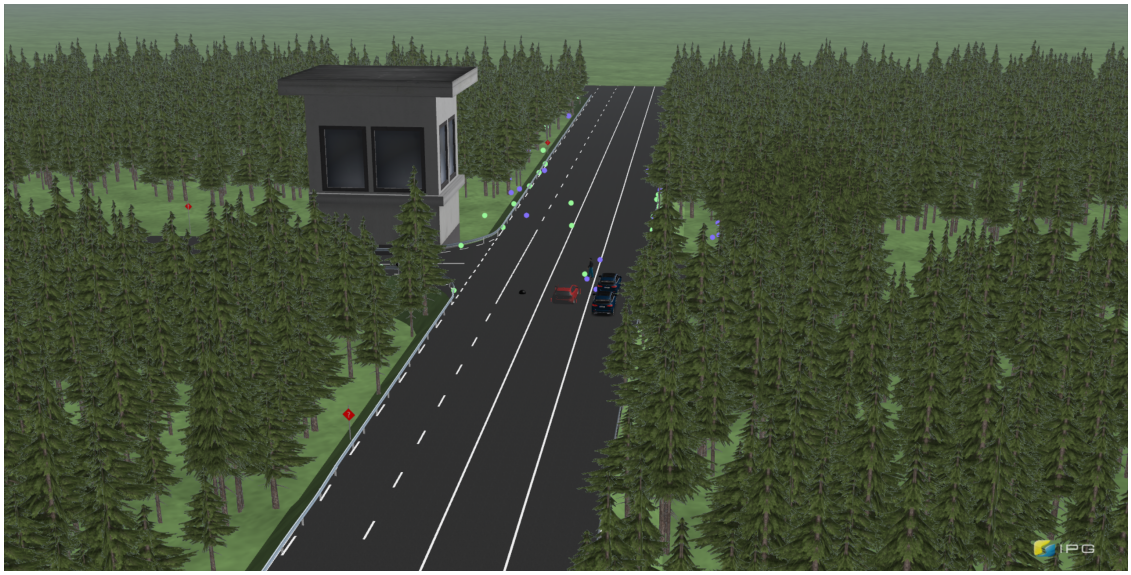
CHALMERS
UNIVERSITY OF TECHNOLOGY



Digital Twin-Based Simulated Automotive Radar for Virtual Testing

Master's thesis in Electrical Engineering

SILAN KARADAG

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Digital Twin-Based Simulated Automotive Radar for Virtual Testing

SILAN KARADAG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Communication Systems
Chalmers University of Technology
Gothenburg, Sweden 2024

Digital Twin-Based Simulated Automotive Radar for Virtual Testing
A Step Towards High-Fidelity Virtual Testing
SILAN KARADAG

Supervisor: Devrat Singh, APTIV
Co-supervisor: Musa Furkan Keskin, Department of Electrical Engineering
Examiner: Henk Wymeersch, Department of Electrical Engineering

Master's Thesis 2024
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Screenshot from CarMaker showing simulation environment (Vehicle Model from [1]).

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Digital Twin-Based Simulated Automotive Radar for Virtual Testing
A Step Towards High-Fidelity Virtual Testing
SILAN KARADAG
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Modern vehicles are increasingly characterized by software-based systems, and perception sensing technology plays an important part in the safety and the driving experience. The development of a reliable and efficient methodology for perception technology in a virtual environment holds the potential to accelerate the development of autonomous vehicles while ensuring safety and reliability. Between these perception systems, the development and validation of radar systems for autonomous driving is a major challenge due to the complex and dynamic nature of real-world driving scenarios. Within the scope of this thesis, a digital twin-based simulated environment for automotive radar perception analysis has been implemented. This work presents a method for the parameterization and validation of the digital twin of the selected test scenarios defined by Euro NCAP. The results are analyzed to evaluate the simulation-to-reality gap, indicating how closely the simulated data matches real-world observations. Additionally, the effects of sensor fidelity are examined to understand how accurately the sensor models replicate real-world performance. Lastly, a sensitivity analysis is conducted to identify critical elements. The outcome of the thesis indicates that virtual testing is a promising approach for radar sensor model analysis based on the idea of digital twin, especially using high-fidelity twins.

Keywords: Digital Twin, Automotive Radar, Autonomous Driving, Vehicle Simulation, Virtual Testing, IPG CarMaker.

Acknowledgements

As the final pages of my thesis turned, I found myself thinking of the successes and struggles I had in my academic journey; maybe, this is just the starting point.

Firstly, I would like to express my deepest gratitude to all my teachers, starting with the founder of the Turkish Republic, Mustafa Kemal Atatürk. His leadership has always illuminated my path and motivated me despite challenges. Special thanks to Taylan Memduh Ergörün and Alper Bereketli, PhD for believing in me when I was much younger and only dreaming of becoming an engineer.

I extend my gratitude to Björn Wennberg for trusting me to start my journey at such a great workplace and giving me the opportunity to perform my thesis at APTIV. I would like to thank to my supervisors, Devrat Singh and Musa Furkan Keskin, for their support, feedback, and the tools they provided through this work, as well as the IPG Team Sweden.

To my family, my mom, Selen, and my dad, Yener, your support has always been the cornerstone of my accomplishments, and I will always be grateful for always prioritizing my and my sister's education. My wonderful sister, Beril, I am so lucky to have you in my life; you are my endless source of joy and laughter.

Silan Karadag, Gothenburg, June 2024

List of Acronyms

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

AEB	Autonomous Emergency Braking
CPNA	Car-to-Pedestrian Nearside Adult
CPNC	Car-to-Pedestrian Nearside Child
EKF	Extended Kalman Filter
EURO NCAP	European New Car Assessment Programme
FIT	Finite Integration Technique
FL	Front Left
FOV	Field of View
FDTD	Finite Difference Time Domain
FR	Front Right
GT	Ground Truth
HiFi	High Fidelity
HiL	Hardware-in-the-Loop
IFF	Input From File
KPI	Key Performance Indicator
LiDAR	Light Detection and Ranging
LRR	Long-Range Radar
MAE	Mean Absolute Error
MRR	Mid-Range Radar
RADAR	Radio Detection and Ranging
RCS	Radar Cross Section
RMSE	Root Mean Square Error
RSI	Raw Signal Interface
RSDS	Raw Signal Data Stream
RT-Range	Real Time-Range
RTS	Rauch–Tung–Striebel
SNR	Signal-to-Noise Ratio
SRR	Short-Range Radar
VCS	Vehicle Coordinate System
VRU	Vulnerable Road User
VUT	Vehicle Under Test

Contents

List of Figures

List of Tables

1

Introduction

Modern vehicles are more and more characterized by software-based systems, the perception sensing technology being the one which is playing an important part in the safety and the driving experience. This advancement is heavily reliant on data, which shows the significance of the data in the development and verification of these systems. Collecting big amounts of driving data can be relatively straightforward, however obtaining high-quality data poses a significant challenge. In addressing this challenge, the concept of a digital twin and virtual testing emerges as a solution.

Within the scope of this thesis, digital twin-based simulated automotive radar has been implemented including all components existing in the physical scenario, presenting a method for the parametrization and validation of the digital twin of the selected test scenarios that is defined by Euro NCAP and referred as 'Car-to-Pedestrian Nearside Adult (CPNA)' and 'Car-to-Pedestrian Nearside Child (CPNC)' [2].

1.1 Motivation

The European Union (EU) has established the tight goals to improve road safety and cut the number of deaths. As per the EU Road Safety Policy Framework 2021-2030 the aim is to cut the number of road deaths by half by 2030. In particular, as a first step toward "Vision Zero," which calls for zero fatalities and serious injuries by 2050, the EU hopes to significantly reduce the number of fatalities and serious injuries on European roads [3]. One of the key benefits of using digital twin concept is that allows to identify the possible risks on the roads and lets to build systems accordingly.

1.2 Aim

Radar measurements are sparser and more stochastic in comparison to data from other common perception sensors used such as camera and LiDAR, making it challenging to interpret visually. Consequently, the evaluation of radar data is highly reliant on the assessor's expertise, resulting in subjectivity and the risk of inaccuracies. Current evaluation methods are often qualitative and not easily scalable [4].

Moreover, the current literature has evaluations on low fidelity sensor data, however, high fidelity sensor data has not been studied yet. This need is addressed by proposing a methodology for validation of simulated radar data in the context of virtual testing. The development of a reliable and efficient methodology for radar validation in a virtual environment holds the potential to accelerate the development of autonomous vehicles while ensuring safety and reliability. Following research questions will be answered within the scope of selected Euro NCAP test scenario in order to assist in answering the main question 'To what extent can simulation and virtual testing replace real-world testing?':

- Which properties are significant to represent simulations accurately to the reality?
- What should be the steps for creating the methodology for the quantification of the simulation-to-reality gap?
- How does the simulation-to-reality gap varies among the different test runs?
- How does the fidelity of the sensor model affect the execution efficiently representing the necessary degree of fidelity?
- Which key performance indicators are critical for the credibility assessment of simulations?
- Which parameters have a strong influence on the performance of simulations?
- How do these parameters influence the simulation results?

To answer these questions, this thesis integrates multiple domains such as physics, radar signal processing, simulation dynamics, and autonomous vehicle technology.

1.3 Objectives

The objectives of the thesis focuses on credibility assessment of simulations, primarily through implicit validation of sensor models. By developing a step by step methodology, it is established comparability in order to assess the experiments and provide insights about the research questions mentioned above.

This study also covers the impact of sensor model fidelity in the context of standardized Euro NCAP testing. By exploiting the availability of sensors with different fidelity levels in simulation, the accuracy of object properties derived from each sensor is compared. For this purpose, CarMaker built-in sensors will be used by adjusting its parameters reflecting the physical radar parameters. Two radar sensor models are implemented on the virtual environment:

- High Fidelity Radar Sensor: The environment is perceived by a reduced propagation [5].

- Raw Signal Interface (RSI) Radar Sensor: Improved radar sensor model which is based on the physical propagation [5].

The results are expected to be used as input to downstream AEB functionality in order to help identify the necessary degree of fidelity needed for testing. Further contributions of the thesis include the integration and correct parametrization of the radar sensor models, accurately fusing the information from multiple radars and the developed algorithm for tracking objects using detection points, the sensitivity analysis approach to identify critical sensor effects. These contributions aim to advance the state of the art in sensor model validation for virtual testing.

The Key Performance Indicator (KPI) is defined as the positioning error of the target, analyzed in terms of the lateral and longitudinal movements of the target.

1.4 Limitations

Validating radar simulations poses unique challenges due to the complex and dynamic nature of real-world driving scenarios. Although the simulation data is raw data, the real radar data has been through some processing. Low-level signal processing of radar data is considered a black box system; evaluations will be performed on processed data. The processing method might differ between the different sources of data collection.

During the simulations, it has been observed that not having a complete digital model of the vehicle used for real-world tests is one of the most important limitations, as the vehicle dynamics determine the behavior of the car even though input data is given for the vehicle maneuvers.

Another critical factor is the limitations coming from the modeling of the radar itself, which means the modeling of radar signal propagation, reflection, and interference within the simulation environment in replicating the real world.

1.5 Project partners

The thesis is being conducted within the problem context of the EVIDENT (Enabling Virtual Validation and Verification for ADAS and AD Features) research project [6]. The aim of this project is to measure the level of how closely the test environment mimics real traffic scenarios in order to enhance or substitute data from actual recordings.

1.5.1 APTIV

The authors are representatives of project partner company APTIV, a provider of vehicle components including electrical, electronic, and safety technology. The contributions within the project includes hardware and software components for performing real and virtual testing. The company seeks to utilize from simulations and virtual testing, anticipating benefits in technological advancement and cost efficiency.

1.5.2 AstaZero

Another key enabler from EVIDENT is AstaZero, which is a test environment that is built for testing purposes to see how cars and other vehicles behave for automated transport systems by offering high fidelity digital twins of the test environment [7]. For our data collection, the APTIV vehicle is tested in this environment. Some additional resources from AstaZero are available as well, such as the environment model used for virtual tests.

1.6 Literature Review

The importance of virtual development and testing within the future of automotive industry is increasing day by day. Several studies have been carried out related to the scope of the thesis which have proposed different methods and techniques. This section provides a background by giving an overview of the existing literature related to the scope of this thesis.

The digital transformation in the automotive industry started in recent years, driven by the continuous shift towards electromobility and the availability of high-performance microelectronics [8]. This digitization is again confirmed by the fact that "almost every real situation will sooner or later be modeled" [9], emphasizing the importance behind simulation efforts. Moreover, the inconsistency of test scenarios usually leads to the inability to reproduce results due to the many variables that come into play at the same time such as other road users and weather conditions. In particular, the testing of critical driving situations, where the vehicle is still controlled by a human driver, involves a certain degree of risk for the test driver. This need drives simulation efforts, which stand to decrease resources needed to gather data that is essential for further development.

Various simulators have been developed for virtual test driving having their own pros and cons. Car Learning to Act (CARLA), LG Silicon Valley Lab (LGSVL), Aerial informatics and Robotics Simulation (AIRSIM) are some examples for open-source softwares [10]. IPG CarMaker, Simcenter Prescan, NVIDIA DRIVE Sim, CarSim are some examples for commercial simulators. MATLAB also still plays a key role by offering Automated Driving Toolbox. Between these, it has been found that CARLA and IPG CarMaker has the most use in literature.

A major problem for the development and verification of the radar systems for the autonomous driving is the necessity of the testing in various environmental conditions and scenarios. Of course, real-world testing is expensive, takes a lot of time, and has a limited capacity to recreate the rare and possibly dangerous situations. Besides, the ethical and regulatory issues frequently occur when conducting tests of the autonomous functions (such as autonomous emergency braking) and vehicles on the public roads. Consequently, there is a growing demand for virtual testing methodologies that can comprehensively validate radar sensors in a controlled and repeatable environment. This demand introduces the theme of this thesis; how do we

perform sufficient verification and validation of a virtual test run, the environment and how do we quantify uncertainties related to this.

According to literature, the digital twin-based simulation of automotive radar has the potential to reduce the time and cost of testing automotive radar systems, but there are some limitations to this approach. The main limitation of the simulation models is their accuracy. Moreover, the scalability of the approach is restricted by the computational needs of the simulation models, which are growing with the complexity of the system.

1.7 Standards and Regulations

1.7.1 Euro NCAP

Euro NCAP stands for the European New Car Assessment Programme. Euro NCAP has been founded in 1997 and since then it is playing a key role in advancing vehicle safety regulations and promoting the use of safety technology in the industry. Euro NCAP has been supporting the continuous improvement of vehicle safety standards around the world by introducing test procedures and assessment criterias to address safety challenges.

Different questions to be answered within the EVIDENT project have been subjected to various scenarios predefined by Euro NCAP. The test case for this project is selected to match a subset of those cases chosen in the EVIDENT project, more specifically, the case realized by APTIV, which is a Euro NCAP test scenario [2]. The test scenario that is selected for this thesis is referred to as 'Car-to-Pedestrian Nearside Adult (CPNA)' and 'Car-to-Pedestrian Nearside Child (CPNC)' as can be seen in Figure 1.1 [2]. It involves a situation where a vehicle moves forward, approaching an adult pedestrian who is walking from the nearside, and is struck by the front part of the vehicle at 50% impact location of the vehicle's width. Variations of the test scenario are also available to ensure the reliability of results. The selected test cases are presented in Table 1.1.

1.7.2 ASAM OpenX

ASAM stands for Association for Standardisation of Automation and Measuring Systems. This open standard provides standards for simulation model data exchange. It defines the file formats for the exchange of data within the automotive industry, particularly in the context of vehicle testing and validation.

1.7.2.1 ASAM OpenDrive

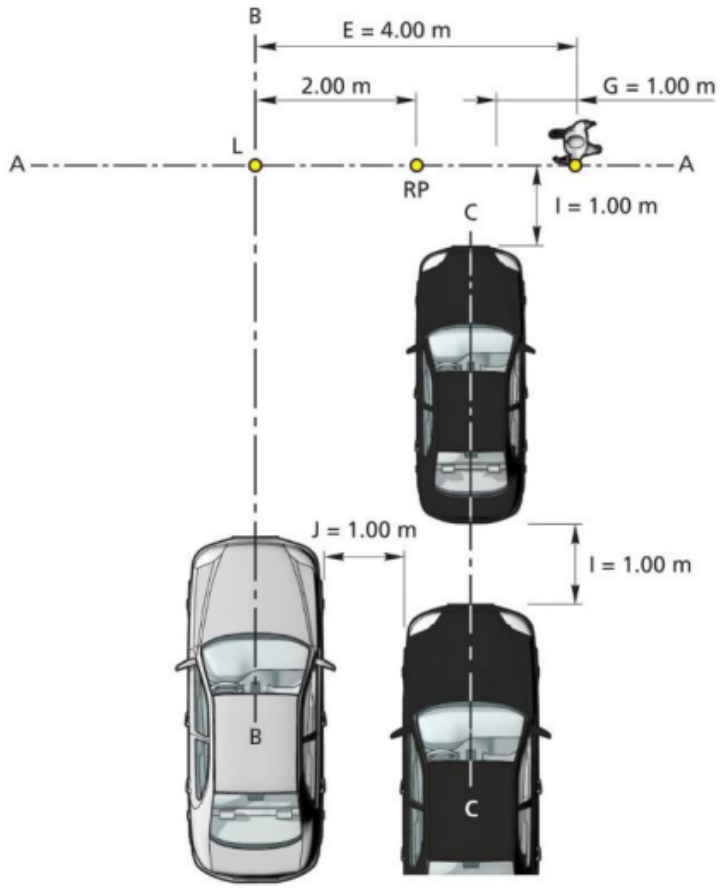
OpenDRIVE stands for Open Dynamic Road Information for Vehicle Environment. OpenDrive files define a static environment, represented by LiDAR-based scans of the testing area in AstaZero proving ground as a digital twin of the environment, including road geometry, guard rails, poles, lane markings, etc.

1.7.2.2 ASAM OpenScenario

OpenScenario defines actions of all dynamic objects in the scenario. Description elements include vehicle's maneuvers and trajectory, details about the vehicle, driver model as well as environment parameters. An OpenScenario file is available for the specific scenario, however could not be imported to IPG CarMaker environment. Therefore the scenario is created from scratch, Section III, Methodology explains this process.

Table 1.1: Test Cases.

Case Number	Host Speed	Pedestrian Model	Collision Point
1	15 km/h	Adult	50%
2	15 km/h	Child	50%
3	30 km/h	Adult	50%
4	30 km/h	Child	50%



where:

- AA - Trajectory of Pedestrian Dummy
- BB - Centerline of Vehicle Under Test (VUT)
- CC - Centerline of Obstruction Vehicles
- G - Acceleration Distance of Pedestrian Dummy
- I - Distance of Pedestrian Dummy to Front of Obstruction Vehicle
- J - Distance Between VUT and Obstruction Vehicle
- L - Impact position
- RP - Reference Point

Figure 1.1: CPNA Scenario – Occluded Child from Nearside Obstruction Vehicles [2].

2

Theory

This chapter provides the theoretical background on digital twins, radars, and, in particular, automotive radars, modeling processes, and object tracking algorithms.

2.1 Digital Twin

As the thesis title indicates, a digital twin of the defined traffic scenario is produced. This section provides an overview of the concept of Digital Twin.

The idea of the term digital twin was first introduced by NASA and Air Force researchers in 2012 and stemmed from the realization that traditional design approaches were insufficient in addressing new demands [11]. A solution for that is the digital twin, which is a virtual replica of a real-world entity. The digital twins typically consist of three main components: the physical model, the corresponding virtual model, and the connection enabling comparison between them [12] [13] [14]. Physical model refers to the object, system or process used in reality to be modeled. The virtual representation refers to its digital version, reflecting its physical characteristics.

Using digital twins makes it possible to carry out simulations of various situations and perform an experiment virtually. This leads to improved decision-making and identifying potential hazards that can occur before the systems are used in real time. It also facilitates real-time system monitoring, enabling continuous observation of the system performance as it happens. Besides that, it provides remote operation capabilities, allowing control of systems from remote locations.

Due to these opportunities, the digital twin concept has applications across various industries. For example, digital twins are used in manufacturing to optimize production processes, predict equipment failures, and improve product quality. Virtual replicas of human organs are developed in healthcare to understand diseases better and help patients. In aerospace, digital twins enable aircraft design and testing. Within the automotive industry, it allows exploring many driving functions to enhance driver safety and experience.

The increase in the usage of digital twins comes with significant benefits, including improved efficiency, cost savings, and enhanced safety. Moreover, it supports innovation by enabling rapid prototyping, iterative design improvements, and data-driven decision-making. However, this process is not always perfect, and determining how similar the twins are to each other remains a challenge [15]. This is commonly referred to as the 'fidelity' of the digital twin and relates to how well the digital twin mimics the actual behavior of the physical component. Depending on the accuracy of this behavior of the twins, digital twins can be classified according to different levels of fidelity. These are low fidelity, medium fidelity, and high fidelity. As with almost every engineering problem, it comes with a trade-off. Low-fidelity replicas are usually simpler models and time and cost effective solutions; however, these models typically lack reality. In contrast, high-fidelity replicas are more realistic in representing the physical component; however, these are usually more complex models, requiring more sources to run these models.

Creating digital twins within the thesis scope includes the elements below for the components used in the selected scenario. Section III, Methods, explains each component.

- Vehicle
- Driver
- Pedestrian
- Road
- Data Processing
- Sensors

2.2 Automotive Radar

The equipment of the vehicle is shown in Figure 2.9. As illustrated, the vehicle has two radars mounted on the left and right sides of the front bumper. The thesis is more concerned with analyzing the output of these radars rather than developing them. However, it is essential to understand the working principle of the radars in order to interpret the sensor output.

Radar stands for Radio Detection and Ranging. Radar sensors transmit radio waves and receive their reflections [16]. Radar systems can measure the range, velocity, and angle of detected objects by analyzing the time delay between the transmitted signal and the echo.

The one-way distance between the radar and the target can be calculated as in Equation 2.1. However, this equation alone does not fully represent the radar equation. In a broader perspective, the radar equation is given in Equation 2.2, which is a fundamental equation in radar systems.

$$R = \frac{c \cdot \Delta t}{2} \tag{2.1}$$

$$P_r = \frac{P_t G \sigma}{(4\pi) R^2} \tag{2.2}$$

In these equations, R is the distance from radar to target, c is the speed of light, Δt is the round-trip time, P_r is the reflected power from the target, P_t is the transmitted power, G is the gain of the antenna, σ is the radar cross-section of the target. Figure 2.1 shows a simple illustration of how the range can be measured by using transmitted and received waves.

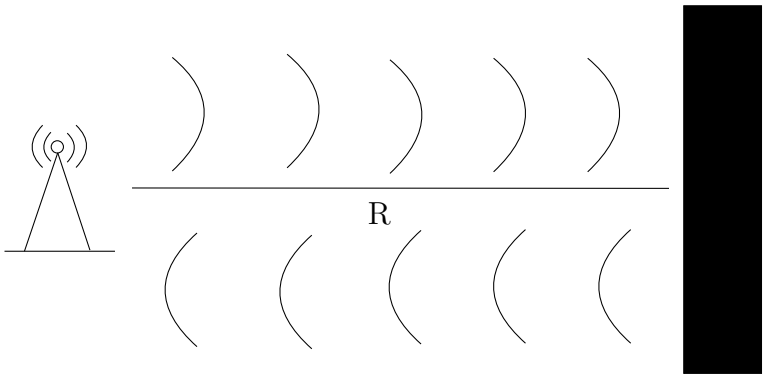


Figure 2.1: The transmitted and received electromagnetic waves from the radar.

Automotive radars can be classified depending on the maximum range; these are the Long-Range RADAR (LRR), Mid-Range RADAR (MRR), and Short-Range RADAR (SRR). Figure 2.2 shows how they can be classified and their typical range.

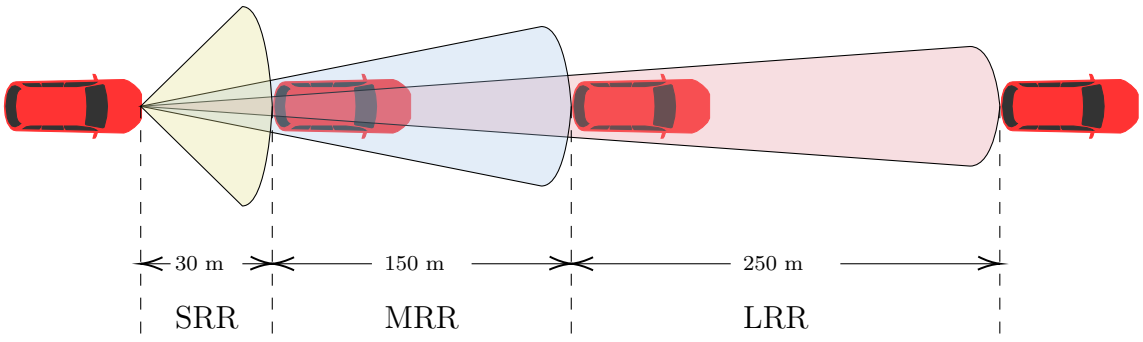


Figure 2.2: Classification of Automotive Radar, Typical Range and Field of View

Radar sensor models can be developed by applying different methods: physics-based models, data-driven models, and a combination of the two, known as hybrid models.

Physics-based sensors solve Maxwell's equations by using methods like the finite-difference time-domain (FDTD) and finite integration technique (FIT). However,

solving these equations is computationally heavy. An alternative approach to that is using ray-tracing methods [17]. Ray-tracing approximates electromagnetic wave propagation using geometrical optics with multiple rays to find alternative paths between the antenna and the target [18]. Even if this method is less accurate than using FDTD and FIT, it helps to facilitate simulations [18].

Data-driven models use empirical data to learn the connection between input and output variables, in contrast to physics-based models. Machine learning methods are often used for radar sensor data-driven model development [19]. Because there is no need to solve complex physics-based equations, data-driven models are computationally less intensive than physical models. However, these models are less accurate, which is a drawback.

Hybrid models combine both physics-based and data-driven approaches [20].

The sensor models that are used in the simulation are physics-based sensors. Between them, the working principle of the RSI Radar Sensor is ray-tracing, which produces detection points similar to physical radar. A detection is a reflection point that provides information about the range, angle, and amplitude of the object from which the detection originated. These objects are formed through a sequence of algorithms utilizing radar detections through a system known as tracker.

2.3 Object Tracking

Autonomous vehicles perceive their environment based on objects rather than detection points and take action based on that. Object tracking algorithms are used to achieve this, allowing the shape of objects using detection points. Due to this fact, the sensor output is compared on object-level data.

Although radar detections are less affected by the weather and atmospheric conditions than detections made by other sensors, such as camera-based detections, other sources may cause inaccuracies due to noise, like imperfections in the sensors and electromagnetic interference. This noise results in inaccurate object state estimations, making it challenging to detect objects reliably based only on raw radar measurements. These are the challenges requiring the use of estimation techniques to track objects accurately.

The output file format from CarMaker cannot be used as input for the real tracker, also called the APTIV radar tracker, because it requires a specific file format. The use of Carmaker output with the APTIV radar tracker was outside the constraints of the project. Hence, a tracker exhibiting similar behavior has been developed instead for object-level comparison. This section presents the developed tracker for this purpose.

The developed tracker consists of two main steps: the Extended Kalman Filter and Rauch-Tung-Striebel (RTS) Smoothing.

2.3.1 Kalman Filter

Kalman Filter is an estimator used to estimate the state of a dynamic system from a series of noisy measurements. It works in a recursive way by estimating the present state of a system by making predictions from previous estimated states and refining the estimates when more measurements are made. Traditional Kalman Filter performs optimally only under the assumption that both the system dynamics and measurement processes are linear Gaussian. Usually, in automotive applications, although the noise is Gaussian, the measurement process is not linear. The Extended Kalman Filter (EKF) is a modification of the Kalman Filter that can be applied to systems with nonlinear dynamics and/or measurements. In applications such as object tracking, since object motion and radar measurements are nonlinear in most cases, the EKF is applied to estimate the state.

The following text explains the steps of the implemented Extended Kalman Filter.

2.3.1.1 Extended Kalman Filter

Define System Dynamics:

The state information contains the position and velocity information of the pedestrian. Considering this, the state vector $\hat{x}$ is represented as:

$$\hat{x}_k = \begin{bmatrix} x_k & v_{x_k} & y_k & v_{y_k} \end{bmatrix}^\mathsf{T}$$

where:

- x, y: Longitudinal and lateral motion of the target
- v_y, v_y: Longitudinal and lateral velocity of the target
- k: Discrete time step of the measurements

The matrix A is referred to as the state transition matrix and describes how the system changes over time. Δt being the time interval between measurements, the state transition matrix A is given by:

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix H describes the mapping between the measurement and the state variable. Within the problem context of the thesis, we are interested in the estimates of position information x and y. The measurement matrix H is given by:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

A and H matrices define the system and are kept constant. The design parameters of the Kalman Filter are Q and R. These matrices are tuned through trial and error based on the outcome of the filter.

The process noise covariance matrix Q is given by:

$$Q = \begin{bmatrix} \sigma_{x_q}^2 & \sigma_{v_{x_q}}^2 & \sigma_{y_q}^2 & \sigma_{v_{y_q}}^2 \end{bmatrix}$$

where:

- $\sigma_{x_q}^2$, $\sigma_{y_q}^2$: Variance of the process noise in the longitudinal and lateral axis of the motion of the target.
- $\sigma_{v_{x_q}}^2$, $\sigma_{v_{y_q}}^2$: Variance of the process noise in the longitudinal and lateral axis of the velocity of the target.

The measurement noise covariance matrix R is given by:

$$R = \begin{bmatrix} \sigma_{x_\mathrm{r}}^2 & \sigma_{y_\mathrm{r}}^2 \end{bmatrix}$$

where:

- $\sigma_{x_\mathrm{r}}^2$: Variance of the measurement noise in the longitudinal axis of the motion of the target.
- $\sigma_{y_\mathrm{r}}^2$ is the variance of the measurement noise in the in the lateral axis of the motion of the target.

Initialization:

During the initialization step, the initial values of the state vector x and covariance matrix P are set, representing the estimate of the system state and its uncertainty, respectively.

The initial state vector $\hat{x}_0$ is defined when k=0 and represented as:

$$\hat{x}_0 = \begin{bmatrix} x_0 & v_{x_0} & y_0 & v_{y_0} \end{bmatrix}^\intercal$$

The initial covariance matrix P_0 is defined when k=0 and represented as:

$$P_0 = \begin{bmatrix} \sigma_{x_k}^2 & \sigma_{v_{x_k}}^2 & \sigma_{y_k}^2 & \sigma_{v_{y_k}}^2 \end{bmatrix}$$

Prediction Step:

In the prediction step, the EKF predicts the next state of the system based on the current state estimate and the system dynamics.

The linearization occurs when computing the Jacobian matrix of the motion model, which is used to linearize the nonlinear motion model around the current state estimate. This step allows the EKF to handle nonlinearities.

1. **State Prediction:**

   For Linear Kalman Filter: $\hat{x}'_{k+1} = A\hat{x}_k$

   Introducing the Linearization Function: $A\hat{x}_k \rightarrow f(\hat{x}_k) = \begin{bmatrix} x + \Delta t \cdot v_x \\ v_x \\ y + \Delta t \cdot v_y \\ v_y \end{bmatrix}$

   For Extended Kalman Filter: $\hat{x}'_{k+1} = f(\hat{x}_k)$

   where:

   - $\hat{x}'_k$ is the predicted state estimate.

2. **Covariance Prediction:**

$$P'_k = AP_{k-1}A^T + Q$$

Update Step:

In the update step, the EKF corrects the predicted state using new measurements obtained from sensors. For simplicity, there has been no linearization introduced at this step. This is often done in practice when the nonlinearities between the state and measurements are not significant.

1. **Measurement Update:**

$$z'_k = H\hat{x}_k$$

   - z'_k is the updated measurement.

2. **Kalman Gain Calculation:**

$$K_k = P'_k H^T (HP'_k H^T + R)^{-1}$$

   where:

   - K_k is the Kalman gain.

3. **Update State Estimate:**

$$\hat{x}_k = \hat{x}'_k + K_k(z_k - z'_k)$$

   where:

   - z_k is the current measurement set.

4. **Update Covariance:**

$$P_k = (I - K_k H_k)P'_k$$

   where:

   - I is the identity matrix.

The coefficients Q and R are critical in estimation, and their selection plays a major role in the results obtained.

- Decreasing Kalman Gain decreases the contribution of the measurements. On the other hand, the contribution of the prediction increases. In other words, the variation of the estimate decreases because the estimate is less affected by the measurement [21].

- Taking the given equation for evaluating the Kalman gain, the gain decreases as the value of R increases. R should be increased to reduce the influence of the measurements and obtain an estimate with less variation [21].

- Similar to the previous sentence, taking the given equation for evaluating the Kalman gain, the gain increases as the value of Q decreases. Q should be decreased to reduce the influence of the measurements and obtain an estimate with less variation. On the other hand, the prediction of the error covariance decreases [21].

2.3.1.2 Rauch-Tung-Striebel (RTS) Smoother

The developed tracker overall consists of two main steps. The first step is the forward pass to calculate the estimates of the states. The second step uses the results from the forward pass and operates backwards to smooth the data.

Initialization

The initial state vector $\hat{\hat{x}}_N$ is set as the last element of each estimated state.

$$\hat{\hat{x}}_N = \begin{bmatrix} \hat{x}_N & \hat{v}_{x_N} & \hat{y}_N & \hat{v}_{y_N} \end{bmatrix}^\intercal$$

where:

- N is the number of measurements. The subscript N corresponds to the last element from the estimates of the EKF output.

Smoothing

1. **RTS Gain Calculation**: Quantifies the correction needed to smooth the estimate.

   $$J_k = P_k A^\top (P_{k+1})^{-1}$$

   where:

   - J is the RTS gain.

2. **Update Smoothed State Estimate**:

   $$\hat{\hat{x}}_k = \hat{x}_k + J_k(\hat{x}_{k+1} - x'_{k+1})$$

3. **Update Smoothed Covariance:** Account for the refinement in the estimate.

$$P_k = P_k + J_k(P_{k+1} - P'_{k+1})J_k^\top$$

The Extended Kalman filter is followed by the Rauch-Tung-Striebel (RTS) algorithm, which smooths the raw estimates. This is done by going backward through the filtered states in order to improve the estimates using future data.

The RTS smoother leads to better estimations with reduced noise by taking future data into consideration. This, in turn, results in a smoother trajectory compared to the output generated directly from raw filtered estimates of the Extended Kalman filter.

For each case, the developed tracker is adjusted based on the APTIV radar tracker's performance, as each case has its challenges. Since the real tracker has more advanced characteristics, a perfect match is not expected; however, it should provide a comparable result. Figure 2.3, 2.4, 2.5 and 2.6 provide insight into how well the developed tracker behaves for each case, respectively, and relative to the vehicle coordinate system.

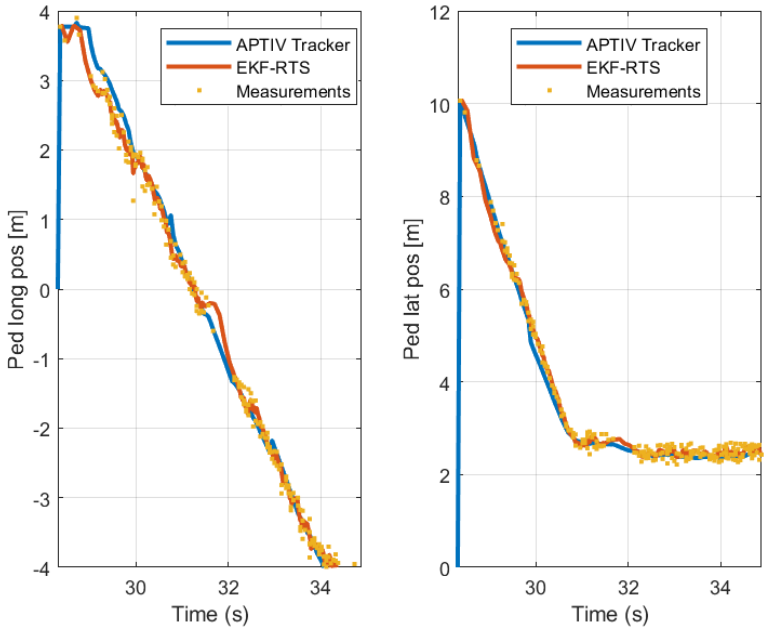


Figure 2.3: Comparison Between APTIV Tracker and the Developed Tracker shown with Detection Points for Case 1.

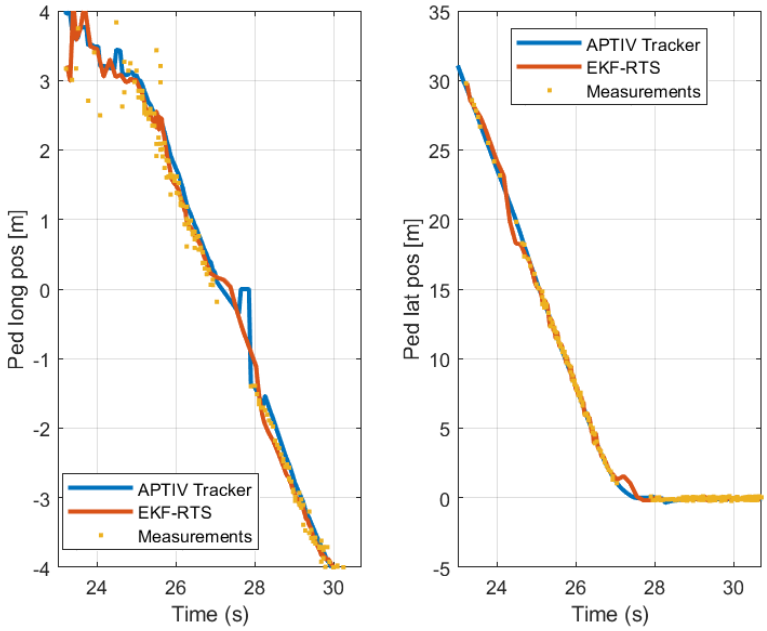


Figure 2.4: Comparison Between APTIV Tracker and the Developed Tracker shown with Detection Points for Case 2.

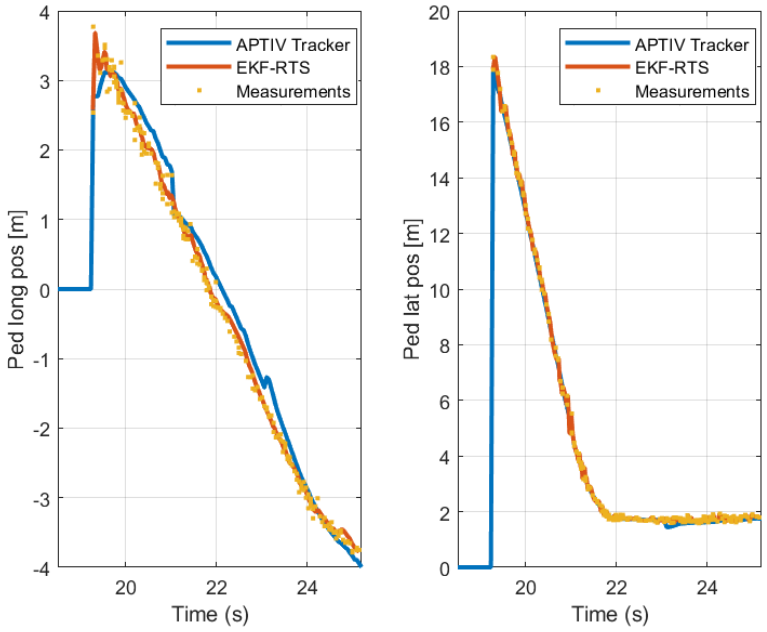


Figure 2.5: Comparison Between APTIV Tracker and the Developed Tracker shown with Detection Points for Case 3.

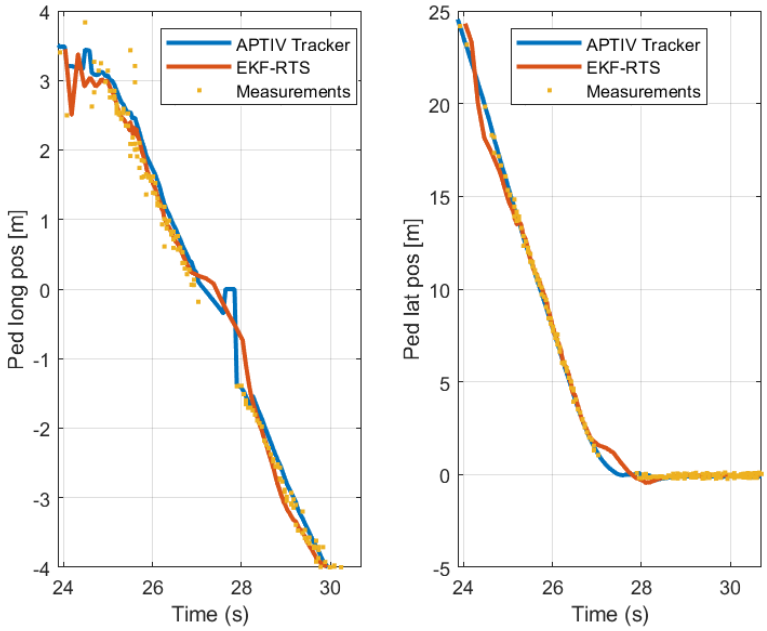


Figure 2.6: Comparison Between APTIV Tracker and the Developed Tracker shown with Detection Points for Case 4.

2.3.2 Data Smoothing

The data obtained during the simulations using the High Fidelity radar sensor has oscillations, as can be observed from the output data. Therefore, a smoothing process is required. This situation is addressed by the question, 'How should the data be filtered to improve the state estimates, and how does the resulting data differ from the original data?'. The answer to this question is to perform the Savitzky-Golay Filter on the output data.

This filter is chosen for properties such as preserving signal characteristics while minimizing distortions, hence allowing the handling of overlapping signals [22]. The filter operates on a moving window of data points. Least squares polynomial fitting is applied to the data within each window. This polynomial is evaluated within each window to get the smoothed output value. This process is repeated for each sample of the data [23].

The result of this step can be seen in Figure 2.7. The plots on the left represent the Ground Truth data and the raw output from the HiFi radar before applying the Savitzky-Golay filter. The plots on the right side represent the Ground Truth data and the filtered output after applying the Savitzky-Golay filter. As can be observed, the curves on the right are a much smoother and more accurate representation of ground truth and reality.

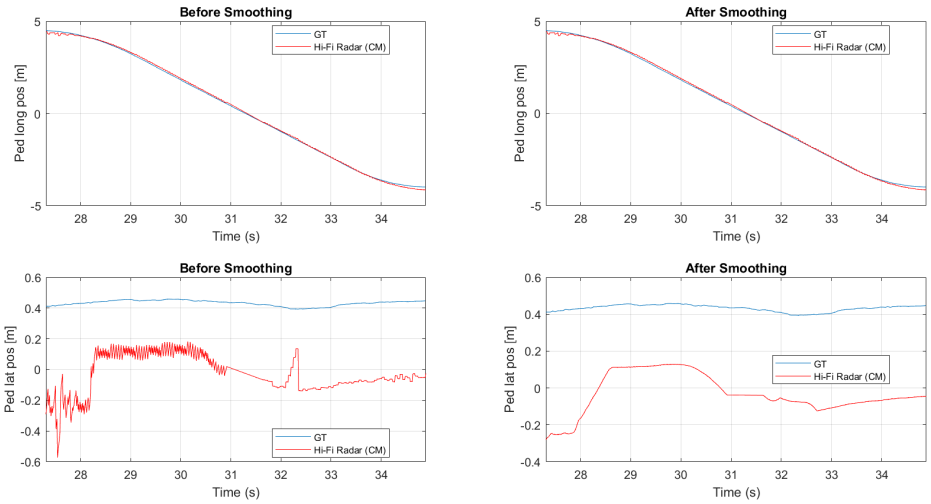


Figure 2.7: Savitzky-Golay Filtering Process on HiFi Radar Data.

2.4 Data Processing

The collected data consists of:

- Ground Truth Data: It is the output from INS equipment and consists of state information of the host vehicle and the target. The state information contains the position, velocity, and acceleration. The data is represented in world coordinates and the states are relative to a reference point.

- Experimental Data: It is the output from the test run and consists of state information of the host vehicle and APTIV MRR radars, which belong to early radar technology. The data is represented in the vehicle coordinate system (VCS), which is the center point of the front bumper.

- Simulation Data: The simulation run outputs state information of the host vehicle and radars. Radar sensor data is relative to the sensor frame as defined in [5].

As each data is represented in a different coordinate system, the first step is to convert them in a way to collect them all in the same coordinate system. For this purpose, the data is converted in a way to be represented the same way as Ground Truth Data, which is relative to a reference point. The reference point is marked with L in Figure 1.1. The following section explains how each data is processed for comparability.

2.4.1 Ground Truth Data

It has been observed that the reference point has been shifted by +0.25 cm along the lateral motion of the pedestrian during the experimental data collection. An

offset corresponding to the same amount has been applied along the same axis of the VRU to compensate for this.

2.4.2 Experimental Data

The state information traveled at each timestep relative to VCS is recorded for the experimental data. For this conversion, the total distance traveled at each data point is added to the state information and referenced to the start position so that the reference point becomes point zero as in GT coordinate system.

2.4.3 Simulation Data

The conversion has been applied in two steps. The first step deals with the conversion from the sensor frame to the VCS coordinate system. The sensors mounted on the vehicle are rotated for better coverage, which requires a rotation matrix.

The rotation operation is defined as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

where θ defines the angle of rotation, that is, the mounting angle, x and y represent the original coordinates of a point, and x' and y' represent the transformed coordinates after rotation.

After this conversion, the sensor position is added as offset to x' and y', so the resulting data becomes relative to VCS. As the last step, the same process described under experimental data conversion is applied.

Illustration of longitudinal and lateral movements of the vehicle is shown in Figure 2.8, which is important to understand the vehicle's and pedestrian's behaviour in the upcoming sections. Longitudinal movement refers to the forward motion of a vehicle along the length of the road. On the other hand, lateral movement refers to the side-to-side motion of the vehicle. For pedestrian movement, these definitions remain the same; however, the axes are exact opposites as the pedestrian is walking nearside the host vehicle.

2.5 Equipment and Tools

2.5.1 Equipment

The vehicle is equipped with several sensors to collect the data mentioned above. The equipment of the vehicle includes the following and can be seen in Figure 2.9:

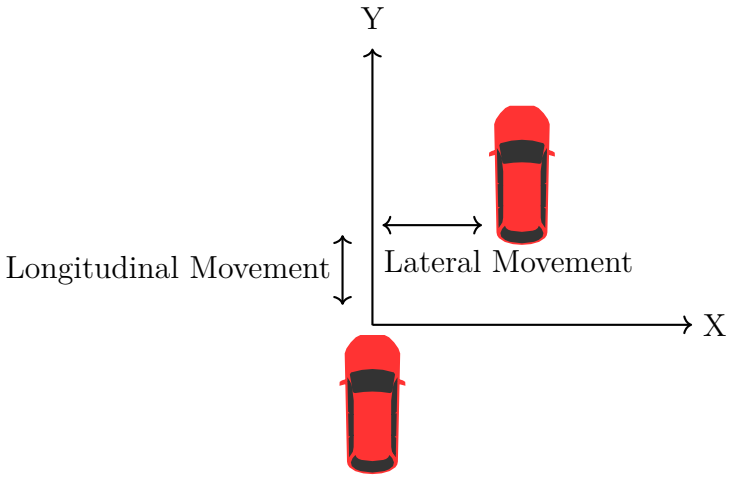


Figure 2.8: Illustration of Longitudinal and Lateral Movements of a Vehicle

- Vehicle: The vehicle refers to car used during the real data collection. A human driver controls the vehicle integrated with APTIV's AEB threat assessment and control. The vehicle is also called host and referred to as Vehicle Under Test (VUT).

- Pedestrian: The pedestrian model during the real test is controlled by a robot. The robot is pulling the pedestrian with a rope. The pedestrian is also referred to as Vulnerable Road User (VRU).

- GNSS/INS System: The vehicle and the pedestrian are equipped with AstaZero's RT-Range (Real Time-Range) system manufactured by OxTS [24], also known as Motionpack. This data is regarded as ground truth data and used as input data for CarMaker.

- Radar Sensor: The vehicle is equipped with two front-facing radars that are located on the front left and front right. APTIV MRR radars are used for real data collection. Low-level signal processing of radar data is considered a black box system; evaluations are performed on processed data.

- Camera Sensor: A camera with a Mobileye vision system is also available during real data collection. Vision equipment is used to collect reference data only.

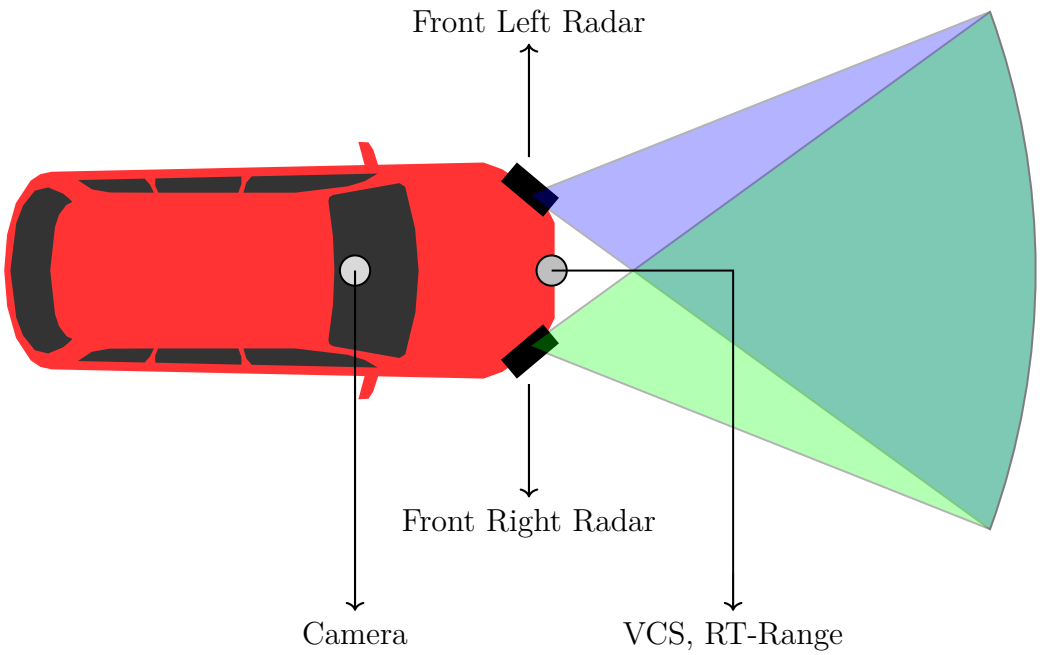


Figure 2.9: The Equipment of the Vehicle Under Test (VUT).

2.5.2 Tools

- CarMaker: CarMaker is used for performing simulations.
- MATLAB: MATLAB is used for analyzing the collected data and producing the results.
- Visualization Tool: APTIV internal tool within the company is used to visualize the collected real data.

An overview of the process and tools utilized is illustrated in Figure 2.10.

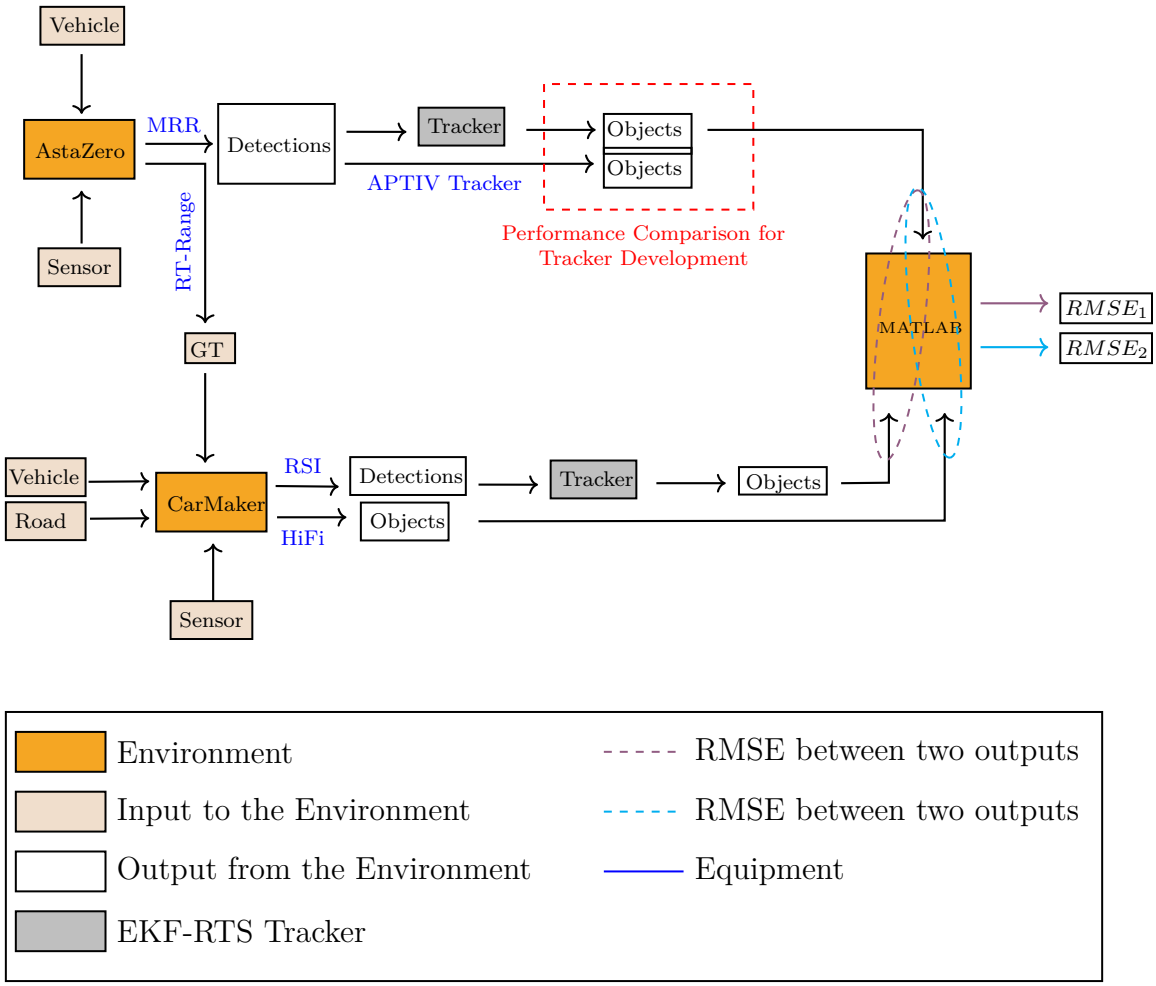


Figure 2.10: An Overview of the Process and Toolchain.

3

Methods

In this section, the steps of the performed work are described. The first section below describes how the simulation environment is created in the order they are executed throughout the project. However, it is important to mention that there is repetitive interaction among the sections to achieve the best research result. The following section explains how the data has been processed for comparability between real and simulation data.

3.1 Simulation Environment

The simulation environment needs to be modeled as realistically as possible to analyze the system's behavior. The primary components of the simulation environment are the car, traffic environment, traffic object, road, and the sensors involved in the project. It is crucial to parameterize each element correctly to represent them in the simulation environment. This section outlines the steps taken to create the digital twin of the test scenario in reality.

3.1.1 Vehicle

The simulation flow begins by specifying the host vehicle in the simulated domain. Selecting the appropriate model for the real vehicle becomes significant as it determines the overall dynamics of the specific car. Although several car models exist on CarMaker, the digital twin of the host vehicle used while conducting real tests, does not exist on CarMaker. Hence, the specified digital model of the car is selected from the available set of models and is parametrized appropriately to mimic the behaviour of the real car. The properties that should be set up within the vehicle include Car model, Maneuvers, and Driving Mode.

3.1.1.1 Maneuvers

To imitate the same scenario, CarMaker has the functionality to use real-world measurements by using an input file. This is done through the 'Input From File' window, which allows control of the simulation drive. The 'Input From File' window is shown in Figure 3.1. To best represent the real drive, the input data can contain

information about many vehicle properties, such as the speed profile, steering angle, and gear number.

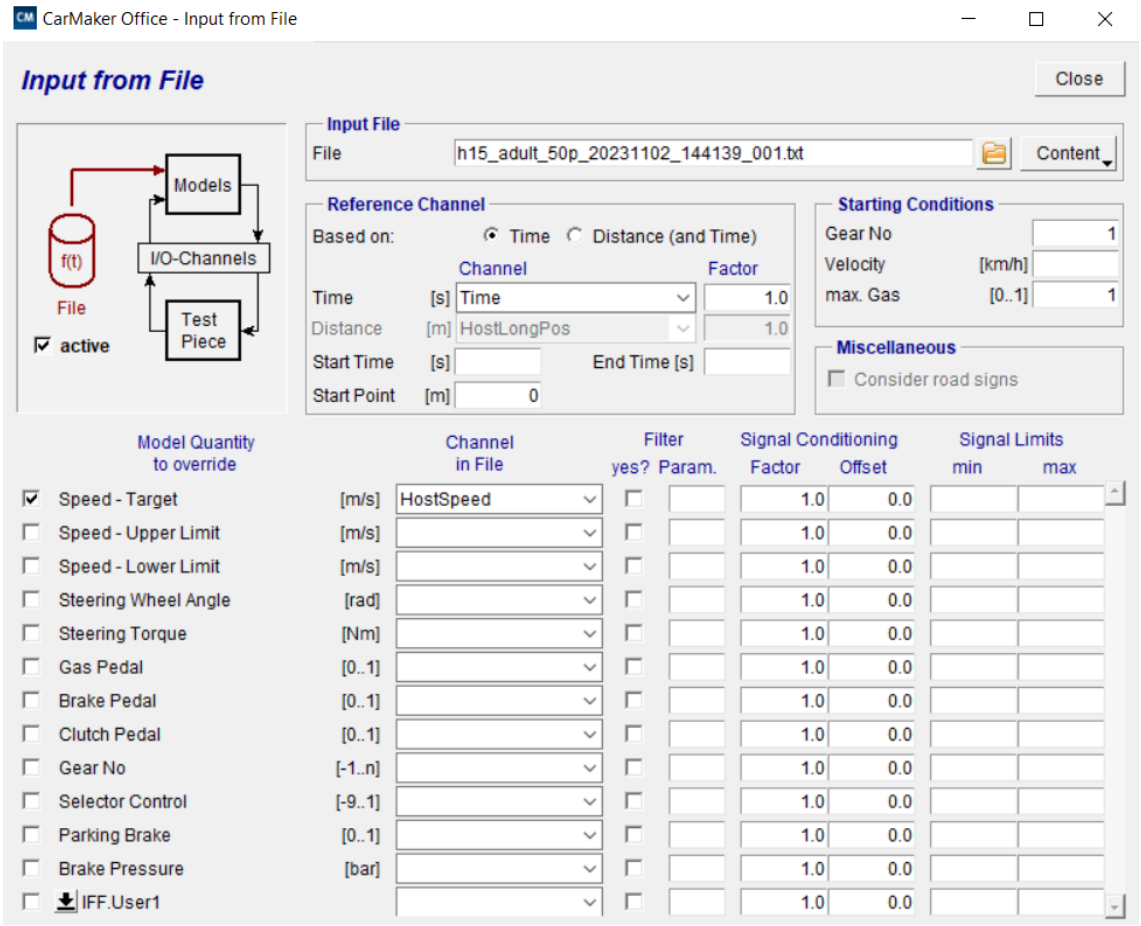


Figure 3.1: IPG CarMaker Input From File: Configuration of Input Data

The input data given encompasses the vehicle's time and longitudinal speed profile corresponding to a particular time step. This data is then assigned on the maneuver tab. It is set as the longitudinal speed profile, and a time limit is chosen for the stopping criteria, as shown in Figure 3.2.

Besides the longitudinal dynamics of the car, the lateral dynamics are also specified. It is possible to control the lateral dynamics via the steering wheel angle. However, this requires an accurate representation of the car dynamics. Due to this, the lateral movement is configured using the 'Edit Trajectory' option. This step is shown in Figure 3.3, where it is assigned a lateral position corresponding to the longitudinal position. The maximum number of entries is limited, so some selected data points are used.

3.1.1.2 Dynamics

The car's dynamics define the vehicle behavior in CarMaker and become a limiting factor while performing the drive. Once the car model is chosen and the simulation

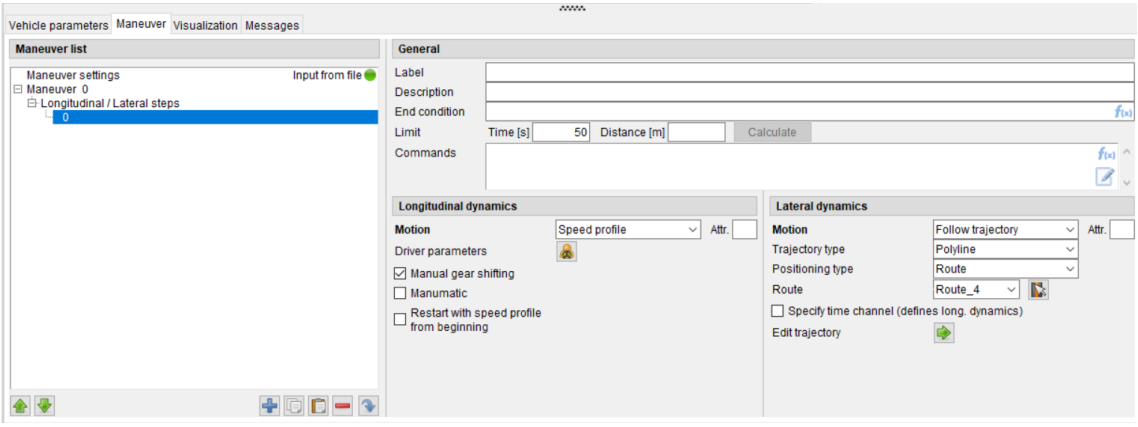


Figure 3.2: IPG CarMaker Maneuver: Defining Vehicle Maneuvers.

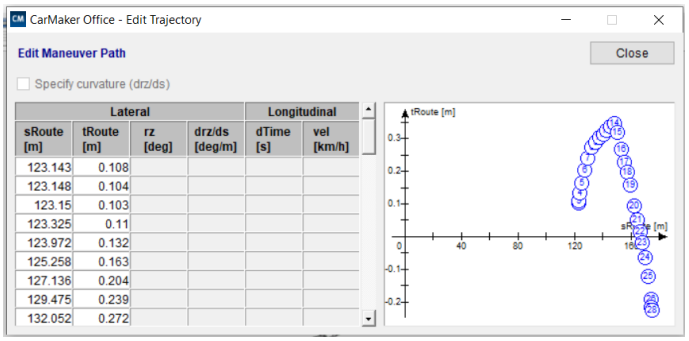


Figure 3.3: IPG CarMaker Edit Trajectory: Visualizing Lateral Maneuvers Aligned with Longitudinal Coordinates.

is run using the input file, it is possible to parameterize and adjust the parameters accordingly.

3.1.1.3 Driver

One of the driver parameters is the driving mode. Predefined driving modes are defensive, normal, and aggressive. Creating a custom driver profile by applying Driver Adoption is also possible. This is an important functionality for adapting the simulation driver to the real-world driver and learning the vehicle's limits. The normal driver mode is selected, and maximum acceleration and deceleration values are adjusted based on real-world data.

3.1.2 Traffic

Within the scope of this section, the road is defined, and additional traffic elements are added to the road, such as people, traffic objects, animals, buildings, and barriers.

3.1.2.1 Road

The route has to be defined for performing the test drive, and the first step is setting the road profile. The digital twin of a highway section is provided by AstaZero,

which is also a partner in the project. The provided file is a detailed map of a highly accurate representation of the test track in an OpenDrive format and can be seen in Figure 3.4. This file is imported to CarMaker directly without any changes. The virtual road map includes lanes, road markings, curvature, elevation changes, interchanges, junctions, roundabouts, borders, traffic signs, and light poles.

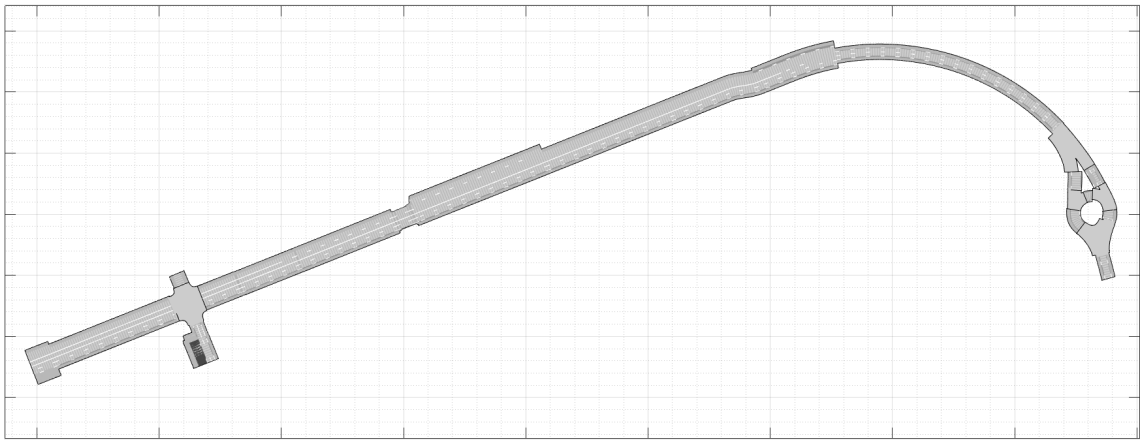


Figure 3.4: Highway Section Road Profile Provided with OpenDrive File from AstaZero.

3.1.2.2 Traffic Objects

Some of the details in the scenario include the pedestrian model, stationary vehicles, barriers, buildings, and the surrounding environment, such as trees. These properties could not be imported to CarMaker through the OpenDrive file and, therefore, added to the simulation environment separately.

3.1.2.3 Pedestrian

The pedestrian, also referred to as the Vulnerable Road User (VRU), must mimic its movement as close to reality as possible. Depending on the test case, a different model of a pedestrian, an adult, or a child is added to the simulation environment. These models are predefined by EuroNCAP and selected from the built-in models. Figure 3.5 shows a visual representation, and Table 3.1 shows the properties of the pedestrian dummy. Test cases 1 and 3 target an adult, while test cases 2 and 4 target a child. The movement of the pedestrian is dynamic and controlled by an input file, similar to the 'Input From File' window. However, unlike car movement, the pedestrian trajectory is not defined by their dynamics and, therefore, follows the exact data given as input.

Figure 3.5: Euro NCAP Pedestrian Dummy Targets (Adult on the left and Child on the right)

Property	Value
Adult Height	181 cm
Adult Width	50 cm
Child Height	114 cm
Child Width	30 cm

Table 3.1: Properties of Euro NCAP Pedestrian Dummy Targets

3.1.2.4 Stationary Vehicles

Two stationary cars are present in the scenario, occluding the pedestrian. These vehicles are added to the simulation environment to represent vehicles of similar size to those in the real world. The position of the two stationary cars is adjusted according to the scenario details given in 1.1.

3.1.2.5 Barriers

Barriers are present on the real test track on each side of the road and, therefore, are added to the simulation environment separately. These barriers significantly impact the generation of detection points and object tracking due to the high level of reflections they cause.

3.1.2.6 Buildings

There is one building on the real test track, which is the warehouse. A similar warehouse model is added to the simulation environment that reflects the actual sizes of the building.

3.1.2.7 Trees

The road where the tests are conducted is lined with trees on both sides. This has been applied to CarMaker by adding a tree strip.

3.1.3 Sensor Models

This section describes the parametrization to create the digital twin of the sensor models. A significant amount of this work belongs to sensor modeling. In particular, radar sensor modeling is already challenging due to many properties that must be considered during correct parameterization.

3.1.3.1 High Fidelity (HiFi) Radar

The radar sensor analyzes the signal-to-noise ratio (SNR) to detect obstacles. This sensor model incorporates object occlusion effects, antenna gain modeling, and propagation losses. Besides providing information about the relative position, velocity, and acceleration of detected objects, the sensor indicates the probability of detection levels, probabilities of existence, and the received power.

3.1.3.2 Raw Signal Interface (RSI) Radar

The Radar Raw Signal Interface sensor mimics a physical sensor by simulating the transmission of electromagnetic waves within a digital environment. This sensor model considers physical propagation, including multipath propagation, doppler shift, and the possibility of false positive or negative object detection. The output data consists of the number of detections at each time stamp of the ray tracing job, cartesian coordinates of detections, power received, and velocity for each detection in the point cloud.

The RSI Radar output data is not directly available from CarMaker. The sensor cluster window is configured, which allows the data to be streamed and the sensor cycle time and cycle offset to be adjusted. The sensor data is transferred via the RSDS (Raw Signal Data Stream) client on a TCP/IP connection. As two radars are running, two separate sensor clusters are configured, and each has the same socket number but a unique host port number for the TCP/IP data interface. The response from the connection interface is shown in Figure 3.6.

```
$ ./rsds-client.exe
Hostip : localhost
Hostport : 2211
Socket : 304
RSDS: can't connect 'localhost:2211'
        retrying in 1 second... (19)
RSDS: can't connect 'localhost:2211'
        retrying in 1 second... (18)
RSDS: Connected: IPGMovie 13.0 2023-11-13
RSDS Client: deleting Radar structs
Closing RSDS-Client...
TOTAL: 685 Frames received, for 1077940 bytes (1MB)
```

Figure 3.6: Response from RSDS Client.

The parameterization of the sensors has been applied by adapting information available in [25], [26]. Some of the final properties of the radar sensor are shown in Table 3.2.

3.1.4 Environment

The environment parameters have also been set, representing the atmosphere, temperature, and weather, representing the same conditions as the real test drive. These parameters from the day of data collection in the test track are shown in Table 3.3.

Table 3.2: Sensor Properties Based on Technical Characteristics.

Property	Value	Unit	Radar Model	Sensor Type
Operating Frequency	76.5	GHz	HiFi & RSI	FL&FR
FOV (azimuth)	150	deg	HiFi & RSI	FL&FR
FOV (elevation)	10	deg	HiFi & RSI	FL&FR
Maximum Range	150	m	HiFi & RSI	FL&FR
Probability of Detection	0.5	-	HiFi	FL&FR
Probability of False Alarm	10^{-4}	-	HiFi	FL&FR

A study on the effects of these parameters is available in Section IV, Results under sensitivity analysis.

Table 3.3: Environment Properties on the Data Collection Day.

Property	Value	Unit
Temperature	3	deg
Rain Rate	0	mm/h
Fog	0	
Wind	15	km/h

3.2 Data Collection

The collected data consists of Ground Truth Data, Experimental Data, and Synthetic Data. The environments in each data have been collected and the information they contain are presented in Table 3.4.

Table 3.4: Collected Data, Collection Environment, and Information Available.

Data	Environment	Host	Target	Sensor
Ground Truth Data	Real-World Test Drives	X	X	
Experimental Data	Real-World Test Drives	X		X
Synthetic Data	Simulation Test Drives	X	X	X

Each data type is referenced to a different point; therefore, the first step is to convert the data to the same coordinate system, which is the ground truth coordinate system referenced to a single point on the map.

For visualization, Figure 3.7 and Figure 3.8 represent the same time instance from the data collection at AstaZero and CarMaker, respectively, from the camera view. The data is processed both on the detection level and object level.

3.2.1 Detection Level

Point cloud data is used to process detection-level data.

Figure 3.7: Camera View of the CPNA Scenario (from Real Data Collection at AstaZero).

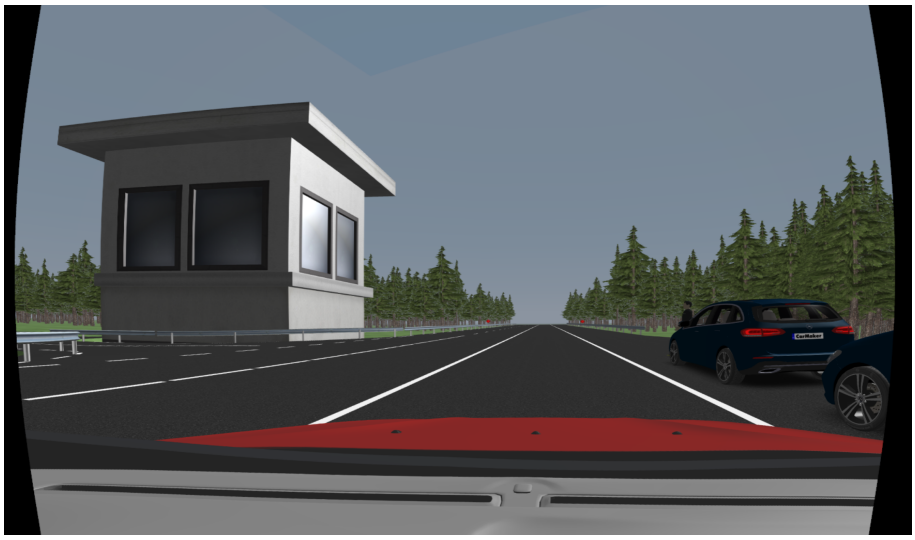


Figure 3.8: Camera View of the CPNA Scenario (from Synthetic Data Collection on IPG CarMaker).

3.2.1.1 Experimental Data

The recorded experimental data includes the state of the detection point relative to VCS and an ID indicating a possible object. This data is transformed into the ground truth coordinate system by taking the cumulative travel distance of the vehicle at each time step and then subtracting the total traveled distance. The purpose of this step is to describe the detection points relative to the reference point.

The ID of the target that is obtained with the company's internal tool is used to find the detection points associated with the target. As objects are created around these detection points, this ID is used to filter the detection points and generate the

object around the detection points. The object-level data processing step explains it more clearly.

The point cloud of the detection points for the entire log is shown in Figure 3.9. The data shown in the left figure is unfiltered based on the object ID. Although the target's trajectory is identifiable, not all the points are associated with the target. The right figure displays the filtered detections associated with the target.

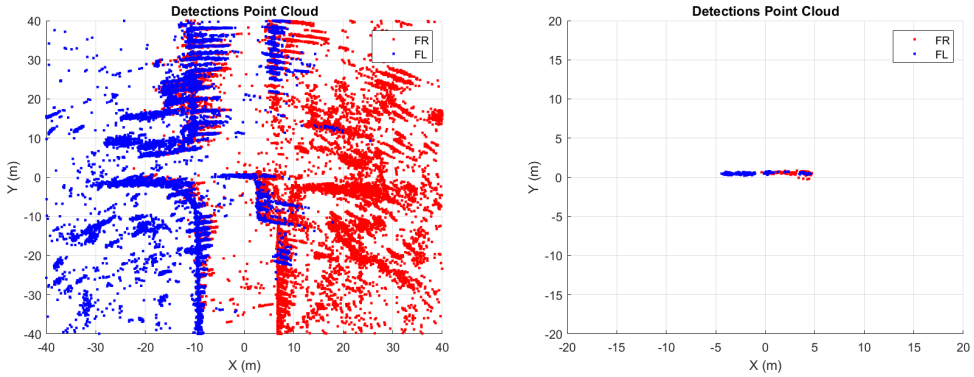


Figure 3.9: APTIV MRR Radar Sensor Output from Real Data Collection (Unfiltered Data on the Left, Filtered Data on the Right).

3.2.1.2 Simulation Data

The detection points are produced using RSI Radar. Figure 3.10 shows a screenshot from the scene. The blue points represent the detection points of the front right radar, and the purple points represent the detection points of the front left radar.

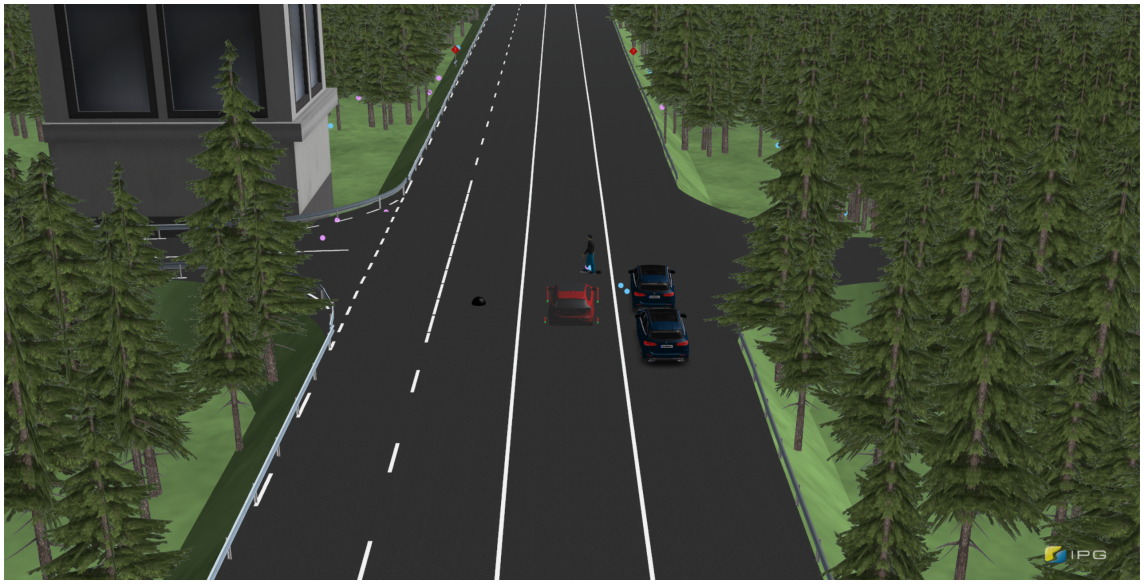


Figure 3.10: Screenshot from IPG CarMaker Showing Detection Points Created Using RSI Radar.

Referring to the properties of RSI data, only the raw signals are provided. As a result of that, the raw detection points are relative to the sensor frame. For the

comparability, the first step of conversion is to transform these detection points to the vehicle coordinate frame. Following that, these points are converted into the ground truth coordinate system.

It can be seen from Figure 3.11 that the point cloud seems quite messy due to reflections from the environment and other noise sources. It is important to eliminate this noise to get a more accurate representation of the environment. To implement this, some of these points are removed by applying a threshold to the received signal strength and filtering those points below this threshold. This threshold is set higher for the adult case and lower for the child case. This is as per the formula of the radar equation provided in Equation 2.2, which shows how the received signal strength is proportional to the RCS of the target. The RCS of a child is smaller that of an adult given that the size of the adult dummy is larger than that of a child dummy, which can be observed from Table 3.1.

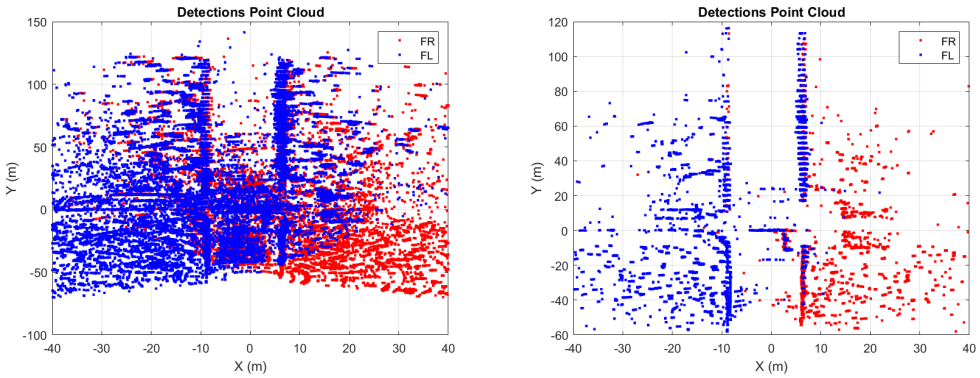


Figure 3.11: IPG CarMaker Radar Sensor Output from Synthetic Data Collection (Unfiltered Data on the Left, Filtered Data on the Right).

3.2.2 Object Level

Tracked object properties are used for processing object-level data.

3.2.2.1 Experimental Data

The experimental object-level data is available in two forms. One form is the output from APTIV's radar tracker. This data can be visualized by the company's internal tool. This tool is used to find the objects associated with the target. Once the ID of the target is determined, the radar tracker output is extracted. The visualization on the internal tool is shown in Figure 3.12; each rectangle corresponds to a possible object.

The extracted output from the APTIV's radar tracker is used as a standard of performance comparison with the developed tracker.

The second form of the object-level experimental data generation process is a two-step process. The first step is extracting the detection points associated with the

target. In the second step, these detection points are processed through the developed tracker explained in Section II, Theory.

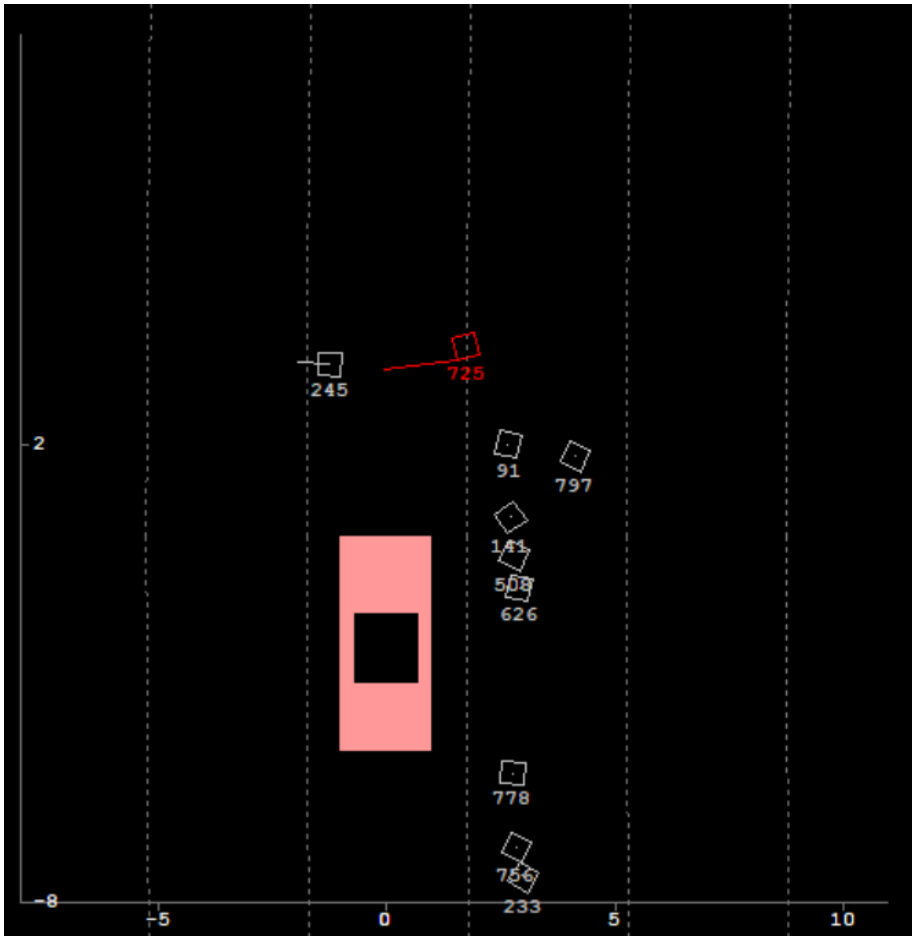


Figure 3.12: Object Visualization using the Visualization Tool (Target ID: 725).

3.2.2.2 Simulation Data

The objects are produced using HiFi Radar. Figure 3.13 shows a screenshot from the scene. The blue rectangles represent the objects seen by the front right radar, and the purple points represent those seen by the front left radar. Besides the coordinates of the objects, some more properties of the object are also available, such as the probability of detection, probability of existence, signal strength, and signal-to-noise ratio. The object output has been filtered by using only the ones above a certain probability of detection and probability of existence.

Unlike the RSI Radar, this radar tracks the objects, not the detections. The output data consists of the cartesian coordinates of the tracked objects, and it is, again, relative to the sensor frame. The data has gone through a rotation matrix to get the data relative to the vehicle coordinate system. The resulting data appears to have significant fluctuations and, therefore, has to go through another processing step. At this step, Savitzky-Golay filtering is applied.

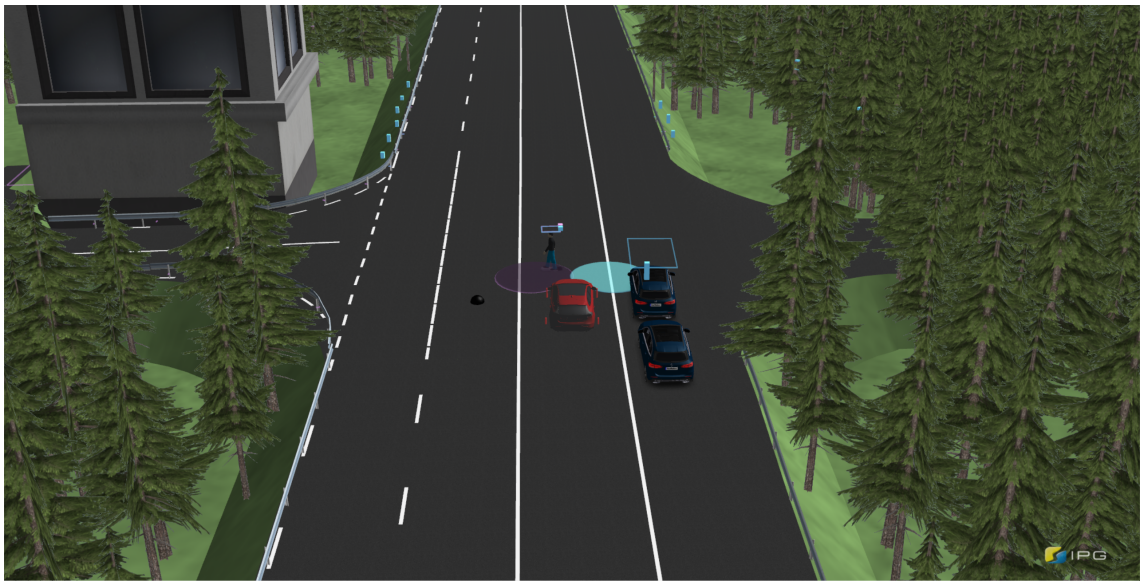


Figure 3.13: Screenshot from IPG CarMaker Showing Objects Created Using HiFi Radar.

The last step of processing the data is applying an interpolation for the comparability of results. The interpolation criteria is the time the object is detected at the latest.

4

Results

This section explores answers to the research questions defined in the previous sections. The simulation parameters are set to final values for comparability and reproducibility of the results. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) is used as a comparison metric to measure the difference between real and simulated data. The closer the resulting error value value is to zero, the more similar the results are.

The Root Mean Square Error (RMSE), in general, is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.1}$$

The Mean Absolute Error (MAE), in general, is calculated using the following formula:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

where:

- n is the number of observations,
- y_i is the observed value,
- $\hat{y}_i$ is the predicted or estimated value.

Specifically, y_i is the real data, and $\hat{y}_i$ is the synthetic data that is the simulated data. The difference between these metrics is that the square operation makes the RMSE more sensitive to outliers. The results for RMSE and MAE are generated in meters.

4.1 Simulation-to-Reality Gap

The simulation-to-reality gap refers to the difference between the data from the real-world test and the data from the simulation results. This section is seeking answers to the following research questions:

1. **Which properties are significant to represent simulations accurately to the reality?**

   Among the important properties observed is the selected vehicle model, besides the accurate input data. It stands out as a critical factor influencing vehicle movement, which directly affects the recognition of the environment depending on the vehicle's position. Besides that, the correct parametrization of the sensor is equally imperative.

2. **Does the proposed method allows comparability to the simulation-to-reality gap?**

   The proposed method followed an iterative approach. At first, the deficiencies are identified by checking the current output and then integrating the missing element to correct the deficiencies in the current output and produce the digital twin of the component. The simulation results are shown in Figures 4.9, 4.10, 4.11, and 4.12 in the order of test cases. The simulations have shown a good correlation to real tests and are comparable within the limitations. It must be added that the model still has the potential to be improved, and the suggested improvements are provided in Section V, Conclusion.

3. **How does the simulation-to-reality gap varies among the different test runs?**

   The sensor parameters remain consistent across all test cases, allowing the simulation-to-reality gap to be assessed fairly. The simulation results for the VUT are shown in Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8, presenting the longitudinal and lateral movements in the order of test cases. Each test run is a combination of the driver's behavior and the vehicle's reaction. The overall comparison is summarized in Table 4.1. The table clearly shows that the minimum RMSE value is for Case 1, hence the minimum simulation-to-reality gap. Case 2 follows that, Case 3 and Case 4 coming after.

   It can be observed from Figure 4.1 and Figure 4.3 that the vehicle is moving at a slower speed for Case 1 and Case 2. Thus, there is no need for sudden speed changes, which does not require a fast response from the car in the simulation environment. This leads to achieving a better simulation-to-reality gap. Similarly, Figure 4.5 and Figure 4.7 indicate that the vehicle is moving faster. Hence, vehicle dynamics play a critical role in giving the same response as the real vehicle. Specifically, it can be observed from Figure 4.7 that case

4 shows the highest deviation, mainly due to the late acceleration response of the vehicle, leading to the total distance being considerably smaller than the actual traveled distance.

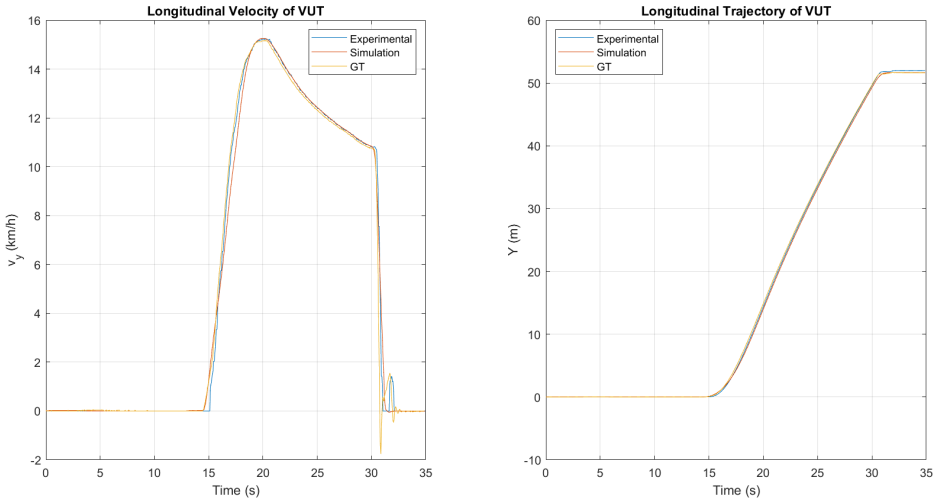


Figure 4.1: Longitudinal Profile of the Vehicle Under Test (VUT) for Case 1 (Velocity and Distance, $v_{host(long)} = 15$ km/h).

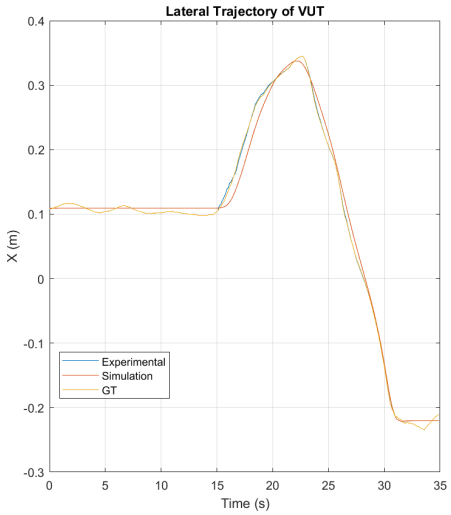


Figure 4.2: Lateral Profile of the Vehicle Under Test (VUT) for Case 1 (Distance, $v_{host(lat)} \approx 0$ km/h).

Table 4.1: Comparison Between the VUT Trajectory - MAE(m).

Case Number	Host Speed	Pedestrian Model	MAE (Long)	MAE (Lat)
1	15 km/h	Adult	0.18	≈ 0
2	15 km/h	Child	0.36	≈ 0
3	30 km/h	Adult	0.4	≈ 0
4	30 km/h	Child	3.86	≈ 0

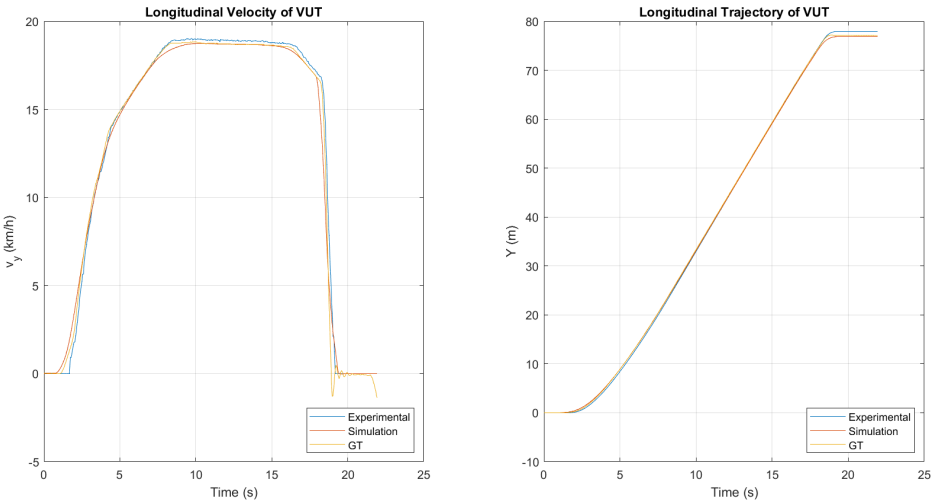


Figure 4.3: Longitudinal Profile of the Vehicle Under Test (VUT) for Case 2 (Velocity and Distance, $v_{host(long)} = 15$ km/h).

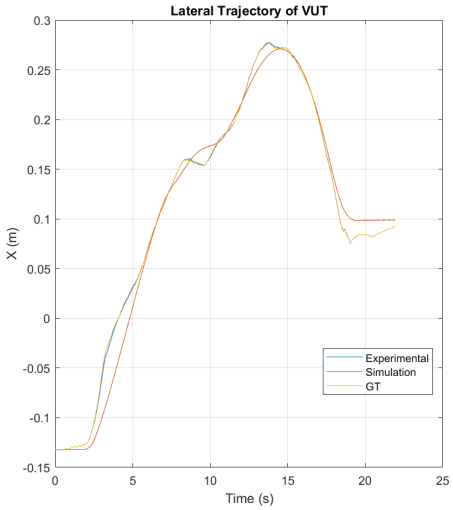


Figure 4.4: Lateral Profile of the Vehicle Under Test (VUT) for Case 2 (Distance, $v_{host(lat)} \approx 0$ km/h).

4.2 Sensor Fidelity

It is often difficult to know whether a highly detailed radar sensor model is always necessary for all simulation scenarios; the question of how to choose the appropriate level of detail for the model is still an ongoing area of research [27].

As previously mentioned, the vehicles perceive their environment in terms of objects. Therefore, the results are generated for object-level comparison and include:

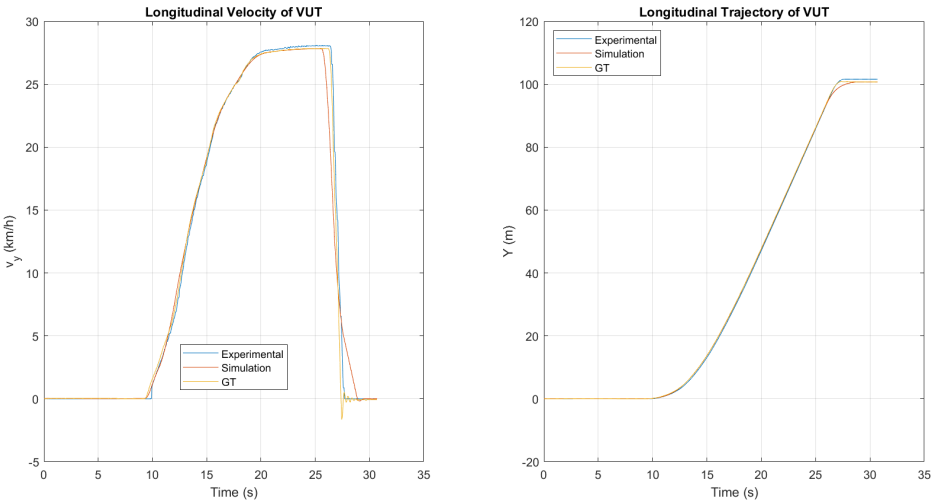


Figure 4.5: Longitudinal Profile of the Vehicle Under Test (VUT) for Case 3 (Velocity and Distance, $v_{host(long)} = 30$ km/h).

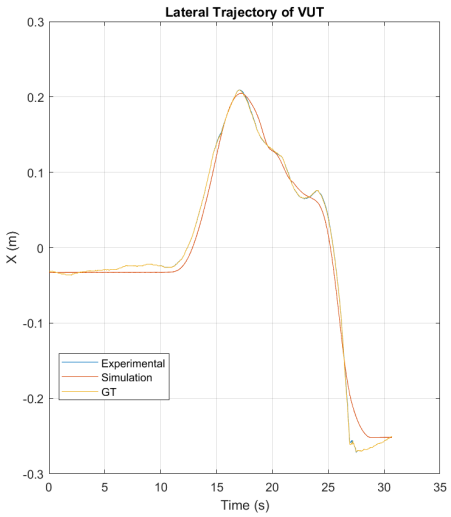


Figure 4.6: Lateral Profile of the Vehicle Under Test (VUT) for Case 3 (Distance, $v_{host(lat)} \approx 0$ km/h).

- Experimental Data: Detection points from the real world are processed using the developed tracker.
- Simulation Data: The data is available in two forms for sensor fidelity comparison. The first dataset includes data from the RSI radar, representing high-fidelity radar sensing. The second dataset includes data from the Hi-Fi radar that is of lower fidelity despite its name suggesting otherwise.

This section answers the following research questions:

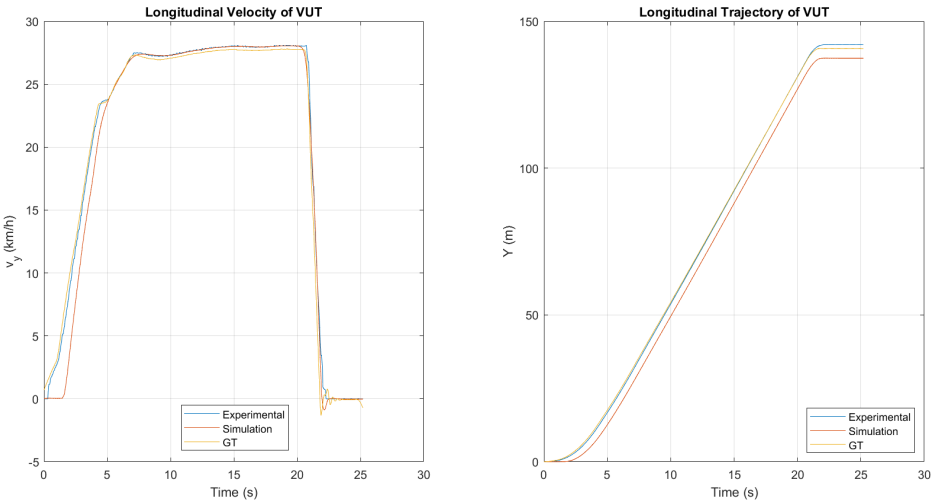


Figure 4.7: Longitudinal Profile of the Vehicle Under Test (VUT) for Case 4 (Velocity and Distance, $v_{host(long)} = 30$ km/h).

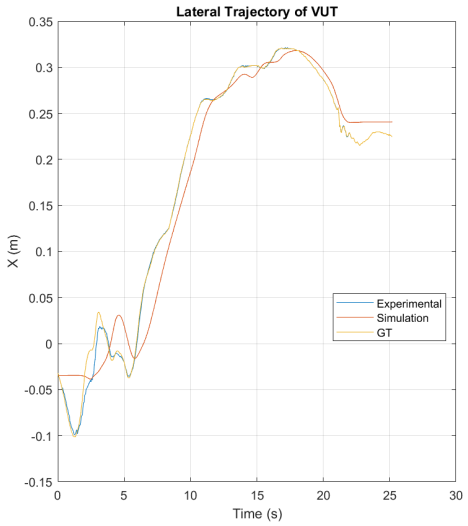


Figure 4.8: Lateral Profile of the Vehicle Under Test (VUT) for Case 4 (Distance, $v_{host(lat)} \approx 0$ km/h).

1. **Does the proposed method allow comparability to the sensor fidelity?**

   The proposed approach identifies the necessary steps to replicate the same scenario and implements to a certain extent. The results demonstrate that the signal characteristics are preserved, and the VRU is successfully tracked by following the proposed approach. The outputs of the simulation sensors have shown a good correlation to real test results and can be considered comparable to the sensor fidelity within the limitations.

2. **How does the fidelity of the sensor model affect the execution efficiently representing the necessary degree of fidelity?**

   The graphical results are presented in Figures 4.9, 4.10, 4.11, and 4.12, corresponding to the order of test cases. Each figure displays the longitudinal and lateral tracking of the pedestrian. Numerical comparison results are provided in Table 4.2 and Table 4.3.

   Looking at all these figures together, there is a gap between the sensor output during the tests in the real world and the simulation. This gap is related to the vehicle's different movements between the real-world tests and the simulation runs due to the vehicle model not being a perfect representative of the physical car, as mentioned under the previous subsection, the simulation-to-reality gap. It is recommended that the results of the sensors be evaluated considering this gap.

   Figure 4.10 and Figure 4.12 show that the detection points for children are sparser compared to the test cases where the pedestrian is an adult in Figure 4.9 and 4.11, as expected, considering their smaller radar cross-section compared to adults. Particularly when the child is occluded behind stationary vehicles, fewer detection points were observed. An important detail observed from the graphical results is that the RSI radar closely matches the corresponding time instance when the object is detected, indicating sensor reliability.

   On the contrary, the HiFi sensor failed to identify the target when the Vulnerable Road User (VRU) was occluded by stationary cars due to high reflection resulting from them. This is an expected situation which can be referred to as object merging due to the effects of low resolution. This can be said to be a huge difference compared to the RSI radar.

   One more thing that can be observed is that target detection proves to be a challenge as the speed of the VUT increases, as evidenced by the delay in recognizing the target after it has already traveled a certain distance. This can be seen clearly from Figure 4.9 and 4.10.

   Overall, across all test cases, the RSI radar outperformed the HiFi radar. The reason for that is the fidelity of RSI radar is much higher compared to the HiFi radar as a result of a more realistic perception of the environment. Notably, the objects are created using the same processing method (using detection points) by processing through the same tracker, directly influencing the output.

4.3 Sensitivity Analysis

Sensitivity analysis is a technique used to understand how variables affect the simulation. In other words, it helps to understand how sensitive the output of the simulation given the changes in its inputs. During the early stages of the thesis, it

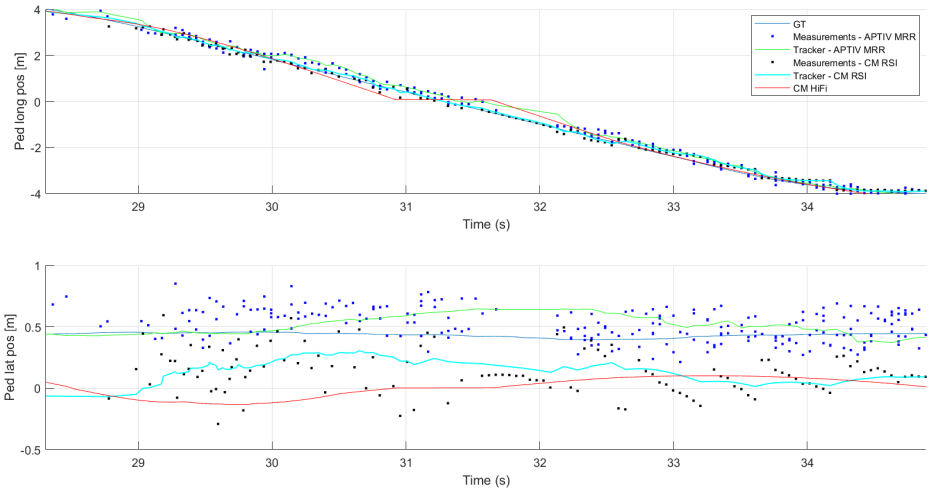


Figure 4.9: Sensor Output for Case 1 (v_{host} = 15 km/h & Adult).

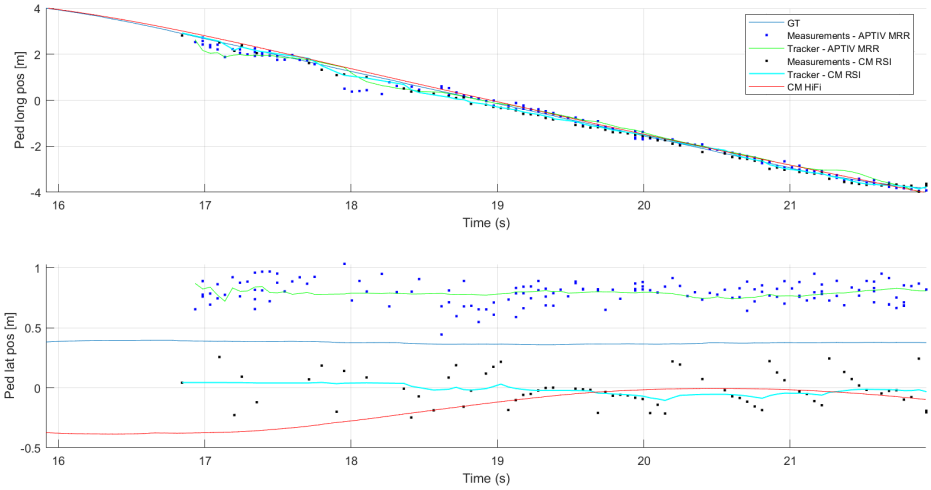


Figure 4.10: Sensor Output for Case 2 (v_{host} = 15 km/h & Child).

was observed that some parameters are highly critical for the performance of sensor models while others are not. The main objective of the sensitivity analysis is to find those input parameters that have significant effects on the simulation and how they affect the system. This section focuses on parameters specific to the RSI radar and some environmental parameters, following previous work addressing HiFi radar parameters [28]. Sensitivity analysis is conducted by changing one parameter at a time while keeping others unchanged. To this end, results are generated across various parameters for Case 1. The best-case scenario with the minimum simulation-to-reality gap, Case 1, is selected for comparison.

This section answers the following research questions:

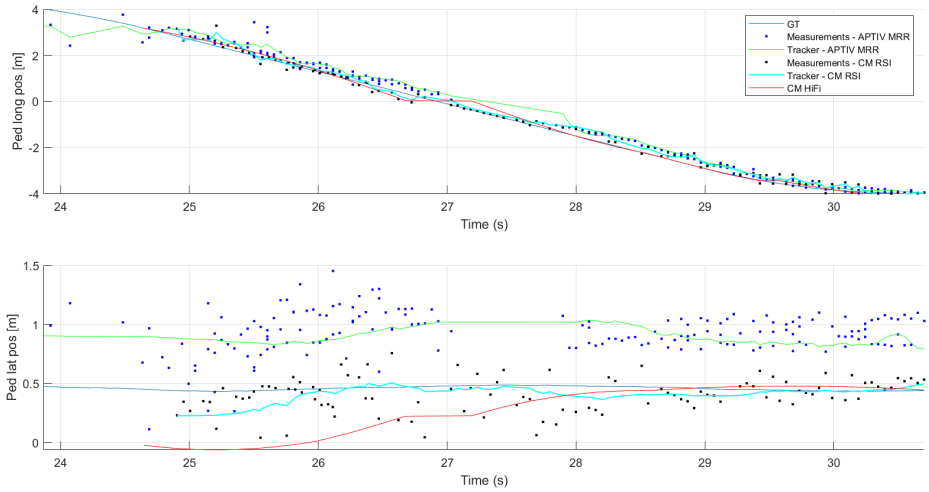


Figure 4.11: Sensor Output for Case 3 (v_{host} = 30 km/h & Adult).

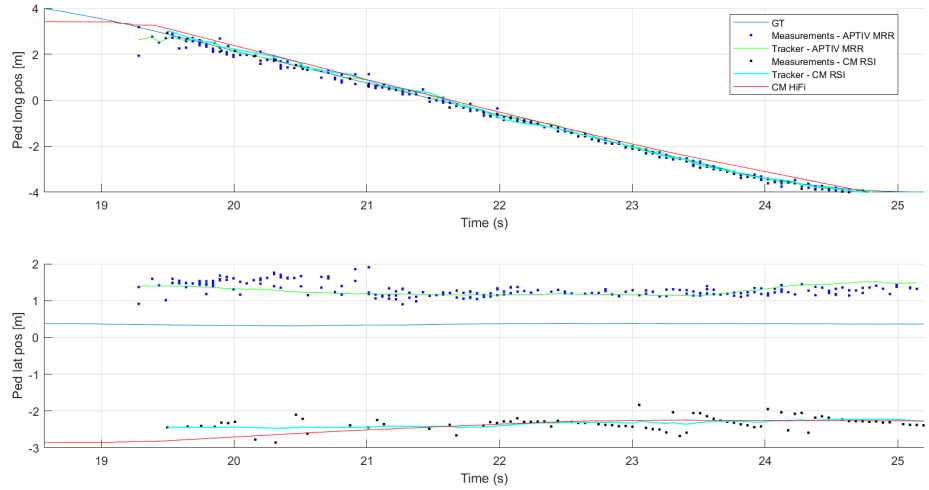


Figure 4.12: Sensor Output for Case 4 (v_{host} = 30 km/h & Child).

1. **Which parameters have a strong influence on the performance of simulations?**

   Besides the key parameters such as field of view, frequency of operation, and range, there are adjustable parameters during the parameterization of the radar, which are the noise bandwidth, number of rays, and number of reflection points. There are also adjustable environmental parameters that are specified within the next question.

2. **How do these parameters influence the simulation results?**

   Some critical parameters for the RSI Radar include noise bandwidth, number of rays, and number of reflection points.

Table 4.2: Root Mean Square Error Values between Simulated and Real Data for Pedestrian Longitudinal Movement Tracking - RMSE(m).

Case	RMSE (HiFi)	RMSE (RSI)	RMSE (m)
h15 Adult	0.24	0.21	0.013
h15 Child	0.23	0.21	0.016
h30 Adult	0.35	0.31	0.014
h30 Child	0.10	0.11	0.026

Table 4.3: Root Mean Square Error Values between Simulated and Real Data for Pedestrian Lateral Movement Tracking - RMSE(m).

Case	RMSE (HiFi)	RMSE (RSI)	RMSE (m)
h15 Adult	0.50	0.39	0.18
h15 Child	0.92	0.80	0.48
h30 Adult	0.67	0.5	0.57
h30 Child	3.63	3.6	4.01

As seen in Table 4.4, the noise bandwidth does not influence results.

Table 4.4: Influence of Noise Bandwidth - RMSE(m).

Noise Bandwidth [Hz]	RMSE (Long)	RMSE (Lat)	Total Distance [m]
100	0.21	0.39	51.6401
200	0.21	0.39	51.6401
300	0.21	0.39	51.6401

Table 4.5 illustrates the influence of the number of rays used for modeling the RSI Radar. It is observed that increasing the number of rays results in decreased simulation accuracy. While it may seem that a higher number of rays offers a better perception of the environment, this is not always the case. Increasing the number of rays can also reduce the degrees between rays, thus shrinking the reception sphere. This effect is particularly pronounced when the target size is small, and the environment is not complex. Consequently, using more rays results in fewer or different paths [29].

Table 4.5: Influence of Number of Rays - RMSE(m).

Number of Rays (H × V)	RMSE (Long)	RMSE (Lat)	Simulation Duration (s)
1287 × 100	0.21	0.39	34.9
2586 × 200	0.23	0.47	108
6483 × 500	0.24	0.47	358

Table 4.6 illustrates the influence of the number of reflections. This parameter is configured through the Sensor Configuration Window. It has been observed that there was no impact on results, particularly due to the small target size, it was found to be accurate with few reflection points.

Table 4.6: Influence of Number of Reflections - RMSE(m).

Number of Reflections	RMSE (Long)	RMSE (Lat)
3	0.21	0.39
20	0.21	0.39
50	0.21	0.39

Rain rate has been found not to affect the simulation results, as can be seen in Table 4.7.

Table 4.7: Influence of Rain Rate - RMSE(m).

Rain Rate [mm]	RMSE (Long)	RMSE (Lat)	Total Distance [m]
0	0.21	0.39	51.6401
10	0.21	0.39	51.6401
20	0.21	0.39	51.6401

The temperature, the speed of the wind, and the angle of wind have been found to influence the distance traveled, hence affecting the results as shown in Table 4.8, Table 4.9, and Table 4.10.

Table 4.8: Influence of Temperature - RMSE(m).

Temperature [°C]	RMSE (Long)	RMSE (Lat)	Total Distance [m]
-30	0.21	0.40	51.6367
3	0.21	0.39	51.6401
30	0.31	0.36	51.6392

Table 4.9: Influence of Speed of Wind - RMSE(m).

Speed of Wind (km/h)	RMSE (Long)	RMSE (Lat)	Total Distance [m]
0	0.21	0.39	51.6401
30	0.24	0.42	51.6510
60	0.64	0.48	51.65

Table 4.10: Influence of Wind Angle - RMSE(m).

Wind Angle [°]	RMSE (Long)	RMSE (Lat)	Total Distance [m]
13	0.21	0.39	51.6401
60	0.27	0.43	51.6441
-60	0.23	0.38	51.6441

5

Conclusion

5.1 Summary

Within the scope of this thesis, a digital twin-based simulated environment for automotive radar perception analysis has been implemented. A significant focus of the thesis is on replicating the digital twin of the components existing in the physical scenario in the virtual environment due to the complexities present in the physical world. To validate the methodology and tests, different cases were simulated using data collected from variations in host speed and pedestrian targets within the same scenario. This approach proved to be helpful in assessing the effect of varying driving profiles, as well as of target models on the simulations.

The virtual environment is IPG CarMaker. The output data consists of objects using HiFi Radar and detection points using RSI Radar. As autonomous vehicles perceive their environments based on objects, the results are evaluated on object-level data. The detection points are processed through an object tracking algorithm for that purpose. The results are analyzed by using RMSE and MAE metrics. The analysis evaluates the simulation-to-reality gap, the effects of sensor fidelity, and a sensitivity analysis.

The findings suggest that digital twin-based analyses of automotive radars are a promising approach for virtual testing. It has been shown that the RSI sensor provides a more detailed analysis but also requires more work to process signals and more hardware sources. Although the study has certain limitations, its outcomes agree with the statistics of the real data. Further research will provide more detailed results while this work is a step towards high-fidelity virtual testing.

5.2 Future Work

There are several sensors available to sense the surroundings such as lidar, cameras, ultrasonic sensors besides radar. Today's vehicles do not depend on information from only one type of sensor but instead fuses data from multiple sensors. Thus, validation of the radar simulations should be done both isolated from other sensor

simulations as well as together with other sensor simulations in a larger sensor fusion context to ensure virtual testing of the overall system.

An interesting area for future research is the development of HiL testing methods tailored for radar simulations. This involves the integration of physical radar hardware into simulation environments. Advanced HiL setups will provide a more accurate representation of real-world scenarios and, ultimately, a more accurate validation process.

Overall, further research is recommended to improve the accuracy and scalability of the simulation models. It will enhance the efficiency of this approach for the virtual testing of automotive radar systems.

5.3 Societal, ethical and ecological aspects

Virtual testing enhances automotive safety by allowing extensive testing of driving systems overall, potentially reducing accidents and fatalities. It can also lead to the development of safer and more affordable vehicles, making advanced safety features more accessible to a broader population.

The proposed concept of digital twins has environmental and climate effects. Running the physical tests in real-world scenarios brings huge costs to the environment. First of all, running the tests will require the use of batteries and fuel. The virtual testing concept prevents these costs since these resources are not needed anymore. Considering the mentioned costs, CO_2 consumption will also expected to be reduced significantly.

Being a complex problem, simulations tend to be computationally heavy and require considerable computational resources. This will cause increasing processing needs. In such cases, the solution can be simplified.

Ethical considerations often arise with virtual testing, depending on how much these tests can replace or complement traditional testing. Relying too much on virtual tests can pose risks, which makes it very important to understand and take into account the limitations of the tests.

These are important points that should be kept in mind for determining the future usability of digital twins and virtual testing.

Bibliography

[1] RIG Models. Volvo V40 3D model. https://rigmodels.com/model.php?view=Volvo_V40-3d-model__bec5dce963304f5b9cdeca163931a600&searchkeyword=v40&manualsearch=1.

[2] EUROPEAN NEW CAR ASSESSMENT PROGRAMME (Euro NCAP) TEST PROTOCOL – AEB Car-to-Car systems. https://cdn.euroncap.com/media/58226/euro-ncap-aeb-vru-test-protocol-v303.pdf. Accessed: Apr. 22, 2024.

[3] Next steps towards vision zero. https://visaozero2030.pt/wp-content/uploads/08-Next_steps_towards_vision_zero.pdf, 2024. Accessed: 2024-05-17.

[4] Anthony Ngo. *A Methodology for Validation of a Radar Simulation for Virtual Testing of Autonomous Driving*. PhD thesis, Universität Stuttgart, January 2023.

[5] IPG Automotive Group. *CarMaker User's Guide*, 2020.

[6] Enabling VIrtual valiDation and vErificatioN for ADAS and AD feaTures. https://www.saferresearch.com/index.php/projects/evident. Accessed: Apr. 22, 2024.

[7] AstaZero. About us. https://www.astazero.com/en/about-us/, 2024. Accessed: 2024-05-14.

[8] Christoph Sohrmann. Virtual testing of automotive sensor systems, March 2022.

[9] Ashley F. Emery. Special issue: Sandia v&v challenge problem. *Journal of Verification, Validation and Uncertainty Quantification*, 1, 2016.

[10] Vishakha R. Nalawade, Manish Kumar, Alok Singh Bhandari, and Thrilochan Sharma. Comparison of simulators based on scenario-based testing of autonomous vehicles. In *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, pages 1–5, 2022.

[11] Edward Glaessgen and David Stargel. The digital twin paradigm for future nasa and u.s. air force vehicles. 04 2012.

[12] Michael Grieves. Origins of the digital twin concept, August 2016.

[13] Mohd Javaid, Abid Haleem, and Rajiv Suman. Digital twin applications toward industry 4.0: A review. *Cognitive Robotics*, 3:71–92, 2023.

[14] IBM. What is a digital twin? https://www.ibm.com/topics/what-is-a-digital-twin, 2024. Accessed: 2024-05-14.

[15] Paula Muñoz. Measuring the fidelity of digital twin systems. In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, MODELS '22, page 182–188, New York, NY, USA, 2022. Association for Computing Machinery.

[16] Keith Raney. Radars. In Eni G. Njoku, editor, *Encyclopedia of Remote Sensing*, pages 547–558. Springer New York, New York, NY, 2014.

[17] S. Pliefke, M. Germer, A. Höfer, et al. Validation of a ray-tracing-based radar sensor model. *ATZ Electron Worldw*, 16:40–43, 2021.

[18] Jónas Agnarsson. *Simulation of a Radar in Flames: A Ray Based Radar Model*. Dissertation, Uppsala University, 2013.

[19] Vivek Kumar. *Physics-Based and Data-Driven Methods for Structural Health Monitoring At Fine Spatial Resolution*. PhD thesis, Princeton University, 2021.

[20] Takashi Owaki and Takashi Machida. Hybrid physics-based and data-driven approach to estimate the radar cross-section of vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 673–678, 2019.

[21] Phil Kim. *Kalman Filter for Beginners: with MATLAB Examples*. CreateSpace, 2011.

[22] Boris Zimmermann and Achim Kohler. Optimizing savitzky–golay parameters for improving spectral resolution and quantification in infrared spectroscopy. *Applied Spectroscopy*, 67(8):892–902, 2013. PMID: 23876728.

[23] Ronald W. Schafer. What is a savitzky-golay filter? [lecture notes]. *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.

[24] RT-Range ADAS. `https://www.oxts.com/products/rt-range-adas/`. Accessed: May 5, 2024.

[25] Aptiv. Technical characteristics data: Middle range radar. `https://www.aptiv.com/docs/default-source/technical-characteristics-data/technical-characteristics-data_mrr_040218.pdf?sfvrsn=dee5b9f7_2`, 2018. Accessed: 2024-03-20.

[26] AutonomouStuff. Aptiv middle range radar (mrr). `https://autonomoustuff.com/products/aptiv-mrr`. Accessed: 2024-03-20.

[27] Anthony Ngo, Max Paul Bauer, and Michael Resch. A sensitivity analysis approach for evaluating a radar simulation for virtual testing of autonomous driving functions. In *2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 122–128, 2020.

[28] Christian Frieß. Methodology for parametrization and validation of a virtual radar sensor model. Online, 2018.

[29] MathWorks. Ray tracing for wireless communications. Online.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY