, pp. 143–153, 2007.

[20] M. Schmidhammer, B. Siebler, C. Gentner, and S. Sand, "Reflection Point Localization Without Prior Environmental Knowledge," in *2025 19th European Conference on Antennas and Propagation (EuCAP)*, Stockholm, Sweden, 2025, pp. 1–5. doi: 10.23919/EuCAP63536.2025.11000020.

[21] Sullivan, C. and Kaszynski, A. et al., "PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)," *Journal of Open Source Software*, vol. 4, no. 37, p. 1450, 2019. doi: 10.21105/joss.01450.

[22] J. Morais and A. Alkhateeb, "Localization in Digital Twin MIMO Networks: A Case for Massive Fingerprinting," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, Denver, CO, USA, 2024, pp. 276–281. doi: 10.1109/ICCWorkshops59551.2024.10615652.

[23] M. Z. Karakusak, H. Kivrak, S. Watson, and M. K. Ozdemir, "Cyber-WISE: A Cyber-Physical Deep Wireless Indoor Positioning System and Digital Twin Approach," *Sensors*, vol. 23, no. 24, p. 9903, 2023. doi: 10.3390/s23249903.

[24] M. Zhu, L. Cazzella, F. Linsalata, M. Magarini, M. Matteucci, and U. Spagnolini, "Toward Real-Time Digital Twins of EM Environments: Computational Benchmark for Ray Launching Software," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 6291–6302, 2024. doi: 10.1109/OJCOMS.2024.3463963.

[25] L. Zhou, S. Zhou, and V. Yao, "Joint Maximum-Likelihood and MMSE Channel Estimator for UWB Communications," *Tsinghua Science and Technology*, vol. 11, no. 5, pp. 568–573, Oct. 2006. doi: 10.1016/S1007-0214(06)70235-8.

[26] L. Qinghua, L. Sicheng, Y. Yipeng, J. Chunyu, and Y. Xiaozhen, "An Accurate Maximum Likelihood Location Method Based on UWB Platform," in *2019 14th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, Changsha, China, 2019, pp. 515–521. doi: 10.1109/ICEMI46757.2019.9101442.

[27] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-Based Wireless Location: Challenges Faced in Developing Techniques for Accurate Wireless Location Information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, July 2005. doi: 10.1109/MSP.2005.1458275.

A

Appendix 1

Genie Aided result segregated by map type for all rays

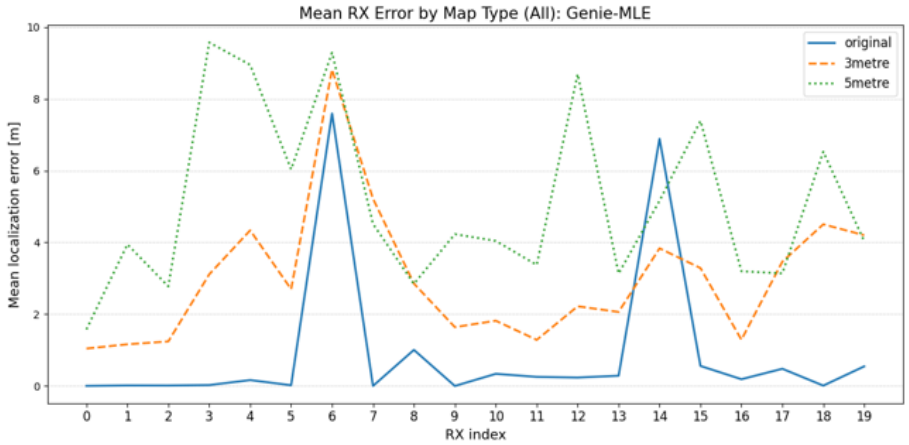

Figure A.1: Sionna simulation result of CDF overlay group by type

Genie Aided result segregated by map type for type1 rays only

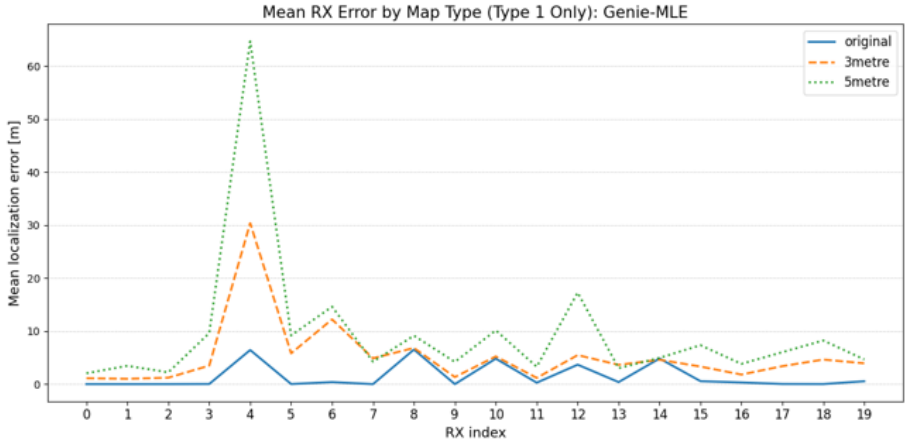

Figure A.2: Sionna simulation result of CDF overlay group by type

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

CHALMERS
UNIVERSITY OF TECHNOLOGY